Review of Ensembles of Multi-Label Classifiers: Models, Experimental Study and Prospects

Article *in* Information Fusion · January 2018

DOI: 10.1016/j.inffus.2017.12.001

Jose Moyano
University of Cordoba (Spain)
13 PUBLICATIONS **115** CITATIONS
SEE PROFILE

Eva Gibaja
University of Cordoba (Spain)
58 PUBLICATIONS **567** CITATIONS
SEE PROFILE

Krzysztof Cios
Virginia Commonwealth University
208 PUBLICATIONS **5,497** CITATIONS
SEE PROFILE

Sebastian Ventura
University of Cordoba (Spain)
326 PUBLICATIONS **11,437** CITATIONS
SEE PROFILE

Review of Ensembles of Multi-Label Classifiers: Models, Experimental Study and Prospects

Jose M. Moyano[a], Eva L. Gibaja[a], Krzysztof J. Cios[b,c], Sebastián Ventura[a,d,e,*]

[a]Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
[b]Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA
[c]Polish Academy of Sciences, Institute of Theoretical and Applied Informatics, Gliwice, Poland
[d]Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia
[e]Knowledge Discovery and Intelligent Systems in Biomedicine Laboratory, Maimonides Biomedical Research Institute of Córdoba, Spain

Abstract

The great attention given by the scientific community to multi-label learning in recent years has led to the development of a large number of methods, many of them based on ensembles. A comparison of the state-of-the-art in ensembles of multi-label classifiers over a wide set of 20 datasets have been carried out in this paper, evaluating their performance based on the characteristics of the datasets such as imbalance, dependence among labels and dimensionality. In each case, suggestions are given to choose the algorithm that fits best. Further, given the absence of taxonomies of ensembles of multi-label classifiers, a novel taxonomy for these methods is proposed.

Keywords: Multi-label classification, Ensemble methods

1. Introduction

Ensemble learning combine individual learners from heterogeneous or homogeneous modeling in order to obtain a combined learner that improves

[*]Corresponding author
Email address: sventura@uco.es (Sebastián Ventura)

the generalization ability and reduces the overfitting risk of each one of them [1, 2]. In [3], Dietterich specified three reasons why an ensemble classifier is better than a single classifier. First, when picking only one single classifier, we run the risk of choosing a bad one. Second, many learning algorithms use local search and may not find the optimal classifier, so running several times the learning algorithm and combining the obtained models may result in a better approximation to the optimal classifier than any single one. Third, since in most machine learning problems the optimal function cannot be found, the optimal classifier may be reached by combining several classifiers. Ensemble methods have been successfully applied in many fields such as finance [4], bioinformatics [5], medicine [6], image retrieval [7] and recommender systems [8].

The development of ensemble models has been used in a large number of machine learning tasks, such as in Multi-Label Learning (MLL) and more specifically in Multi-Label Classification (MLC), where each object may have multiple labels associated with it [9, 10]. Despite the fact that in MLC many algorithms are based on combining several classifiers, only those whose combine several classification methods that are able to deal with multi-label data are considered as Ensembles of Multi-Label Classifiers (EMLCs) [11]. EMLCs have been successfully applied in image retrieval [12] and predicting drug resistance [13].

Given the advantages of ensemble methods over simple methods, it is interesting to perform an experimental study of the state-of-the-art EMLCs. In [11] and [14] two experimental studies of MLC algorithms were performed. However, none of them includes the state-of-the-art EMLCs. Given the absence of experimental studies contemplating the state-of-the-art and the special characteristics of the EMLCs, the objective of this paper is to perform an experimental comparison and analysis of the state-of-the-art EMLCs over a range of datasets and evaluation metrics. This study is performed taking into account the characteristics of the data, such as imbalance, relationship among labels or dimensionality, indicating which EMLC achieves better performance in each case and giving some guidelines to select the best algorithm according to the characteristics of the dataset.

The rest of the paper is organized as follows: Section 2 presents background in MLC, Section 3 describes different EMLC methods and categorizes them into a novel taxonomy, Section 4 shows the experimental design, Section 5 describes and discusses the results of the experiments and also gives some guidelines to select the best EMLC according to the characteristics of the dataset and finally Section 6 presents conclusions of this work.

2. Background

In this section the formal definition of MLC and the main categorization of MLC methods are presented.

2.1. Formal definition of multi-label classification

Given a d-dimensional input space $\mathcal{X} = X_1 \times \cdots \times X_d$ and an output space of q labels $\mathcal{Y} = \{\lambda_1, \lambda_2, ..., \lambda_q\}, q > 1$, being the cardinality of each label $|\lambda_i| = 2$, a multi-label example can be defined as a pair $(\mathbf{x}, Y)$ where $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$ and $Y \subseteq \mathcal{Y}$ is called labelset. $\mathcal{D} = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq m\}$ is a multi-label dataset composed of a set of m instances [15].

The goal of multi-label classification is to construct a predictive model $h : \mathcal{X} \to 2^{\mathcal{Y}}$ which would provide a set of relevant labels for an unknown instance. Each instance may have several labels associated with it from the previously defined set of labels. So, for each $\mathbf{x} \in \mathcal{X}$, we have a bipartition $\left(Y, \overline{Y}\right)$ of the label space $\mathcal{Y}$, where $Y = h(\mathbf{x})$ is the set of relevant labels and $\overline{Y}$ the set of irrelevant ones.

2.2. Multi-label classification algorithms

MLC algorithms are categorized into three main groups: problem transformation, algorithm adaptation and EMLCs [11]. Problem transformation methods transform a multi-label problem into one or several single-label problems, as Binary Relevance (BR) [16] that decomposes the multi-label learning problem into q independent binary classification problems, and Label Powerset (LP) [17], which generates a single-label dataset where each distinct labelset is considered as a different class. A more extensive list of problem transformation methods can be found in [15].

Algorithm adaptation methods adapted almost all classification techniques for multi-label learning, such as decision trees [18, 19], support vector machines [20, 21], neural networks [22, 23] and instance-based algorithms [24, 25].

Finally, the third group of MLC methods includes the EMLCs. There are many algorithms in multi-label classification, such as BR, that involve combination of several classifiers instead of a single classifier. However, in MLC only are considered as EMLCs those ensembles that involve the combination of several classification methods which are able to deal with multi-label data, so algorithms such as BR are not considered as EMLC since they combine several single-label methods. EMLCs are the object of this work and are described in detail in Section 3.

3. Ensembles of MLC

In this section a total of 16 state-of-the-art ensemble methods for MLC are described. They are first categorized depending on the method they are based. Once the methods have been described, they are then categorized based on a proposed taxonomy.

3.1. Ensembles based on Binary Relevance (BR)

BR combines predictions of several binary classifiers, one for each label in the multi-label dataset [16]. BR is simple, intuitive and resistant to overfitting label combinations because it does not take into account predefined combinations of labels. Thus, it can handle irregular labeling and is able to predict combinations of labels that did not appear in the original training set. However, BR's weakness is the fact that it does not take into account the relationship among the labels, assuming that the labels are independent while in most cases they are not. Several ensemble methods, which are defined below, were developed to overcome BR's problem.

3.1.1. Ensemble of Binary Relevance classifiers (EBR)

The Ensemble of Binary Relevance classifiers (EBR) [26] is generated using bagging [27] for each BR classifier. Generating an ensemble of BRs each with a random selection of instances improve performance of BR due to the diversity among base classifiers. However, EBR still does not take into account the relationship between labels.

3.1.2. Ensemble of Classifier Chains (ECC)

Classifier Chains (CC) [28] generate a chain of q binary datasets, where the feature space of each classifier is augmented with the label predictions of previous classifiers. The order of the selected chain has a direct impact on performance of the classifier, due to the error propagation along the chain at classification time when some of the classifiers in the chain predict poorly. To overcome this problem, Ensemble of Classifier Chains (ECC) [26] trains n CC, each one with a random chain built over a random selection of m training instances sampled with replacement. Then, the final prediction is obtained by averaging the confidence values for each label. Finally, a threshold function is used to create a bipartition of relevant and irrelevant labels.

Both CC and ECC pass label information among binary classifiers, so they take into account correlations among labels and overcome the problem of BR, which ignores such correlations. Also, using ECC reduces the risk of selecting a bad chain ordering which can lead to a bad prediction performance of the classifier. The diversity in ECC is generated by using different chains and by selecting random subsets of instances.

3.1.3. Multi-Label Stacking (MLS)

Multi-Label Stacking (MLS) [29], also called 2BR, involves applying BR twice. MLS first trains q independent binary classifiers, one for each label. Then, it learns a second (or meta) level of binary models, taking as additional inputs the outputs of all the first level binary models, thus taking into account the relationship among labels in the meta-level.

There exist several approaches of MLS depending on the way they gather the predictions in the base-level. $\text{MLS}_{\text{train}}$ uses the full training set for both base and meta levels, but this can lead to biased meta-level training data. MLS_{cv} partitions the data into F disjoints parts, generating each base-level classifier F times, each using $F-1$ partitions for training and the remaining for gathering the predictions. In this way it obtains a non-biased meta-level training set. However, this method is much slower than $\text{MLS}_{\text{train}}$. Another one, MLS_{ϕ} tries to not introduce irrelevant information into the meta-level. If a label is completely uncorrelated with the one being modeled, including its predicted value in the meta-level classifier introduces non interesting information and noise, which could lead to a worse performance. For that, MLS_{ϕ} uses the ϕ correlation coefficient [30] to determine if two labels are correlated or not, pruning labels that are not correlated with the one being modeled in each meta-level classifier.

In all MLS variants, the diversity of the ensemble is achieved by using different feature space in each classifier.

3.1.4. Hierarchy Of Multi-label classifiERs (HOMER)

Hierarchy Of Multi-label classifiERs (HOMER) [31] is a method designed for domains with large number of labels. It transform a multi-label classification problem into a tree-shaped hierarchy of simpler multi-label problems. At each node with more than one label, c children are created by distributing the labels among them with the balanced k-means method [31], making labels belonging to the same subset as similar as possible. To classify a new instance, HOMER starts with the root classifier and passes the instance to each child only if the parent predicted any of its labels. The union of the predicted labels by the leaves generates output for the given instance.

HOMER is based on BR since in each node a binary classifier is used, which predicts if any or none of the labels in the node are associated with the instance. The diversity in HOMER is generated by selecting a subset of the labels and also by filtering the instances in each classifier, keeping only those which are annotated with at least one label.

3.1.5. AdaBoost.MH

AdaBoost algorithm [32] was extensively studied and used in several machine learning tasks [33–35]. The AdaBoost.MH [36] method not only maintains a set of weights over the instances as AdaBoost does, but also over the labels. Thus, training instances and their corresponding labels that are hard to predict, get incrementally higher weights in following classifiers while instances and labels that are easy to classify get lower weights. The diversity in AdaBoost.MH is generated by using different weights for both instances and labels.

AdaBoost.MH is based on BR since each instance is passed to q binary classifiers, so it is the same as applying AdaBoost to q binary classifiers [36].

3.2. Ensembles based on Label Powerset (LP)

LP generates a single-label dataset with a different class for each different combination of labels. In this way, LP takes into account label correlations but its complexity is exponential with the number of labels and it is not able to predict a labelset which does not appear in the training dataset. Besides, many labelsets are usually associated with only few examples, which may lead to an imbalanced dataset and make the learning process more difficult. Many multi-label ensemble classifiers, described below, have been proposed to overcome these disadvantages of LP.

3.2.1. Ensemble of Label Powerset classifiers (ELP)

Ensemble of Label Powerset (ELP) classifiers is proposed in this paper to compare it as a baseline with other LP-based methods. It uses bagging to generate diversity of classifiers and then combines the predictions of the base classifiers by majority voting. As it combines several LP predictions, ELP is able to predict a labelset that does not appear in the original training set. However, its complexity is not reduced with respect to LP as well as the problem of imbalance is not solved.

3.2.2. Ensemble of Pruned Sets (EPS)

Pruned Sets (PS) [37] is similar to LP but it focuses on the most important relationships of labels by pruning the infrequently occurring labelsets, reducing the complexity of the algorithm.

Ensemble of Pruned Sets (EPS) was proposed in [37] to prevent from overfitting effects of pruning and to allow predicting labelsets that do not appear in train data. EPS trains n PS models, each over a subset without replacement of the instances of the original training set. The predictions of each classifier are combined into a final prediction by a voting scheme using a prediction threshold t. The diversity in EPS is created by the random data selected for each base classifier.

3.2.3. RAndom k-labELsets (RAkEL)

RAndom k-labELsets (RAkEL) [38] randomly breaks the set of labels into several small-sized labelsets. RAkEL selects n random k-labelsets and learns n LP classifiers, each one focusing on its own k-labelset. Each model provides binary predictions for each label in its corresponding k-labelset. These outputs are combined for a multi-label prediction following a majority voting process for each label.

RAkEL has several advantages over LP. First, the LP tasks of each classifier are much simpler since they only consider a small subset of the labels. Also, the base classifiers include a much more balanced distribution of classes than using LP with the full set of labels. Further, RAkEL allows to predict a labelset that does not appear in the original training set.

A variation of RAkEL, called RAkEL++ [39], uses the confidence values of each classifier instead of bipartitions in order to generate the final prediction for each label. Another variation, called RAkELd [38], generates disjoint subsets of k labels, taking into account each label exactly once and reducing the complexity of other RAkEL variants

The diversity in all the variants of RAkEL is generated by different selection of labels in each classifier.

3.2.4. Triple Random Ensemble for Multi-Label Classification (TREMLC)

Triple Random Ensemble for Multi-Label Classification (TREMLC) [40] is based on the random selection of features, labels and instances in each classifier of the ensemble. In this way, TREMLC uses three ways to obtain diversity of base classifiers. Then, a LP is built over each randomly selected data. The final prediction of TREMLC is obtained by majority voting.

3.2.5. Chi-Dep Ensemble (CDE)

Chi-Dep [41] groups dependent labels by χ^2 score [42], building a LP classifier for each group of dependent labels and a binary classifier for each independent label. In this way, it achieves an optimal trade-off betwee simple (single-label) and complex (LP) models respectively, reducing the disadvantages of both approaches.

Based on the Chi-Dep algorithm, an Ensemble of Chi-Dep classifiers (CDE) was proposed in [43]. CDE first randomly generates a large number (e.g., 10000) of possible label sets partitions. Then, a score for each partition is computed based on the χ^2 score for all label pairs in the partition. Finally, CDE selects the n distinct top scored partitions, generating a Chi-Dep algorithm with each partition. For the classification of a new instance, a voting process with a threshold t is used to calculate the final prediction.

The diversity in CDE is generated by selecting a different partition on each classifier.

3.3. Ensembles based on Predictive Clustering Trees: Random Forest of Predictive Clustering Trees (RF-PCT)

Predictive Clustering Trees (PCTs) [19] are decision trees that can be viewed as a hierarchy of clusters in such a way that the intra-cluster variation is minimized. The root node of PCT contains all data, and it is recursively partitioned into smaller clusters in children nodes. In order to construct a PCT in MLC, the distance between two instances is usually computed as the sum of Gini Indices [44] of the labels.

The Random Forest of Predictive Clustering Trees (RF-PCT) [45] generates an ensemble which uses PCTs as base classifiers. As random forest [46], each base classifier of RF-PCT uses a different set of instances sampled by bagging, and also selects at each node of the tree the best feature from a random subset of the attributes. This double random selection over the instances and the features provides diversity to the base classifiers of the ensemble.

For the prediction of a new instance, it averages the confidence values of all base classifiers for each label, and uses a threshold t to determine if the label is relevant or not.

3.4. Ensembles independent of base classifiers: Clustering-Based for Multi-Label Classification (CBMLC)

The previously described EMLCs were designed based on a specific multi-label method. However, there also exist EMLCs which are completely independent of the multi-label classifier used.

Clustering-Based method for Multi-Label Classification (CBMLC) [47] has two steps. In the first step, CBMLC groups the training data into c clusters using a clustering algorithm and only considers the features (not the labels). It is expected that similar objects are associated with similar labels, which results in a reduced label space in each of the classifiers. This may improve the predictive performance of each classifier, as well as to reduce the training and testing time. In the second step, it uses the multi-label algorithm to build a classifier over the data of each cluster, producing c multi-label classifiers. For the classification of an unknown instance, CBMLC first finds the cluster closest to the instance and then uses the corresponding classifier to classify it. LP was used as multi-label classifier and k-means [48] was used as clustering algorithm.

CBMLC obtains diverse classifiers by the selection of instances and also labels in each cluster.

3.5. Taxonomy of EMLCs

While overviewing state-of-the-art EMLCs we found some points by which to group or categorize them. As no taxonomy in the literature covers the characteristics of the EMLCs, we propose the following taxonomy for EMLCs, as shown in Table 1. First, the EMLCs can be categorized based on the multi-label method they are based, such as BR, LP, PCT or independent. Second, each EMLC can be categorized based on the way it generates diversity in the ensemble. Based on the taxonomy proposed in [49], we identified four ways or levels to generate diversity in EMLCs:

- Classifier level: at this level, different algorithms are used in the ensemble. The difference among classifiers can be given in several ways such as the use of different algorithms or the use of different parameters in the same algorithm. ECC and CDE are categorized at this level.

Table 1: EMLC methods. State-of-the-art ensemble of MLC methods are categorized depending on the method they are based (BR, LP, PCT or independent) and the way the diversity of the ensemble is obtained as A (*classifier level*), B (*label level*), C (*feature level*) or D (*data level*).

Abbreviation	**Method name**	**Level**				**Reference**	**Year**
		A	**B**	**C**	**D**		
	Ensembles based on BR						
EBR	Ensemble of Binary Relevance classifiers (bagging)				•	[26]	2011
ECC	Ensemble of Classifier Chains	•			•	[26]	2011
$\text{MLS}_{\text{train}}$	Multi-Label Stacking using train data for meta-level			•		[29]	2009
MLS_{cv}	Multi-Label Stacking using cv for meta-level			•		[29]	2009
MLS_{ϕ}	Multi-Label Stacking pruning meta-level			•		[29]	2009
HOMER	Hierarchy Of Multi-label classifiERs		•		•	[31]	2008
AdaBoost.MH	AdaBoost.MH		•		•	[36]	2000
	Ensembles based on LP						
ELP	Ensemble of Label Powerset classifiers (bagging)				•	-	-
EPS	Ensemble of Pruned Sets				•	[37]	2008
RA*k*EL	Random *k*-labELsets		•			[38]	2011
RA*k*EL++	Random *k*-labELsets using confidences		•			[39]	2014
RA*k*ELd	Random *k*-labELsets with disjoint labelsets		•			[38]	2011
TREMLC	Triple Random Ensemble for MLC		•	•	•	[40]	2010
CDE	Chi-Dep Ensemble	•				[43]	2010
	Ensembles based on PCT						
RF-PCT	Random Forest of Predictive Clustering Trees			•	•	[45]	2007
	Ensembles independent of the multi-label classifier						
CBMLC	Clustering-Based for Multi-Label Classification		•		•	[47]	2009

- Label level: at this level, each classifier of the ensemble is built over a different subset of labels. It usually implies a reduction in complexity of each classifier and also increase the diversity of classifiers in the ensemble. HOMER, AdaBoost.MH, RA*k*EL, TREMLC and CBMLC are categorized at this level.

- Feature level: at this level, each classifier of the ensemble uses a different subset of the features of the original training set, either disjoint or overlapping. This make each classifier focus on a subset of the input features, increasing the diversity and accuracy of the ensemble. MLS, TREMLC and RF-PCT are categorized at this level.

- Data level: at this level, each classifier is built over a different subset of the training dataset, either with or without replacement. It is a simple, effective and widely used method to generate a diverse ensemble [46]. EBR, ECC, HOMER, AdaBoost.MH, ELP, EPS, TREMLC, RF-PCT and CBMLC are categorized at this level.

These levels are not mutually exclusive, so each one of the EMLCs may appear in several groups simultaneously, i.e., an EMLC could be created by training each classifier of the ensemble over a subset of the labels and also over a subset of the features, belonging to both *label* and *feature levels*.

4. Experimental design

As mentioned in Section 1, the objective of this study is to perform an experimental comparison and analysis of the state-of-the-art EMLCs. In this section, first the chosen evaluation metrics and datasets are shown, then the default parameters of the EMLCs are presented and, finally, the experimental setup is explained.

4.1. Evaluation metrics

Evaluation metrics for multi-label classification are commonly distinguished in two groups: example-based metrics such as Hamming loss, accuracy or $\text{FMeasure}_{\text{ex}}$ which are calculated for each instance, and label-based metrics such as $\text{FMeasure}_{\text{mac}}$ and $\text{FMeasure}_{\text{mic}}$ which are calculated with respect to labels. The formulation of the metrics used in this study are shown in Table 2, being Y the true labels, Z the predicted labels, m_{test} the number of instances of the test dataset and Δ computes the symmetric difference between two sets. In addition, tp, fp and fn refer to true positives, false positives and false negative of the contingency table respectively. A wider description of these evaluation metrics can be found in [9].

Table 2: Multi-label evaluation metrics used in this study.

↓ Hamming loss	$\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\frac{1}{q}\lvert Z_i\Delta Y_i\rvert$
↑ Accuracy	$\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\frac{\lvert Z_i\cap Y_i\rvert}{\lvert Z_i\cup Y_i\rvert}$
↑ $\text{FMeasure}_{\text{ex}}$	$\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\frac{2\lvert Z_i\cap Y_i\rvert}{\lvert Z_i\rvert+\lvert Y_i\rvert}$
↑ $\text{FMeasure}_{\text{mac}}$	$\frac{1}{q}\sum_{i=1}^{q}\frac{2\cdot tp_i}{2\cdot tp_i+fn_i+fp_i}$
↑ $\text{FMeasure}_{\text{mic}}$	$\frac{\sum_{i=1}^{q}2\cdot tp_i}{\sum_{i=1}^{q}2\cdot tp_i+\sum_{i=1}^{q}fn_i+\sum_{i=1}^{q}fp_i}$

4.2. Datasets

Multi-label datasets have special characteristics that can be measured and identified by different characterization metrics. Density and diversity measure distribution of labels [50]. The density (*dens*) is defined as the mean number of relevant labels for each example divided by the total number of labels, and diversity (*div*) as the ratio of labelsets that appear in the dataset of the total of possible number of distinct labelsets. The *avgIR* measures the imbalance of the dataset by averaging the imbalance ratio of each label [51]. The greater the *avgIR* value, the greater the imbalance ratio of the labels of the dataset. Finally, the ratio of unconditionally dependent labels pairs by chi-square test (*rDep*) measures the relationship among labels. The *rDep* is defined as the number of pairs of labels which are dependent at 99% confidence level by chi-square test divided by the total number of label pairs [52]. The greater *rDep*, the greater the relationship between the labels.

A set of 20 datasets from 8 different domains and with different characteristics is used in this study. The datasets range from 194 to 43910 instances, from 16 to 1449 features, and from 5 to 101 labels. Half of the datasets have on average less than 10% of the labels associated with an instance, but there are also several that have up to 40%. The diversity ranges from 0.003 (only the 0.3% of the possible labelsets appear in the dataset) to 1 (which means that all the possible labelsets appears in the dataset). The *avgIR* ranges from values near to 1 to values of more than 250, showing a great variety in the imbalance of the datasets. Finally, there is also a great variety in the degree of relationship among labels, with values of *rDep* ranging from near to zero (non related datasets) to near to 1 (hihgly related datasets). Table 3 shows the datasets including their domain and the values of the previously defined characterization metrics; the datasets are ordered by dimensionality, defined as $m \times d \times q$ according to [53].

All datasets were downloaded from a new repository of multi-label datasets[1]. Furthermore, the characterization of the datasets was performed using the MLDA tool [66].

4.3. Methods and configurations

All the methods described in Section 3, and listed in Table 4, were run using the default parameters proposed by their authors. For RA*k*EL and RA*k*EL++, two different configurations were used, according to the recommendations of their authors. Unless otherwise specified, all the methods used $n = 10$ classifiers in the ensemble, a threshold value of $t = 0.5$ and the C4.5 decision tree (Weka's J48 [67]) as a single-label base classifier. It has been

[1] `http://www.uco.es/kdis/mllresources/`

Table 3: Multi-label datasets.

	Domain	***m***	***d***	***q***	***dens***	***div***	***avgIR***	***rDep***	**Ref**
Flags	Image	194	19	7	0.485	0.422	2.255	0.381	[54]
CHD_49	Medicine	555	49	6	0.430	0.531	5.766	0.267	[55]
Water-quality	Chemistry	1060	16	14	0.362	0.778	1.767	0.473	[56]
Emotions	Music	593	72	6	0.311	0.422	1.478	0.933	[31]
3s_reuters1000	Text	294	1000	6	0.188	0.219	1.789	0.667	[57]
3s_guardian1000	Text	302	1000	6	0.188	0.219	1.773	0.667	[57]
3s_bbc1000	Text	352	1000	6	0.188	0.234	1.718	0.733	[57]
Birds	Audio	645	260	19	0.053	0.206	5.407	0.123	[58]
Yeast	Biology	2417	103	14	0.303	0.082	7.197	0.670	[59]
Scene	Image	2407	294	6	0.179	0.234	1.254	0.933	[17]
PlantPseAAC	Biology	978	440	12	0.090	0.033	6.690	0.318	[60]
HumanPseAAC	Biology	3106	440	14	0.085	0.027	15.289	0.418	[60]
Genbase	Biology	662	1186	27	0.046	0.048	37.315	0.157	[61]
Yelp	Text	10810	671	5	0.328	1.000	2.876	0.700	[62]
Medical	Text	978	1449	45	0.028	0.096	89.501	0.039	[63]
Slashdot	Text	3782	1079	22	0.054	0.041	19.462	0.273	[53]
Enron	Text	1702	1001	53	0.064	0.442	73.953	0.141	[37]
Langlog	Text	1460	1004	75	0.016	0.208	39.267	0.035	[53]
20NG	Text	19300	1006	20	0.051	0.003	1.007	0.984	[64]
Mediamill	Video	43910	120	101	0.043	0.149	256.405	0.342	[65]

shown that ensemble learning works well when decision trees are used as the base classifier [46]. Although some EMLCs use other base classifiers in addition to C4.5 in their original papers, it has been used as base classifier in almost all the studied EMLCs to obtain a greater consistency in the results.

4.4. Experimental setup

All the experiments were implemented using Meka [68] and Mulan [69] frameworks. Meka and Mulan are open-source Java frameworks for learning from multi-label datasets, which include a wide variety of state-of-the-art algorithms and provide an API to use their functionalities in Java code. In addition, CLUS library [19] was used to execute the RF-PCT algorithm. In order to ensure that all metrics are calculated in the same way, all the algorithms were executed by Meka. For that, the Meka's wrapper for Mulan algorithms and the Mulan's wrapper for CLUS algorithms have been used.

All the algorithms were executed over a stratified 5-folds cross-validation partitioning of the full dataset, using the Iterative Stratification method [70] to guarantee the distribution of labels in the partitions is as similar as possible. For algorithms which use random numbers (such as EBR, ECC, ELP, EPS, RA*k*EL, RA*k*EL++, RA*k*ELd, TREMLC, CDE, RF-PCT and

Table 4: Algorithms and default parameters proposed by their authors.

Algorithm	Parameters
EBR	$bagSizePercent = 100$
ECC	$bagSizePercent = 100$
$\mathbf{MLS_{train}}$	-
$\mathbf{MLS_{cv}}$	$numFolds = 10$
$\mathbf{MLS}_{\phi}$	$numFolds = 10$, $phiThreshold = 0.15$
HOMER	$clusteringAlgorithm =$ balanced k-means, $c = 3$
AdaBoost.MH	$baseLearner =$ DecisionStump
ELP	$bagSizePercent = 100$
EPS	$bagSizePercent = 67$, $strategy = A$, $p = 1$, $b = 1$
RA*k*EL1	$k = q/2$, $n = 10$
RA*k*EL2	$k = 3$, $n = 2q$
RA*k*EL++1	$k = q/2$, $n = 10$
RA*k*EL++2	$k = 3$, $n = 2q$
RA*k*ELd	$k = 3$
TREMLC	$k = 3$, $n = 2q$, $bagSizePercent = 70$, $featurePercentage = 51$
CDE	$randomPartitions = 10000$
RF-PCT	$bagSizePercent = 100$
CBMLC	$clusteringAlgorithm = k$-means, $c = 5$

CBMLC) 10 different seeds were used.

Since three versions of MLS and five versions of RA*k*EL are available, first comparisons among MLS and RA*k*EL methods are performed separately to determine which version of each algorithm is the best. Then, the best variants of MLS and RA*k*EL are used in the complete study.

Next, several experiments to compare different EMLCs based on their characteristics are performed. First, the performance of the EMLCs is evaluated given their imbalance ratio. Second, the EMLCs are evaluated taking into account the degree of dependency among labels. Then, the EMLCs are evaluated in terms of efficiency. Finally, an overall comparison of all EMLCs over all evaluation metrics is performed.

In order to compare the performance of different EMLCs, the Skillings-Mack's test [71, 72] was performed for each metric. Skillings-Mack's test is similar to Friedman's test [73] but it can be used with missing values. The cases marked as DNF in the results are treated as missing values for the tests, except for training and testing times, where they are assigned the worst ranking value. In cases where the Skillings-Mack's test indicates that there exist significant differences in the performance of the algorithms, the Shaffer's post-hoc test [74] was used to perform multiple comparisons among all the methods. The use of Shaffer's test was proposed in [75], to the detriment of other tests, such as Nemenyi's [76]. Furthermore, the adjusted p-values [77] were considered in the analysis. The adjusted p-values provide more statistical information since they take into account the fact that multiple comparisons can be directly performed with any significance level. In this work, a significance level of $\alpha = 0.05$ was used.

5. Results and discussion

In this section the experimental results of the EMLCs are presented and discussed. First, results of the comparisons among variants of MLS and RA*k*EL are presented. Then, the results of each of the experiments are presented and discussed, including some tips how to select the best EMLC according to the characteristics of the dataset.

Datasets and detailed results for all experiments are fully described and publicly available to facilitate the replicability of the experiments and future comparisons at the KDIS Research Group website[2]. For the largest datasets, some algorithms did not finish the execution of a single fold within a day using the available resources[3]. These cases are marked as DNF (Did Not Finish) in the result tables.

5.1. Comparison among MLS and RAkEL variants

As previously mentioned, there are three variants of MLS and five variants of RA*k*EL. In order to simplify further study, first a comparison among those method is performed. The results in Table 5 show the Skillings-Mack test value, adjusted p-value and average ranking of each MLS variant and metric over all datasets. The best algorithm for each metric is marked in bold. Although there are no significant differences among MLS variants, $\text{MLS}_{\text{train}}$ is the best algorithm in four of the five metrics.

Table 6 shows the results of the Skillings-Mack and Shaffer tests of each RA*k*EL variant. For each metric, the best algorithm is marked in bold and those algorithms which have significant differences with the best algorithm

[2] `http://www.uco.es/kdis/emlcreview/`

[3] The experiments have been performed on a machine with Debian 8, two Intel Core i7 CPUs at 2.67 GHz and 16 GB of RAM.

Table 5: Skillings-Mack test results for the comparison among MLS variants.

Metric	Statistic	*p-value*	Rankings		
			$\mathbf{MLS_{train}}$	$\mathbf{MLS_{cv}}$	$\mathbf{MLS_{\phi}}$
Hamming loss	0.925	0.630	1.93	2.18	**1.90**
Accuracy	1.900	0.387	**1.75**	2.15	2.10
$\text{FMeasure}_{\text{ex}}$	1.875	0.392	**1.75**	2.13	2.13
$\text{FMeasure}_{\text{mac}}$	2.800	0.247	**1.80**	1.90	2.30
$\text{FMeasure}_{\text{mic}}$	1.575	0.455	**1.78**	2.08	2.15

are marked with •. As seen, RA*k*EL2 is the best algorithm in four metrics, also being the only one that is not significantly different from the best algorithm in any case.

Table 6: Skillings-Mack test results for the comparison among RA*k*EL variants.

Metric	Statistic	*p-value*	Rankings				
			RA*k*EL1	RA*k*EL2	RA*k*EL++1	RA*k*EL++2	RA*k*ELd
Hamming loss	37.510	1.41E-07	• 3.05	2.55	• 3.08	**1.68**	• 4.65
Accuracy	24.090	7.66E-05	2.30	**1.88**	• 3.73	• 3.30	• 3.80
$\text{FMeasure}_{\text{ex}}$	23.930	8.25E-05	2.20	**1.93**	• 3.75	• 3.58	• 3.55
$\text{FMeasure}_{\text{mac}}$	21.830	2.17E-04	2.38	**1.98**	• 3.85	• 3.75	3.05
$\text{FMeasure}_{\text{mic}}$	29.510	6.16E-06	2.30	**1.70**	• 3.83	• 3.33	• 3.85

Based on these results, in further experiments only $\text{MLS}_{\text{train}}$ and RA*k*EL2 are used.

5.2. Experiment 1: results depending on the imbalance of the datastes

In multi-label classification, one of the main problems is to deal with the imbalance of the data. Usually, some labels are very frequent while other are barely present in the dataset. This feature can have direct impact on performance of the algorithms, so we studied the performance of the EMLCs according to the imbalance of the dataset. Sorting the datasets by *avgIR*, we separate them into little, moderately, and very imbalanced datasets. Little imbalanced datasets are those with a $avgIR < 2$, moderately imbalanced are those with $2 \leq avgIR < 20$ and, any dataset with $avgIR \geq 20$ is considered as very imbalanced.

$\text{FMeasure}_{\text{mic}}$ and $\text{FMeasure}_{\text{mac}}$ measure the performance of the algorithms over imbalanced data from two different points of view. While the former is biased by the frequency of occurrence of each label, the latter does not, giving equal importance to all labels independently of their frequency. $\text{FMeasure}_{\text{mac}}$ is more useful than $\text{FMeasure}_{\text{mic}}$ if infrequent or more imbalanced labels are present in evaluation of the classifier. Tables 7 and 8 show the datasets ordered by *avgIR* along with their results for FMeasure$_{\text{mac}}$ and FMeasure$_{\text{mic}}$, respectively. Further, these tables include the average rankings of each algorithm for little, moderately and very imbalanced datasets calculated as the Skillings-Mack's test.

As seen for FMeasure$_{\text{mac}}$, ELP is the best algorithm, on average, in little imbalanced datasets, CDE is the best on average for moderately imbalanced datasets and RAkEL2 is the best on average for very imbalanced datasets. The fact that both CDE and RAkEL2 split the output space into smaller labelsets for each base classifier causes that in cases of high degree of imbalance each base classifier has a more even distribution of labels than if all labels are taken into account at the same time, as in ELP. For very imbalanced datasets CDE is the best in the only two datasets where it finished, but due to its high complexity its average ranking deteriorates. The results for FMeasure$_{\text{mic}}$ are similar to those of FMeasure$_{\text{mac}}$ but in this case, for moderately imbalanced datasets ECC is the algorithm that performs better. Since FMeasure$_{\text{mic}}$ assigns more importance in the evaluation to more frequent labels if ECC predicts well these frequent labels; the performance according to this metric is higher. However it is common to try to predict correctly rare labels. Therefore, for a little imbalanced dataset, ELP is the best option, while for moderately and very imbalanced datasets RAkEL2 is the best one if all labels are considered to be equally important. If the high complexity of CDE does not matter, it also achieves good results for moderately and very imbalanced datasets.

Table 7: Results for $\text{FMeasure}_{\text{mac}}$ ↑ for all the EMLCs and datasets ordered by *avgIR*, including the average ranking ↓ of each algorithm for little, moderately and very imbalanced datasets.

	EBR	**ECC**	**$\mathbf{MLS_{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
20NG	0.650	0.671	0.622	0.609	0.000	**0.686**	**0.686**	0.656	0.574	DNF	0.399	0.328
Scene	0.706	**0.729**	0.647	0.586	0.000	0.704	0.703	0.701	0.700	0.720	0.711	0.598
Emotions	0.633	0.650	0.592	0.564	0.059	0.642	0.637	0.633	0.616	0.637	**0.653**	0.547
3s_bbc1000	0.083	0.123	0.158	0.230	0.000	0.204	0.174	0.195	0.150	0.200	0.051	**0.281**
Water-quality	0.501	0.525	0.465	0.520	0.082	0.466	0.278	0.523	0.508	0.508	**0.546**	0.484
3s_guardian1000	0.061	0.096	0.193	0.212	0.000	0.160	0.150	0.167	0.130	0.144	0.045	**0.294**
3s_reuters1000	0.076	0.104	0.160	0.196	0.000	0.170	0.148	0.185	0.131	0.164	0.058	**0.247**
	7.50	5.00	7.14	5.43	11.86	**4.36**	6.00	4.64	7.64	4.86	6.71	6.00
Flags	0.663	0.683	0.620	0.630	0.562	0.674	0.655	0.684	0.608	**0.689**	0.685	0.597
Yelp	0.710	0.721	0.683	0.675	0.000	0.706	0.706	**0.724**	0.647	**0.724**	0.614	0.600
Birds	0.230	0.239	0.234	**0.270**	0.000	0.250	0.207	0.251	0.191	0.260	0.230	0.179
CHD_49	0.497	**0.524**	0.464	0.492	0.270	0.511	0.511	0.517	0.505	0.516	0.520	0.505
PlantPseAAC	0.081	0.097	**0.160**	0.143	0.000	0.063	0.065	0.117	0.107	0.130	0.059	0.156
Yeast	0.387	0.401	0.395	0.403	0.122	0.380	0.375	0.409	0.389	**0.410**	0.396	0.396
HumanPseAAC	0.091	0.107	**0.150**	0.129	0.000	0.082	0.080	0.133	0.112	0.133	0.073	0.143
Slashdot	0.235	0.248	0.242	0.253	0.000	**0.301**	0.296	0.249	0.154	DNF	0.178	0.150
	7.31	4.50	6.00	5.13	11.88	6.25	7.38	3.13	8.19	**2.88**	7.13	7.50
Genbase	0.738	0.743	**0.747**	0.744	0.000	0.721	0.676	0.744	0.619	**0.747**	0.001	0.725
Langlog	0.032	0.039	0.051	**0.056**	0.000	0.017	0.025	0.045	0.031	DNF	0.005	0.037
Enron	0.153	0.158	0.152	**0.186**	0.013	0.121	0.116	0.164	0.131	DNF	0.111	0.104
Medical	0.352	0.360	0.371	0.343	0.000	0.338	0.327	**0.372**	0.274	**0.372**	0.026	0.178
Mediamill	0.187	0.179	0.211	0.175	0.009	DNF	0.164	**0.233**	0.033	DNF	0.200	0.074
	5.00	4.20	2.70	3.50	11.20	7.30	8.00	**2.20**	8.20	4.10	8.80	8.00

Table 8: Results for FMeasure$_{\text{mic}}$ ↑ for all the EMLCs and datasets ordered by *avgIR*, including the average ranking ↓ of each algorithm for little, moderately and very imbalanced datasets.

	EBR	**ECC**	**MLS$_{\text{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
20NG	0.663	0.681	0.630	0.610	0.000	**0.692**	**0.692**	0.664	0.592	DNF	0.424	0.330
Scene	0.702	**0.722**	0.638	0.576	0.000	0.697	0.696	0.693	0.692	0.714	0.702	0.591
Emotions	0.653	0.666	0.599	0.572	0.105	0.660	0.654	0.648	0.628	0.652	**0.671**	0.557
3s_bbc1000	0.112	0.169	0.192	0.246	0.000	0.233	0.208	0.230	0.192	0.238	0.075	**0.294**
Water-quality	0.563	0.582	0.516	0.568	0.249	0.535	0.413	0.573	0.565	0.575	**0.595**	0.491
3s_guardian1000	0.094	0.144	0.240	0.226	0.000	0.213	0.203	0.211	0.179	0.181	0.076	**0.309**
3s_reuters1000	0.112	0.156	0.198	0.213	0.000	0.218	0.193	0.222	0.177	0.200	0.099	**0.264**
	7.21	5.00	6.93	6.00	11.86	**4.07**	5.93	4.86	7.64	4.57	6.79	6.29
Flags	0.753	0.760	0.720	0.736	0.711	0.749	0.745	0.761	0.745	**0.765**	0.771	0.651
Yelp	0.749	0.759	0.721	0.708	0.000	0.739	0.739	**0.754**	0.672	**0.759**	0.684	0.631
Birds	0.418	0.440	0.375	**0.391**	0.000	0.433	0.413	0.422	0.360	0.432	0.410	0.201
CHD_49	0.654	**0.677**	0.604	0.644	0.598	0.667	0.668	0.665	0.665	0.665	0.676	0.590
PlantPseAAC	0.161	0.205	**0.239**	0.203	0.000	0.148	0.148	0.218	0.200	0.220	0.160	0.204
Yeast	0.626	0.637	0.548	0.585	0.480	0.626	0.625	0.621	0.609	**0.631**	0.636	0.493
HumanPseAAC	0.246	0.292	**0.300**	0.270	0.000	0.243	0.240	0.316	0.266	0.312	0.248	0.206
Slashdot	0.464	0.476	0.473	0.457	0.000	**0.508**	0.513	0.480	0.349	DNF	0.394	0.179
	6.06	**2.56**	6.88	7.63	11.63	5.44	6.19	3.75	7.94	3.19	5.63	10.38
Genbase	0.989	0.988	**0.989**	0.987	0.000	0.979	0.977	0.989	0.903	**0.989**	0.000	0.978
Langlog	0.163	0.189	0.192	**0.150**	0.000	0.101	0.140	0.190	0.147	DNF	0.029	0.040
Enron	0.573	0.587	0.522	**0.526**	0.245	0.512	0.507	0.574	0.558	DNF	0.526	0.126
Medical	0.813	0.816	0.812	0.793	0.000	0.785	0.783	**0.813**	0.724	**0.815**	0.180	0.323
Mediamill	0.617	0.616	0.555	0.549	0.287	DNF	0.600	**0.618**	0.300	DNF	0.621	0.110
	3.20	2.80	4.30	5.90	10.70	8.30	7.60	**2.40**	7.40	4.40	7.80	9.60

5.3. Experiment 2: results depending on the relationship among labels

Another main challenge in multi-label classification is how to deal with the relationship among labels. The labels might be more or less correlated, and taking into account these correlations when learning a model could improve the performance. Sorting the datasets by $rDep$, we separated them into three groups. Those with $rDep < 0.3$ are considered as low dependent datasets, those with $0.3 \leq rDep < 0.7$ are considered as medium dependent datasets and those with $rDep \geq 0.7$ are considered as highly dependent datasets.

FMeasure$_{\text{ex}}$ and Accuracy measure the multi-label prediction of each instance as a whole, which makes it useful for measuring the performance based on the relationship between labels. Tables 9 and 10 show the datasets ordered by $rDep$ and their FMeasure$_{\text{ex}}$ and Accuracy results, respectively. The tables also include the average rankings for low, medium and highly dependent datasets.

For FMeasure$_{\text{ex}}$, ECC is the best algorithm on average, while for Accuracy RAkEL2 is the best in both low and medium dependent datasets. In both metrics ECC, RAkEL2 and CDE are the top three algorithms for low and medium datasets. These three methods take into account the relationship among labels but in a softer way than other LP-based algorithms as ELP, so in cases where the dependency among labels is not very high, ECC and RAkEL2 are the best options. However, for highly dependent datasets ELP is the best algorithm on average. ELP considers all labels at a time, so the relationship among labels can be exploited more exhaustively than if labels are treated more independently.

Table 9: Results for FMeasure$_{\text{ex}}$ ↑ for all the EMLCs and datasets ordered by *rDep*, including the average ranking ↓ of each algorithm for low, medium and high dependent datasets.

	EBR	**ECC**	**MLS$_{\text{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
Langlog	0.098	0.117	**0.130**	0.108	0.000	0.059	0.083	0.123	0.088	DNF	0.016	0.089
Medical	0.782	**0.795**	0.787	0.775	0.000	0.780	0.777	0.788	0.659	0.792	0.110	0.714
Birds	0.069	0.119	0.139	0.205	0.000	0.171	0.153	0.174	0.135	0.176	0.046	**0.283**
Enron	0.553	**0.581**	0.496	0.528	0.231	0.505	0.499	0.556	0.539	DNF	0.511	0.266
Genbase	0.172	0.222	**0.268**	0.224	0.000	0.170	0.168	0.254	0.201	0.245	0.174	0.259
CHD_49	0.623	**0.650**	0.565	0.615	0.580	0.640	0.642	0.640	0.638	0.639	0.650	0.561
Slashdot	0.363	0.391	0.375	0.392	0.000	0.447	**0.452**	0.378	0.251	DNF	0.289	0.430
	6.86	3.79	5.71	5.29	11.29	6.21	6.29	**3.50**	7.57	4.71	8.07	6.14
PlantPseAAC	0.522	**0.544**	0.513	0.506	0.000	0.514	0.514	0.539	0.455	0.532	0.662	0.499
Mediamill	**0.587**	**0.588**	0.529	0.527	0.297	DNF	0.574	0.589	0.282	DNF	**0.594**	0.319
Flags	0.722	0.734	0.685	0.713	0.660	0.721	0.717	0.734	0.728	0.741	**0.754**	0.633
HumanPseAAC	0.102	0.145	0.204	0.153	0.000	0.096	0.095	0.163	0.144	0.159	0.106	**0.240**
Water-quality	0.531	0.553	0.481	0.539	0.232	0.507	0.397	0.543	0.534	0.547	**0.568**	0.458
3s_guardian1000	0.597	**0.613**	0.524	0.559	0.456	0.600	0.599	0.599	0.586	0.606	0.616	0.496
3s_reuters1000	0.170	**0.182**	0.159	0.182	0.000	0.170	0.166	0.179	0.147	0.184	0.160	0.158
Yeast	0.057	0.098	0.187	0.190	0.000	0.154	0.145	0.156	0.126	0.124	0.047	**0.291**
	6.69	3.75	7.13	6.06	11.50	6.44	7.50	**3.63**	8.13	3.81	4.00	7.88
Yelp	0.991	0.990	**0.991**	0.988	0.000	0.986	0.985	0.991	0.830	0.991	0.000	0.986
3s_bbc1000	0.616	**0.675**	0.595	0.524	0.000	0.650	0.649	0.646	0.637	0.663	0.655	0.616
Scene	0.068	0.107	0.154	0.169	0.000	0.159	0.140	0.172	0.122	0.135	0.060	**0.226**
Emotions	0.594	0.621	0.554	0.523	0.061	0.615	0.609	0.608	0.583	0.606	**0.634**	0.539
20NG	0.540	0.580	0.521	0.547	0.000	**0.627**	**0.627**	0.556	0.451	DNF	0.277	0.587
	7.00	4.20	6.90	7.40	11.70	**4.00**	5.10	4.10	8.40	4.70	7.30	6.00

Table 10: Results for Accuracy ↑ for all the EMLCs and datasets ordered by *rDep*, including the average ranking ↓ of each algorithm for low, medium and high dependent datasets.

	EBR	**ECC**	**MLS$_{\text{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
Langlog	0.232	0.249	**0.256**	0.233	0.142	0.198	0.220	0.252	0.222	DNF	0.157	0.219
Medical	0.752	**0.765**	0.756	0.745	0.000	0.756	0.753	0.757	0.627	0.761	0.101	0.692
Birds	0.065	0.113	0.126	0.175	0.000	0.162	0.146	0.160	0.126	0.165	0.045	**0.245**
Enron	0.442	**0.471**	0.390	0.411	0.150	0.399	0.397	0.443	0.429	DNF	0.405	0.209
Genbase	0.163	0.209	**0.234**	0.189	0.000	0.160	0.158	0.231	0.185	0.224	0.162	0.228
CHD_49	0.513	**0.540**	0.457	0.495	0.464	0.529	0.530	0.527	0.523	0.529	0.535	0.440
Slashdot	0.348	0.375	0.359	0.370	0.000	0.431	**0.436**	0.362	0.243	DNF	0.281	0.404
	7.00	**3.43**	5.86	5.57	11.29	5.86	6.00	3.71	7.36	4.50	8.29	6.57
PlantPseAAC	0.725	**0.739**	0.694	0.684	0.250	0.723	0.723	0.737	0.636	0.737	0.577	0.659
Mediamill	**0.489**	**0.489**	0.424	0.419	0.192	DNF	0.476	0.486	0.183	DNF	**0.489**	0.267
Flags	0.615	0.631	0.575	0.598	0.541	0.617	0.615	0.633	0.617	0.637	**0.644**	0.521
HumanPseAAC	0.099	0.142	0.180	0.129	0.000	0.094	0.094	0.153	0.136	0.150	0.104	**0.221**
Water-quality	0.392	0.411	0.347	0.394	0.151	0.369	0.281	0.401	0.392	0.405	**0.424**	0.320
3s_guardian1000	0.486	**0.506**	0.407	0.439	0.335	0.493	0.492	0.482	0.469	0.495	0.505	0.386
3s_reuters1000	0.596	**0.603**	0.544	0.555	0.456	0.594	0.589	0.586	0.563	0.591	0.579	0.501
Yeast	0.054	0.091	0.168	0.168	0.000	0.144	0.137	0.142	0.116	0.111	0.045	**0.248**
	5.88	**3.13**	7.06	7.06	11.50	5.75	6.94	4.31	7.63	4.00	5.38	7.88
Yelp	0.987	0.986	**0.988**	0.984	0.000	0.982	0.980	0.987	0.822	0.987	0.000	0.982
3s_bbc1000	0.600	**0.660**	0.562	0.489	0.000	0.636	0.636	0.624	0.617	0.641	0.639	0.599
Scene	0.062	0.098	0.139	0.144	0.000	0.149	0.132	0.157	0.112	0.120	0.057	**0.189**
Emotions	0.513	0.539	0.461	0.426	0.045	0.536	0.529	0.520	0.499	0.522	**0.548**	0.455
20NG	0.529	0.570	0.506	0.522	0.000	**0.623**	**0.623**	0.541	0.443	DNF	0.276	0.579
	6.80	4.20	6.60	7.80	11.70	**3.90**	5.00	4.40	8.40	4.60	7.30	6.10

5.4. Experiment 3: efficiency

Several EMLCs are computationally demanding, to the point that many of them did not build a single model within one day of computing. Thus, the efficiency of the EMLCs is a factor to be taken into account.

Sorting the datasets by dimensionality (defined as $m \times q \times d$), we separate them into three groups: small, medium and large datasets. We considere as small datasets those with $\mathit{dimensionality} < 1\text{E}6$, as medium datasets those with $\mathit{dimensionality} \in [1\text{E}6, 1\text{E}8)$, and as large datasets those with $\mathit{dimensionality} \geq 1\text{E}8$. In tables 11 and 12 are shown the train and test times, respectively, for all the EMLCs and all the datasets. There are algorithms that are very fast but whose performance is bad, so not only the execution times but also the Hamming loss is shown in Table 13.

CBMLC generates several classifiers with a subset of similar instances and therefore possibly also with a subset of similar labels, which leads to less complex classifiers, being the fastest algorithm for small datasets and the second for medium and large datasets in both training and test times. However, CBMLC is one of the worst algorithms in terms of Hamming loss, regardless of the dimensionality of the dataset, being a fast algorithm but with a very low performance. Also for small datasets EPS is one of the fastest algorithms, getting also a good performance in terms of Hamming loss, so it is the best option for small datasets. On the other hand, for medium and large datasets, RF-PCT is the most efficient algorithm in both train and test. For larger datasets, RF-PCT, which reduces considerably the selection of the attributes in each node of the tree, has a lower complexity than other EMLCs and therefore higher efficiency. In terms of Hamming loss, RF-PCT results are not bad, so it is a great option for medium and large datasets if a very fast but not best in prediction algorithm is needed. Also for medium datasets EPS is one of the best algorithms in Hamming loss, being the best for large datasets. This fact, coupled with acceptable execution time, makes EPS the best option considering both execution time and performance, regardless of the dimensionality of the dataset. CDE did not finish with any of the large datasets, so it has not been assigned any average ranking value.

Table 11: Results for training time ↓ for all the EMLCs and datasets ordered by dimensionality, including the average ranking ↓ of each algorithm for small, medium and large datasets.

	EBR	**ECC**	**MLS$_{\textbf{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
Flags	5.20E-01	5.27E-01	5.57E-01	4.47E-01	3.70E-01	**2.92E-01**	4.10E-01	5.57E-01	5.16E-01	2.62E+00	6.79E-01	4.01E-01
CHD_49	1.10E+00	1.07E+00	1.31E+00	8.25E-01	6.94E-01	7.02E-01	7.38E-01	1.06E+00	8.75E-01	9.61E+00	8.64E-01	**6.39E-01**
Water-quality	2.19E+00	2.29E+00	2.07E+00	1.36E+00	1.18E+00	1.28E+00	7.97E-01	1.74E+00	1.38E+00	4.93E+01	1.67E+00	**7.25E-01**
Emotions	1.54E+00	1.46E+00	1.60E+00	9.50E-01	1.00E+00	1.11E+00	9.46E-01	1.40E+00	1.18E+00	2.06E+01	1.03E+00	**7.43E-01**
	9.25	9.25	10.13	4.50	2.75	3.50	3.00	8.38	6.50	12.00	7.25	**1.50**
3s_reuters1000	6.97E+00	7.07E+00	4.68E+00	3.39E+00	4.04E+00	7.67E+00	7.35E+00	1.01E+01	3.38E+00	1.04E+02	**1.09E+00**	3.41E+00
3s_guardian1000	7.18E+00	7.32E+00	4.72E+00	3.31E+00	4.14E+00	7.87E+00	7.60E+00	1.06E+01	3.63E+00	1.06E+02	**1.12E+00**	3.37E+00
3s_bbc1000	9.52E+00	8.58E+00	6.08E+00	4.09E+00	4.39E+00	1.07E+01	9.82E+00	1.35E+01	4.27E+00	2.06E+02	**1.17E+00**	4.53E+00
Birds	5.67E+00	5.90E+00	4.68E+00	1.84E+00	3.60E+00	2.20E+00	1.63E+00	7.67E+00	3.83E+00	4.15E+02	1.77E+00	**1.27E+00**
Yeast	2.85E+01	1.85E+01	1.46E+01	4.19E+00	5.06E+00	9.49E+00	7.39E+00	1.99E+01	8.62E+00	6.96E+02	5.19E+00	**2.74E+00**
Scene	1.96E+01	1.85E+01	1.52E+01	4.53E+00	6.61E+00	1.00E+01	9.31E+00	1.37E+01	7.01E+00	3.14E+02	3.46E+00	**3.39E+00**
PlantPseAAC	3.89E+01	4.26E+01	3.24E+01	7.27E+00	2.07E+01	1.54E+01	1.67E+01	3.40E+01	1.44E+01	9.24E+02	**2.16E+00**	4.82E+00
HumanPseAAC	3.64E+02	3.94E+02	1.80E+02	4.70E+01	1.12E+02	5.53E+01	8.72E+01	2.57E+02	7.41E+01	6.59E+03	**5.71E+00**	2.31E+01
Genbase	9.25E+00	5.05E+00	4.69E+00	3.14E+00	8.05E+00	2.16E+00	1.94E+00	6.12E+00	3.39E+00	1.84E+03	**1.61E+00**	3.09E+00
Yelp	9.02E+02	7.78E+02	4.46E+02	1.78E+02	5.48E+01	2.53E+02	2.53E+02	4.93E+02	1.34E+02	8.57E+03	**7.15E+00**	4.84E+01
Medical	2.79E+01	3.45E+01	4.11E+01	8.02E+00	3.92E+01	1.60E+01	7.62E+00	3.32E+01	2.51E+01	2.88E+03	**2.34E+00**	2.75E+00
Slashdot	1.66E+03	1.77E+03	1.15E+03	1.11E+02	3.22E+02	8.14E+02	1.12E+03	1.13E+03	2.26E+02	DNF	**7.33E+00**	2.62E+02
Enron	7.99E+02	1.07E+03	6.07E+02	1.03E+02	4.89E+02	1.11E+02	1.19E+02	8.36E+02	2.25E+02	DNF	**4.67E+00**	1.83E+01
	9.31	9.46	7.85	3.15	5.85	6.35	5.81	9.54	4.69	12.00	**1.46**	2.54
Langlog	5.54E+02	7.56E+02	4.13E+02	5.21E+01	6.97E+02	6.05E+01	6.45E+01	5.48E+02	1.41E+02	DNF	**4.86E+00**	1.62E+01
20NG	2.13E+04	2.67E+04	1.22E+04	1.94E+03	2.16E+03	3.39E+03	3.33E+03	1.42E+04	3.49E+03	DNF	**2.80E+01**	5.64E+02
Mediamill	9.21E+03	1.38E+04	9.47E+03	**4.03E+02**	1.02E+03	DNF	2.61E+03	5.41E+03	1.27E+03	DNF	5.95E+02	4.67E+02
	9.00	10.67	8.00	2.33	6.00	7.17	5.33	8.00	6.00	11.83	**1.67**	2.00

Table 12: Results for test time ↓ for all the EMLCs and datasets ordered by dimensionality, including the average ranking ↓ of each algorithm for small, medium and large datasets.

	EBR	**ECC**	**MLS$_{\mathbf{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
Flags	1.85E-01	2.24E-01	1.60E-01	1.08E-01	8.74E-02	**8.30E-02**	1.24E-01	1.69E-01	2.40E-01	1.43E+00	2.91E-01	9.98E-02
CHD_49	4.85E-01	5.86E-01	5.26E-01	2.62E-01	2.90E-01	2.70E-01	2.83E-01	3.75E-01	5.59E-01	6.78E+00	3.36E-01	**2.09E-01**
Water-quality	1.66E+00	1.89E+00	1.16E+00	5.31E-01	3.87E-01	9.13E-01	3.94E-01	1.03E+00	1.60E+00	4.55E+01	6.66E-01	**3.60E-01**
Emotions	9.98E-01	1.02E+00	7.56E-01	3.43E-01	4.51E-01	5.61E-01	4.40E-01	7.67E-01	7.23E-01	1.69E+01	3.33E-01	**3.04E-01**
	9.00	10.50	7.75	3.25	3.50	4.00	4.00	7.50	9.00	12.00	6.00	**1.50**
3s_reuters1000	5.70E+00	5.70E+00	3.76E+00	1.60E+00	3.23E+00	5.57E+00	5.64E+00	8.21E+00	2.72E+00	8.79E+01	**2.81E-01**	2.03E+00
3s_guardian1000	5.87E+00	5.85E+00	3.46E+00	1.65E+00	3.32E+00	5.88E+00	5.56E+00	8.19E+00	2.95E+00	9.29E+01	**2.86E-01**	2.22E+00
3s_bbc1000	7.98E+00	7.26E+00	4.85E+00	2.29E+00	3.87E+00	8.62E+00	7.57E+00	1.13E+01	3.60E+00	2.61E+02	**2.83E-01**	2.92E+00
Birds	5.16E+00	5.41E+00	3.76E+00	1.00E+00	2.28E+00	1.71E+00	1.09E+00	6.92E+00	3.97E+00	3.98E+02	6.91E-01	**6.37E-01**
Yeast	2.77E+01	1.78E+01	1.35E+01	3.17E+00	3.67E+00	8.63E+00	6.77E+00	1.83E+01	1.36E+01	6.55E+02	**1.01E+00**	1.95E+00
Scene	1.83E+01	1.77E+01	1.34E+01	3.15E+00	5.11E+00	8.90E+00	8.43E+00	1.24E+01	8.77E+00	2.76E+02	**5.35E-01**	2.14E+00
PlantPseAAC	3.84E+01	4.18E+01	3.13E+01	6.25E+00	1.92E+01	1.48E+01	1.54E+01	3.22E+01	1.50E+01	8.58E+02	**5.86E-01**	3.78E+00
HumanPseAAC	3.64E+02	3.91E+02	1.78E+02	4.37E+01	1.11E+02	5.46E+01	8.48E+01	2.55E+02	8.71E+01	6.24E+03	**1.20E+00**	2.05E+01
Genbase	9.09E+00	4.91E+00	3.95E+00	2.02E+00	6.00E+00	1.47E+00	1.31E+00	4.58E+00	6.82E+00	1.83E+03	**1.11E+00**	1.58E+00
Yelp	9.02E+02	7.75E+02	4.46E+02	1.78E+02	5.38E+01	2.55E+02	2.54E+02	5.02E+02	2.60E+02	7.08E+03	**1.39E+00**	4.66E+01
Medical	2.73E+01	3.34E+01	4.11E+01	6.29E+00	4.29E+01	1.51E+01	7.23E+00	3.20E+01	3.88E+01	2.80E+03	2.91E+00	**1.92E+00**
Slashdot	1.66E+03	1.78E+03	1.10E+03	1.09E+02	3.19E+02	8.10E+02	1.12E+03	1.13E+03	2.67E+02	DNF	**2.75E+00**	2.59E+02
Enron	8.00E+02	1.08E+03	6.00E+02	1.01E+02	4.85E+02	1.12E+02	1.18E+02	8.35E+02	2.48E+02	DNF	**5.62E+00**	1.72E+01
	9.65	9.50	7.38	2.92	6.00	5.92	5.54	9.38	6.23	12.00	**1.15**	2.31
Langlog	5.57E+02	7.63E+02	4.14E+02	5.09E+01	7.01E+02	6.01E+01	6.38E+01	5.49E+02	1.68E+02	DNF	**8.13E+00**	1.51E+01
20NG	2.13E+04	2.67E+04	1.22E+04	1.96E+03	2.19E+03	3.39E+03	3.34E+03	1.42E+04	7.25E+03	DNF	**7.97E+00**	6.01E+02
Mediamill	9.47E+03	1.41E+04	9.38E+03	3.86E+02	1.00E+03	DNF	2.68E+03	5.44E+03	1.28E+03	DNF	**3.09E+02**	4.86E+02
	9.33	10.67	7.67	2.67	6.00	7.17	5.33	8.00	6.00	11.83	**1.00**	2.33

Table 13: Results for Hamming loss ↓ for all the EMLCs and datasets ordered by dimensionality, including the average ranking ↓ of each algorithm for small, medium and large datasets.

	EBR	**ECC**	**MLS$_{\text{train}}$**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RA*k*EL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
Flags	0.244	0.244	0.265	0.273	0.277	0.253	0.253	0.243	0.253	**0.237**	0.241	0.393
CHD_49	0.301	**0.295**	0.336	0.325	0.307	0.303	0.302	0.305	0.304	0.306	0.312	0.418
Water-quality	**0.292**	0.298	0.333	0.343	0.340	0.319	0.306	0.311	0.315	0.295	0.315	0.572
Emotions	**0.202**	0.204	0.249	0.269	0.302	0.208	0.210	0.221	0.226	0.214	0.209	0.315
	2.13	2.63	9.50	10.25	10.00	5.50	4.75	5.25	6.63	4.00	5.38	12.00
3s_reuters1000	0.210	0.222	0.251	0.301	**0.188**	0.219	0.218	0.238	0.225	0.219	0.201	0.405
3s_guardian1000	0.205	0.219	0.244	0.295	**0.188**	0.219	0.216	0.232	0.230	0.225	0.202	0.385
3s_bbc1000	0.202	0.215	0.246	0.293	**0.188**	0.213	0.214	0.227	0.217	0.215	0.199	0.380
Birds	**0.043**	**0.043**	0.055	0.061	0.053	0.044	0.044	0.049	0.049	0.047	0.045	0.277
Yeast	**0.206**	0.211	0.284	0.259	0.232	0.214	0.213	0.226	0.228	0.216	0.220	0.393
Scene	0.093	**0.092**	0.130	0.151	0.179	0.100	0.100	0.104	0.102	0.097	0.099	0.157
PlantPseAAC	0.093	0.098	0.137	0.143	**0.090**	0.094	0.094	0.109	0.107	0.105	0.097	0.270
HumanPseAAC	**0.085**	0.088	0.119	0.126	**0.085**	0.087	0.087	0.097	0.094	0.095	0.090	0.290
Genbase	**0.001**	**0.001**	**0.001**	**0.001**	0.046	0.002	0.002	**0.001**	0.008	**0.001**	0.046	0.002
Yelp	0.085	0.084	0.098	0.105	0.182	0.088	0.088	0.086	0.108	**0.082**	0.163	0.148
Medical	**0.010**	**0.010**	**0.010**	0.011	0.028	0.011	0.012	**0.010**	0.014	**0.010**	0.026	0.089
Slashdot	**0.042**	0.043	0.043	0.049	0.054	0.043	**0.042**	**0.042**	0.046	DNF	0.043	0.319
Enron	**0.047**	0.048	0.053	0.061	0.062	0.052	0.051	0.048	0.048	DNF	0.050	0.644
	2.27	3.96	8.23	9.38	6.92	5.19	4.73	6.38	7.73	6.12	5.77	11.31
Langlog	0.016	0.016	0.019	0.026	0.016	0.016	**0.015**	0.018	0.016	DNF	0.016	0.463
20NG	**0.029**	**0.029**	0.033	0.039	0.051	**0.029**	**0.029**	0.030	0.033	DNF	0.039	0.144
Mediamill	**0.027**	0.028	0.037	0.038	0.038	DNF	0.029	0.029	0.065	DNF	0.029	0.477
	2.67	3.00	7.17	8.67	7.33	4.17	**2.50**	5.67	6.67	-	5.67	10.67

5.5. Experiment 4: general results

Once the results of the EMLCs have been studied depending on the characteristics of the datasets, we also studied them in terms of overall performance, taking into account all the evaluation metrics.

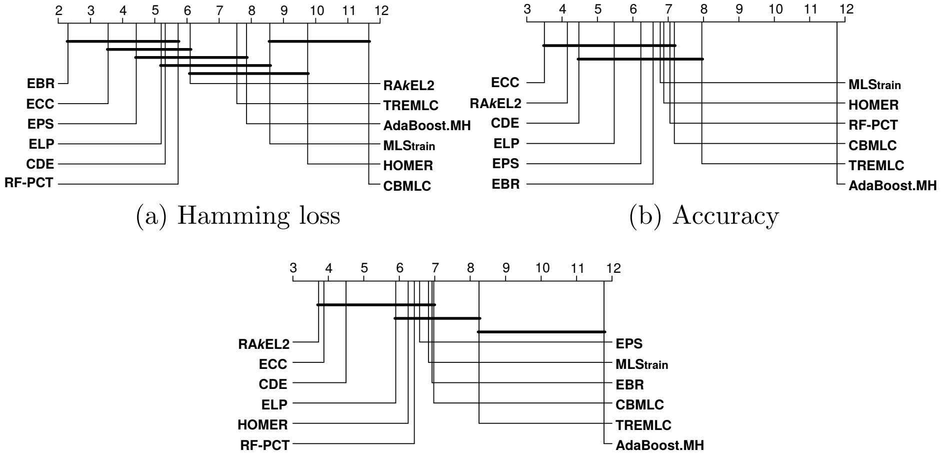
The statistic values and adjusted p-values obtained from the Skillings-Mack's test are shown in Table 14. As Skillings-Mack's test rejects the null hypothesis for all metrics at 95% confidence level, the Shaffer's post-hoc test was also performed. Significant differences among EMLC methods for all performance metrics at 95% confidence level are shown for example-based metrics in Figure 1, for label-based metrics in Figure 2, and for efficiency metrics in Figure 3.

Table 14: Skillings-Mack's test statistic values and p-values for all the evaluation metrics. The null hypothesis is rejected for all the metrics.

	Statistic	***p-value***
Hamming loss	118.78	0.000000
Accuracy	78.50	0.000000
FMeasure$_{\mathbf{ex}}$	77.99	0.000000
FMeasure$_{\mathbf{mac}}$	91.36	0.000000
FMeasure$_{\mathbf{mic}}$	93.00	0.000000
Training time	162.26	0.000000
Test time	170.37	0.000000

Regarding the example-based metrics (Figure 1) we see that EBR performed best for Hamming loss but it is the 6th for accuracy and the 9th for FMeasure$_{\text{ex}}$. ECC had best average ranking for accuracy and second for both Hamming loss and FMeasure$_{\text{ex}}$. Finally, RAkEL2 is the best for FMeasure$_{\text{ex}}$, second for accuracy but the 7th for Hamming loss, having significant differences with EBR at 95% confidence levelf.

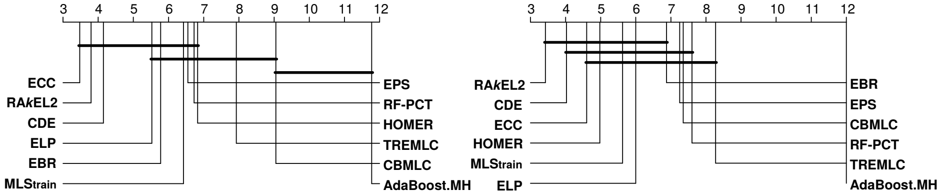
For the label-based metrics (Figure 2), RAkEL2 is the best for FMeasure$_{\text{mac}}$ and the second for FMeasure$_{\text{mic}}$, while ECC is the best in FMeasure$_{\text{mic}}$ and third for FMeasure$_{\text{mac}}$. That means that when all labels are considered to be equal, RAkEL2 performs better, but if the evaluation is biased by the frequency of the labels, ECC is the best choice. Anyhow, these two methods perform better on imbalanced problems than other methods which make the problem even more imbalanced, such as ELP. Also CDE achieves good results in both metrics being the second for FMeasure$_{\text{mac}}$ and the third for FMeasure$_{\text{mic}}$. It is noted that CDE even though there are cases where it does not finish the execution, obtains competitive results.



(a) Hamming loss (b) Accuracy

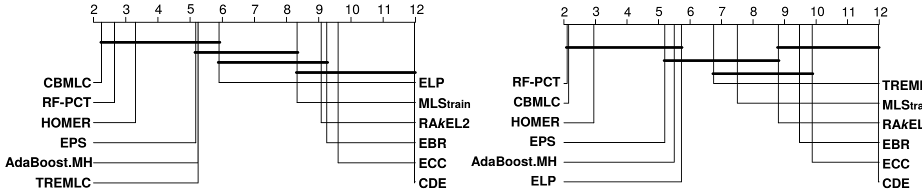
(c) $\text{FMeasure}_{\text{ex}}$

Figure 1: Critical diagrams for the example-based metrics: results of the Shaffer's test at 95% confidence for (a) Hamming loss, (b) accuracy and (c) $\text{FMeasure}_{\text{ex}}$.



(a) $\text{FMeasure}_{\text{mic}}$ (b) $\text{FMeasure}_{\text{mac}}$

Figure 2: Critical diagrams for the label-based metrics: results of the Shaffer's test at 95% confidence for (a) $\text{FMeasure}_{\text{mic}}$ and (b) $\text{FMeasure}_{\text{mac}}$.



(a) Training time (b) Test time

Figure 3: Critical diagrams for the efficiency measures: results of the Shaffer's test at 95% confidence for (a) training time and (b) test time.

AdaBoost.MH is the algorithm with the worst performance in four of the five metrics. This is given because despite combining several Decision Stump classifiers will improve the use of a single of these classifiers, it does not achieve good results against other EMLCs using a more powerful base classifier such as C4.5. CBMLC and TREMLC are also two algorithms that usually are in the last positions of the rankings. Both algorithms reduce significantly the data in each classifier (CBMLC split the data into five different groups and TREMLC uses 70% of the instances and 51% of the features in each classifier), creating weaker models that finally do not achieve good results.

In order to evaluate the efficiency of the algorithms, those that did not finish the execution were assigned a considerably high execution time so that they get the worst ranking in those cases. In terms of efficiency (Figure 3), CBMLC was the fastest algorithm in training, including significant differences with algorithms such as RA*k*EL2, EBR and ECC. The high efficiency of CBMLC is due to the decomposition of the data into several groups of similar data, building models over a dataset with a reduced number of instances and also possibly over a reduced set of labels, which leads to much simpler base classifiers. However, despite its speed, CBMLC does not have a good performance as previously demonstrated. On the other hand, RF-PCT was the algorithm with a shorter average test time since it builds the trees making decisions in small random subsets of features, including significant differences with RA*k*EL2, EBR and ECC among others. In training and testing times, CDE was the slowest algorithm, to the point that it did not finish running on five datasets.

Finally, average rankings for each EMLC and for all performance metrics (not including training and testing times) are shown in Table 15. In order to calculate the average ranking for each algorithm, the ranking values for each metric and all datasets have been averaged. Finally, a meta-ranking was also calculated for each algorithm as the average value of the rankings of all metrics, obtaining a unique value for each algorithm. As shown, ECC achieves the best performance overall for all metrics and datasets, followed by RA*k*EL2 and CDE. It was also shown that ECC is always among the top three algorithms for all evaluation metrics (not considering times), which further strengthens its overall good performance. However, it is one of the worst in both training and test times. On the other hand, CDE, despite being the third best algorithm, has an extremely high complexity which causes sometimes not finished running.

Table 15: Average rankings for all evaluation metrics and meta-ranking values for each EMLC.

	EBR	**ECC**	**MLStrain**	**HOMER**	**AdaB.MH**	**ELP**	**EPS**	**RAkEL2**	**TREMLC**	**CDE**	**RF-PCT**	**CBMLC**
Hamming loss	**2.30**	3.55	8.33	9.45	7.60	5.10	4.40	6.05	7.35	5.05	5.68	11.35
Accuracy	6.50	**3.50**	6.53	6.73	11.48	5.33	6.13	4.13	7.73	4.33	6.88	6.98
FMeasure$_{\text{ex}}$	6.83	3.88	6.58	6.13	11.48	5.75	6.48	**3.70**	8.00	4.35	6.25	6.80
FMeasure$_{\text{mac}}$	6.80	4.60	5.58	4.83	11.70	5.85	7.05	**3.43**	8.00	3.88	7.40	7.10
FMeasure$_{\text{mic}}$	5.75	**3.48**	6.25	6.63	11.48	5.38	6.45	3.80	7.70	3.98	6.58	8.75
Meta-rank	5.64	**3.80**	6.65	6.75	10.75	5.48	6.10	4.22	7.76	4.32	6.56	8.20

5.6. Guidelines to choose the best EMLC based on the characteristics of the data

In this section we summarize the tips and guidelines presented in the discussion of the different experiments to choose the EMLC that best fits to the dataset.

With respect to the imbalance level of the dataset, the results showed that ELP, which deal with all labels at once is the algorithm with best average performance for datasets with small imbalance ($avgIR < 2$), while RAkEL2, which considers the labels in small subsets obtaining a much more balanced output space, is the best algorithm on average for moderately and very imbalanced datasets ($avgIR \geq 2$).

Regarding the relationship among labels, the results indicate that both RAkEL2 and ECC, which take into account the relationships between labels to a lesser extent, are good options for low and medium dependent datasets ($rDep < 0.7$). However, for highly dependent datasets ($rDep \geq 0.7$) ELP is the best algorithm. Since ELP considers all possible relationships among labels, it can exploit the relationship among labels more exhaustively and therefore it achieves a better performance in highly dependent datasets.

In terms of efficiency, CBMLC was the fastest algorithm in training and test times for small datasets ($complexity < 1\text{E}6$), however, it does not achieve a good performance. On the other hand, RF-PCT was the fastest in both training and test times for medium and large datasets ($complexity \geq 1\text{E}6$), also getting an acceptable performance. It is worth mentioning EPS, a combination of good performance and fast algorithm, which is a good option if a fast but also accurate classifier is needed.

Finally, the results of the experimental comparison with all metrics, showed that ECC, followed by RAkEL, was the algorithm with best overall performance for all the metrics used.

6. Conclusions

In this paper, we presented an experimental review of the state-of-the-art EMLC methods, comparing a total of 18 methods over 20 datasets from different domains and with different characteristics. As a result of the study and the fact that no taxonomy for EMLCs was available in the literature, a novel categorization of the EMLC methods was also proposed. EMLC methods have been categorized based on which multi-label method they use, such as BR, LP, PCT or independent of the multi-label classifier. In addition, they have been categorized based on the way the ensemble is built, including the *classifier level*, *label level*, *feature level* and *data level*.

The performance of the EMLCs was evaluated from different points of view and taking into account the characteristics of the datasets such as the imbalance, relationship, and dimensionality. Some guidelines were also given in order to choose the EMLC that best fits to the data in each case.

References

[1] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (1) (2010) 1–39. doi:10.1007/s10462-009-9124-7.

[2] M. Wozniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Information Fusion 16 (2014) 3 – 17, special Issue on Information Fusion in Hybrid Intelligent Fusion Systems. doi:http://dx.doi.org/10.1016/j.inffus.2013.04.006.

[3] T. G. Dietterich, Ensemble Methods in Machine Learning, Springer Berlin Heidelberg, 2000, pp. 1–15.

[4] W. Leigh, R. Purvis, J. M. Ragusa, Forecasting the {NYSE} composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support, Decision Support Systems 32 (4) (2002) 361 – 377. doi:https://doi.org/10.1016/S0167-9236(01)00121-X.

[5] A. C. Tan, D. Gilbert, Y. Deville, Multi-class protein fold classification using a new ensemble machine learning approach, Genome Informatics 14 (2003) 206–217. doi:10.11234/gi1990.14.206.

[6] P. Mangiameli, D. West, R. Rampal, Model selection for medical diagnosis decision support systems, Decision Support Systems 36 (3) (2004) 247 – 259. doi:https://doi.org/10.1016/S0167-9236(02)00143-4.

[7] H.-J. Lin, Y.-T. Kao, F.-W. Yang, P. S. P. Wang, Content-based image retrieval trained by adaboost for mobile application, International Journal of Pattern Recognition and Artificial Intelligence 20 (04) (2006) 525–541. doi:10.1142/S021800140600482X.

[8] A. Schclar, A. Tsikinovsky, L. Rokach, A. Meisels, L. Antwarg, Ensemble methods for improving the performance of neighborhood-based collaborative filtering, in: Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, 2009, pp. 261–264. doi:10.1145/1639714.1639763.

[9] E. Gibaja, S. Ventura, A tutorial on multilabel learning, ACM Computing Surveys 47 (3).

[10] F. Herrera, F. Charte, A. Rivera, M. del Jesus, Multilabel Classification. Problem analysis, metrics and techniques, Springer, 2016. doi:10.1007/978-3-319-41111-8.

[11] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Deroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognition 45 (9) (2012) 3084–3104.

[12] G. Nasierding, A. Kouzani, Empirical study of multi-label classification methods for image annotation and retrieval, 2010, pp. 617–622. doi:10.1109/DICTA.2010.113.

[13] P. Brandt, D. Moodley, A. W. Pillay, C. J. Seebregts, T. de Oliveira, An Investigation of Classification Algorithms for Predicting HIV Drug Resistance without Genotype Resistance Testing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 236–253. doi:10.1007/978-3-642-53956-5_16.

[14] N.-Y. Nair-Benrekia, P. Kuntz, F. Meyer, Learning from multi-label data with interactivity constraints: An extensive experimental study, Expert Systems with Applications 42 (13) (2015) 5723 – 5736. doi:https://doi.org/10.1016/j.eswa.2015.03.006.

[15] E. Gibaja, S. Ventura, Multi-label learning: a review of the state of the art and ongoing research, WIREs Data Mining Knowl Discov 2014. doi: 10.1002/widm.1139.

[16] G. Tsoumakas, I. Katakis, I. Vlahavas, Data Mining and Knowledge Discovery Handbook, Part 6, Springer, 2010, Ch. Mining Multi-label Data, pp. 667–685.

[17] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern recognition 37 (2004) 1757–1771.

[18] A. Clare, A. Clare, R. D. King, Knowledge discovery in multi-label phenotype data, in: Lecture Notes in Computer Science, Springer, 2001, pp. 42–53.

[19] H. Blockeel, L. D. Raedt, J. Ramon, Top-down induction of clustering trees, in: Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 55–63.

[20] M. Petrovskiy, Paired comparisons method for solving multi-label learning problem, in: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems, HIS '06, 2006, p. 42. doi:10.1109/HIS.2006.54.

[21] J. Li, J. Xu, A fast multi-label classification algorithm based on double label support vector machine, in: Proceedings of the 2009 International Conference on Computational Intelligence and Security (CIS 09), 2009, pp. 30–35.

[22] K. Crammer, Y. Singer, A family of additive online algorithms for category ranking, J. Mach. Learn. Res. 3 (2003) 1025–1058.

[23] 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 (2006) 1338–1351.

[24] M.-L. Zhang, Z.-H. Zhou, A k-Nearest Neighbor Based Algorithm for Multi-label Classification, in: Proceedings of the IEEE International Conference on Granular Computing (GrC), Vol. 2, The IEEE Computational Intelligence Society, Beijing, China, 2005, pp. 718–721.

[25] W. Cheng, E. Hullermeier, Combining instance-based learning and logistic regression for multilabel classification, Machine Learning 76 (2-3) (2009) 211–225.

[26] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multilabel classification, Machine Learning 85 (3) (2011) 335–359.

[27] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[28] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: ECML 09: 20th European conference on machine learning, 2009, p. 254269.

[29] G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, I. Vlahavas, Correlation-based pruning of stacked binary relevance models for multi-label learning, in: 1st International Workshop on Learning from Multi-Label Data (MLD'09), 2009, pp. 101–116.

[30] J. Cohen, P. Cohen, S. G. West, L. S. Aiken, Applied Multiple Regression / Correlation Analysis for the Behavioral Sciences, 2002.

[31] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08), 2008.

[32] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119 – 139. doi:http://dx.doi.org/10.1006/jcss.1997.1504.

[33] L. Breiman, Arcing classifiers, The Annals of Statistics 26 (3) (1998) 801–824.

[34] Y. Freund, R. E. Schapire, et al., Experiments with a new boosting algorithm, in: icml, Vol. 96, 1996, pp. 148–156.

[35] R. Maclin, D. Opitz, An empirical evaluation of bagging and boosting, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI'97/IAAI'97, 1997, pp. 546–551.

[36] R. E. Schapire, Y. Singer, Boostexter: A boosting-based system for text categorization, Machine Learning 39 (2000) 135–168.

[37] J. Read, B. Pfahringer, G. Holmes, Multi-label classification using ensembles of pruned sets, in: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, IEEE, 2008, pp. 995–1000.

[38] G. Tsoumakas, I. Katakis, I. Vlahavas, Random k-labelsets for multi-label classification, IEEE Transactions on Knowledge and Data Engineering 23 (7) (2011) 1079–1089.

[39] L. Rokach, A. Schclar, E. Itach, Ensemble methods for multi-label classification, Expert Systems with Applications 41 (16) (2014) 7507 – 7523. doi:http://dx.doi.org/10.1016/j.eswa.2014.06.015.

[40] G. Nasierding, A. Z. Kouzani, G. Tsoumakas, A triple-random ensemble classification method for mining multi-label data, in: ICDMW 2010, The 10th IEEE International Conference on Data Mining Workshops, Sydney, Australia, 13 December 2010, 2010, pp. 49–56. doi:10.1109/ICDMW.2010.139.

[41] L. Tenenboim, L. Rokach, B. Shapira, Multi-label classification by analyzing labels dependencies, in: Proceedings of the 1st international workshop on learning from multi-label data, Bled, Slovenia, 2009, pp. 117–132.

[42] P. E. Greenwood, M. S. Nikulin, A guide to chi-squared testing, Wiley-Interscience 280.

[43] L. Tenenboim-Chekina, L. Rokach, B. Shapira, Identification of label dependencies for multi-label classification, in: Working Notes of the Second International Workshop on Learning from Multi-Label Data, 2010, pp. 53–60.

[44] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth and Brooks, 1984.

[45] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Ensembles of Multi-Objective Decision Trees, Springer Berlin Heidelberg, 2007, pp. 624–631. doi:10.1007/978-3-540-74958-5_61.

[46] L. Rokach, Decision forest: Twenty years of research, Information Fusion 27 (2016) 111 – 125. doi:http://dx.doi.org/10.1016/j.inffus.2015.06.005.

[47] G. Nasierding, G. Tsoumakas, A. Z. Kouzani, Clustering based multi-label classification for image annotation and retrieval, in: Proceedings

of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11-14 October 2009, 2009, pp. 4514–4519. doi:10.1109/ICSMC.2009.5346902.

[48] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.

[49] L. I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.

[50] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, International Journal of Data Warehousing and Mining 3 (3) (2007) 1–13.

[51] F. Charte, A. J. Rivera, M. J. del Jesus, F. Herrera, Addressing imbalance in multilabel classification: Measures and random resampling algorithms, Neurocomputing 163 (Supplement C) (2015) 3 – 16. doi:https://doi.org/10.1016/j.neucom.2014.08.091.

[52] L. Chekina, L. Rokach, B. Shapira, Meta-learning for selecting a multilabel classification algorithm, 2011, pp. 220–227.

[53] J. Read, Scalable multi-label classification, PhD Thesis, University of Waikato.

[54] E. Goncalves, A. Plastino, A. A. Freitas, A genetic algorithm for optimizing the label ordering in multi-label classifier chains, in: IEEE 25th International Conference on Tools with Artificial Intelligence, IEEE Computer Society Conference Publishing Services (CPS), 2013, pp. 469–476.

[55] H. Shao, G. Li, G. Liu, Y. Wang, Symptom selection for multi-label data of inquiry diagnosis in traditional chinese medicine, Science China Information Sciences 56 (5) (2013) 1–13. doi:10.1007/s11432-011-4406-5.

[56] H. Blockeel, S. Deroski, J. Grbovi, Simultaneous prediction of multiple chemical parameters of river water quality with tilde, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 1704 (1999) 32–40.

[57] D. Greene, P. Cunningham, A matrix factorization approach for integrating multiple data views, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I, ECML PKDD '09, 2009, pp. 423–438. doi:10.1007/978-3-642-04180-8_45.

[58] F. Briggs, Y. Huang, R. Raich, K. Eftaxias, Z. Lei, W. Cukierski, S. F. Hadley, A. Hadley, M. Betts, X. Z. Fern, J. Irvine, L. Neal, A. Thomas, G. Fodor, G. Tsoumakas, H. W. Ng, T. N. T. Nguyen, H. Huttunen, P. Ruusuvuori, T. Manninen, A. Diment, T. Virtanen, J. Marzat, J. Defretin, D. Callender, C. Hurlburt, K. Larrey, M. Milakov, The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment, in: IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2013, Southampton, United Kingdom, September 22-25, 2013, 2013, pp. 1–8. doi:10.1109/MLSP.2013.6661934.

[59] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, NIPS'01, 2001, pp. 681–687.

[60] J. Xu, J. Liu, J. Yin, C. Sun, A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously, Knowledge-Based Systems 98 (2016) 172 – 184. doi:http://dx.doi.org/10.1016/j.knosys.2016.01.032.

[61] S. Diplaris, G. Tsoumakas, P. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: Proc. 10th Panhellenic Conference on Informatics (PCI 2005), 2005, pp. 448–456.

[62] Yelp dataset challenge, `http://www.ics.uci.edu/~vpsaini/`, last access: 26-06-2017.

[63] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, W. Duch, A shared task involving multi-label classification of clinical free text, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing (BioNLP '07), 2007, pp. 97–104.

[64] The 20 newsgroups data set, http://qwone.com/~jason/20Newsgroups/, last access: 26-06-2017.

[65] C. Snoek, M.Worring, J. van Gemert, J.-M. Geusebroek, A. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of ACM Multimedia, 2006, pp. 421–430.

[66] J. M. Moyano, E. L. Gibaja, S. Ventura, MLDA: A tool for analyzing multi-label datasets, Knowledge-Based Systems 121 (2017) 1–3. doi:10.1016/j.knosys.2017.01.018.

[67] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: An update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.

[68] Meka: A multi-label extension to weka, http://meka.sourceforge.net/, last access: 31-03-2017.

[69] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: A java library for multi-label learning, Journal of Machine Learning Research 12 (2011) 2411–2414.

[70] K. Sechidis, G. Tsoumakas, I. Vlahavas, On the stratification of multi-label data, Lecture Notes in Computer Science 6913 LNAI (PART 3) (2011) 145–158.

[71] M. Chatfield, A. Mander, The skillingsmack test (friedman test when there are missing data), The Stata journal 9 (2) (2009) 299–305.

[72] P. Srisuradetchai, Skillings.mack: The skillings-mack test statistic for block designs with missing observations, https://CRAN.R-project.org/package=Skillings.Mack, last access: 12-12-2017.

[73] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Statist. 11 (1) (1940) 86–92. doi:10.1214/aoms/1177731944.
URL http://dx.doi.org/10.1214/aoms/1177731944

[74] J. P. Shaffer, Modified sequentially rejective multiple test procedures, Journal of the American Statistical Association 81 (395) (1986) 826–831.

[75] S. Garcia, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, Journal of Machine Learning Research 9 (Dec) (2008) 2677–2694.

[76] P. Nemenyi, Distribution-free multiple-comparisons, PhD Thesis, Pricenton University (USA).

[77] S. P. Wright, Adjusted p-values for simultaneous inference, Biometrics (1992) 1005–1013.