An evolutionary approach to build ensembles of multi-label classifiers

Article *in* Information Fusion · November 2018

DOI: 10.1016/j.inffus.2018.11.013

Jose M. Moyano
University of Seville

Krzysztof Cios
Virginia Commonwealth University

Sebastian Ventura
University of Córdoba

An evolutionary approach to build ensembles of multi-label classifiers

Jose M. Moyano[a,e], Eva L. Gibaja[a,e], 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

In recent years, the multi-label classification task has gained the attention of the scientific community given its ability to solve problems where each of the instances of the dataset may be associated with several class labels at the same time instead of just one. The main problems to deal with in multi-label classification are the imbalance, the relationships among the labels, and the high complexity of the output space. A large number of methods for multi-label classification has been proposed, but although they aimed to deal with one or many of these problems, most of them did not take into account these characteristics of the data in their building phase. In this paper we present an evolutionary algorithm for automatic generation of ensembles of multi-label classifiers by tackling the three previously mentioned problems, called Evolutionary Multi-label Ensemble (EME). Each multi-label classifier is focused on a small subset of the labels, still considering the relationships among them but avoiding the high complexity of the output space. Further, the algorithm automatically designs the ensemble evaluating both its pre-

[*]Corresponding author

Email address: sventura@uco.es (Sebastián Ventura)

dictive performance and the number of times that each label appears in the ensemble, so that in imbalanced datasets infrequent labels are not ignored. For this purpose, we also proposed a novel mutation operator that considers the relationship among labels, looking for individuals where the labels are more related. EME was compared to other state-of-the-art algorithms for multi-label classification over a set of fourteen multi-label datasets and using five evaluation measures. The experimental study was carried out in two parts, first comparing EME to classic multi-label classification methods, and second comparing EME to other ensemble-based methods in multi-label classification. EME performed significantly better than the rest of classic methods in three out of five evaluation measures. On the other hand, EME performed the best in one measure in the second experiment and it was the only one that did not perform significantly worse than the control algorithm in any measure. These results showed that EME achieved a better and more consistent performance than the rest of the state-of-the-art methods in MLC.

Keywords: Multi-label classification, Ensemble, Evolutionary algorithm

1. Introduction

In recent years, the Multi-Label Classification (MLC) task has gained the attention of the scientific community given its ability to solve problems where each of the instances may be associated to several class labels at the same time, instead of just one. Let be $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ the set of q different binary labels (with $q > 2$), and $\mathcal{X}$ the set of m instances, each composed by d input features; let us define the multi-label classification task as learning a mapping from an example $\boldsymbol{x}_i \in \mathcal{X}$ to a set of labels $\boldsymbol{y}_i \subseteq \mathcal{L}$. Labels in the set $\boldsymbol{y}_i$ are called relevant labels, and the rest ($\overline{\boldsymbol{y}}_i$) are called irrelevant. A great deal of real-world problems have been successfully solved thanks to the application of MLC, such as social networks mining, where each user could be subscribed to several groups of interest [1]; multimedia annotation, where each image or multimedia item could be associated to several class labels [2]; and text categorization, where each document could be categorized in several topics simultaneously [3]; among others.

The most challenging problems in MLC are dealing with the imbalance of the data [4], modeling compound dependencies among the labels [5], and the possible high dimensionality of the output space [6]. In many problems the labels do not appear with the same frequency in the dataset, with some labels appearing in most of the instances and other that are barely present, appearing in a few instances. This might lead to an imbalanced dataset where the frequent labels could be much better predicted than the infrequent ones, as there is very little information about the infrequent labels. Besides, labels are not usually independent but tend to be related to each other, where a label may appear more frequently with some labels than with others. The fact of modeling, or lack of, compound dependencies among labels has a decisive effect not only on the predictive performance of the model but also on its complexity. The complexity of the model is also usually related to the size of the output space. The greater the number of labels, the higher the complexity of the model, which can make the problem intractable.

In order to try to overcome these problems, several methodologies have been proposed in the literature. For example, Pruned Sets (PS) [7] was proposed in order to reduce the imbalance in the final problem. Besides, to overcome the problem of modelling the compound dependencies among labels, Classifier Chains (CC) [5] considered the relationship among different binary methods that originally did not take into account. For the output dimensionality problem, RAndom k-labELsets (RAkEL) [8] divided the label space into smaller subsets, resulting in less complex output spaces. Furthermore, the continuous stream of input data is a growing problem in many data mining tasks, and it has been also successfully addressed in MLC [9, 10]. Many of these proposed methods were based on the combination of several classifiers. However, in MLC only those methods that combine several classifiers which are able to deal with multi-label data are considered as Ensembles of Multi-Label Classifiers (EMLCs) [11]. On the other hand, besides tackling the aforementioned problems, ensembles usually perform better than single classifiers. One of the ways to obtain an ensemble that outperform each of the individuals classifiers is to combine a set of diverse classifiers [12, 13]. Despite this fact, many of the proposed ensemble methods in the literature generate diversity only by random sampling of attributes, instances, or labels for each classifier, but not ensuring that the entire ensemble is diverse enough.

In this paper, we propose an evolutionary approach for the automatic generation of ensembles of diverse and competitive multi-label classifiers. The algorithm, called Evolutionary Multi-Label Ensemble (EME), takes into account characteristics of the multi-label data such as the relationships among the labels, imbalance of the data, and complexity of the output space. The ensemble is based on projections of the label space, considering in this way the relationships among the labels but also reducing the computational cost in cases where the output space is complex. These subsets of labels are not only randomly selected but also they evolve with the generations of the evolutionary algorithm, looking for the combinations that perform the best. Also, a novel mutation operator is proposed, so that it considers the relationship among labels favouring more related combinations of labels. Further, EME takes into account all the labels approximately the same number of times in the ensemble, regardless of their frequency or its ease to be predicted; so that the imbalance of the data is considered and the infrequent labels are not ignored. For that, the fitness function takes into account both the predictive performance of the model and the number of times that each label is considered in the ensemble. Finally, the diversity of the ensemble is not taken into account explicitly, but the ensembles evolve selecting their classifiers based on their overall performance.

The experimental study carried out over fourteen multi-label datasets compared EME with classic state-of-the-art methods in MLC and also other EMLCs using five evaluation measures. The first experiment determined that EME performed significantly better than classic MLC methods in three of the five evaluation measures. In the second experiment, EME achieved the best performance in only one measure, but it was the only algorithm that did not perform significantly worse than any of the control algorithm for any evaluation measures. These results showed that EME achieved a better and more consistent performance than the rest of the state-of-the-art methods in MLC.

The rest of the article is organized as follows: Section 2 includes related work in multi-label classification, Section 3 describes the proposed evolutionary algorithm, Section 4 presents the experimental study and Section 5 presents and discusses the results. Finally, Section 6 ends with conclusions.

2. Related work

The traditional single-label classification task aims to predict the class or group associated to each of the instances described by a set or input features. Each of the instances is classified in just one class from a previously defined set of classes. However, in MLC, each instance may be labeled with more than one of the q class labels simultaneously. Given a set of q predefined labels $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$, the subset of relevant labels associated with each of the instances can be viewed as a binary vector $\boldsymbol{y} = \{0, 1\}^q$ where each element is 1 if the label is relevant and 0 otherwise. In this way, the goal of MLC is to predict, for an unseen instance, a bipartition including its sets of relevant ($\hat{\boldsymbol{y}}$) and irrelevant labels ($\overline{\hat{\boldsymbol{y}}}$).

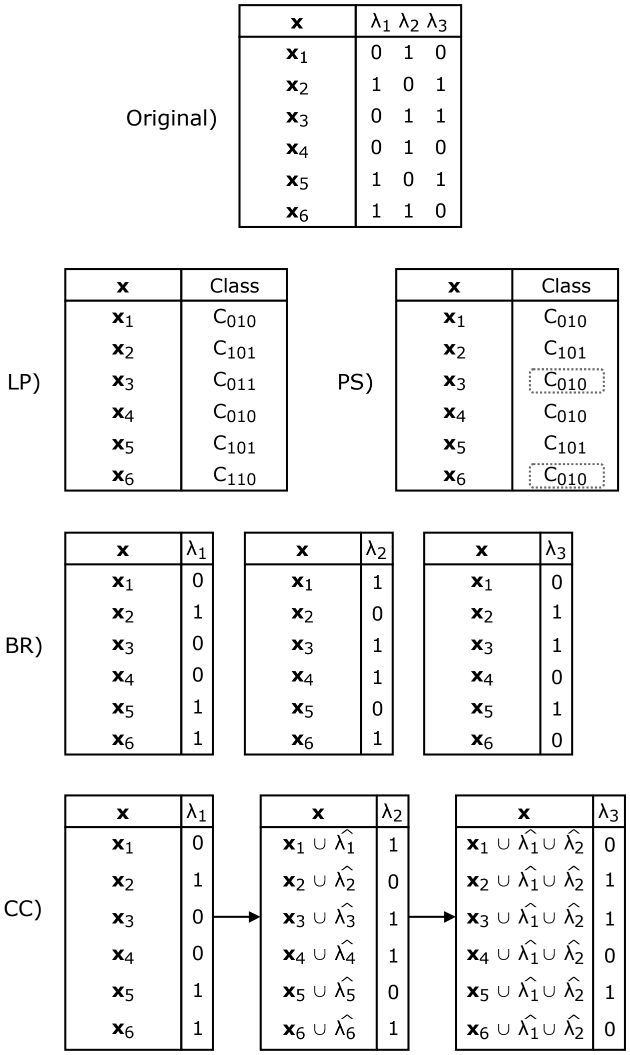
Several methods for MLC have been proposed in the literature, aiming to handle with the three main problems in MLC, such as the imbalance of the output space, the relationship among labels and the high dimensionality of the output space. These methods are categorized into three main groups: problem transformation, algorithm adaptation, and EMLCs [14, 15].

Problem transformation methods transform the multi-label problem into one or more single-label problems. These problems are then solved by using traditional single-label classification methods. For ease of understanding, schemes of the main transformations are presented in Figure 1. Binary Relevance (BR) [16] decomposes the multi-label problem into q independent binary single-label problems, then building q independent binary classifiers, one for each label. BR is simple and intuitive, but the fact of considering the labels independently makes it unable to model the compound dependencies among the labels. BR do not deal with any of the previously described problems in MLC. In order to overcome the label independence assumption of BR, Classifier Chain (CC) [5] generates q binary classifiers but linked in such a way that each binary classifier also includes the label predictions of previous classifiers in the chain as additional input features. In this way and unlike BR, CC is able to model the relationships among the labels without introducing more complexity. However, although it deal with the relationship among labels, it does not consider them, or any other characteristics of the data to select the chain. Since the order of the chain has a determinant effect on its performance, other approaches have been proposed to select the best chain ordering [17, 18].

Label Powerset (LP) [19] transforms the multi-label problem into a multi-class problem, creating a new class for each distinct combination of labels, called labelset, that appears in the dataset. This method is able to strongly model the relationships among the labels, but its complexity grows exponentially with the number of labels; it is also not able to predict a labelset that does not appear in the training set. Therefore, although it is able to handle with the relationship among labels, LP greatly increases the dimensionality of the output space, as well as its imbalance. Pruned Sets (PS) [7] tries to reduce the complexity of LP, focusing on most important combinations of labels by pruning instances with less frequent labelsets. To compensate for this loss of information, it reintroduces the pruned instances with a more frequent subset of labels. Thus, PS considers the imbalance of LP's output space to reduce its dimensionality and complexity. ChiDep [20] creates groups of dependent labels based on the χ^2 test for labels dependencies identification. For each group of dependent labels it builds a LP classifier, while for each single label which is not in any group it builds a binary classifier. ChiDep tries to reduce the disadvantages of the independence assumption of the binary methods and allows for simpler LP methods. Besides, ChiDep considers the relationship among group of labels and the dimensionality of the output space in building phase, therefore being able to reduce the imbalance in each model if the groups are small.

The methods in the algorithm adaptation group adapt or extend existing machine learning methods to directly handle multi-label data. Predictive Clustering Trees (PCTs) [21] are decision trees where the data is partitioned in each node using a clustering algorithm. In order to adapt them to MLC, the distance between two instances for the clustering algorithm is calculated as the sum of the Gini Indices [22] of all labels, so it considers the relationship among labels when building the model. Instance-based algorithms have been also adapted for MLC, such as Multi-Label k-Nearest Neighbors (ML-kNN) [23]. For each unknown instance, first the k nearest neighbors are found, then the number of neighbors belonging to each label are counted, and finally the *maximum a posteriori* principle is used to identify the labels for the given instance. As ML-kNN considers all label assignments of k-nearest neighbors to label a new instance, it implicitly consider the relationship among labels to build the model. On the other hand, the traditional feed-forward neural network have been also adapted in the Back-Propagation for Multi-Label Learning (BP-MLL) [24]. In this way, an error function for multi-label scenarios was proposed, which takes into account the predicted ranking of labels. The ranking of labels also imply the relationship among labels, so BP-MLL considers it in the building phase. A wider description of algorithm adaptation methods in MLC can be found in [14].

The third group of methods includes the EMLCs. Although many of the MLC algorithms are based on the combination of several classifiers, only are considered as EMLCs those that combine several classifiers which are able to deal with multi-label data [11]. Thus, although BR combines several classifiers it is not an EMLC since it combines single-label but not multi-label classifiers. Ensemble of BR classifiers (EBR) [5] builds an ensemble of n BR classifiers where each is trained with a sample from the training dataset, being n the number of desired multi-label classifiers in the ensemble. The



Original)

$\mathbf{x}$	λ_1	λ_2	λ_3
$\mathbf{x}_1$	0	1	0
$\mathbf{x}_2$	1	0	1
$\mathbf{x}_3$	0	1	1
$\mathbf{x}_4$	0	1	0
$\mathbf{x}_5$	1	0	1
$\mathbf{x}_6$	1	1	0

LP)

$\mathbf{x}$	Class
$\mathbf{x}_1$	C_{010}
$\mathbf{x}_2$	C_{101}
$\mathbf{x}_3$	C_{011}
$\mathbf{x}_4$	C_{010}
$\mathbf{x}_5$	C_{101}
$\mathbf{x}_6$	C_{110}

PS)

$\mathbf{x}$	Class
$\mathbf{x}_1$	C_{010}
$\mathbf{x}_2$	C_{101}
$\mathbf{x}_3$	$\boxed{C_{010}}$
$\mathbf{x}_4$	C_{010}
$\mathbf{x}_5$	C_{101}
$\mathbf{x}_6$	$\boxed{C_{010}}$

BR)

$\mathbf{x}$	λ_1
$\mathbf{x}_1$	0
$\mathbf{x}_2$	1
$\mathbf{x}_3$	0
$\mathbf{x}_4$	0
$\mathbf{x}_5$	1
$\mathbf{x}_6$	1

$\mathbf{x}$	λ_2
$\mathbf{x}_1$	1
$\mathbf{x}_2$	0
$\mathbf{x}_3$	1
$\mathbf{x}_4$	1
$\mathbf{x}_5$	0
$\mathbf{x}_6$	1

$\mathbf{x}$	λ_3
$\mathbf{x}_1$	0
$\mathbf{x}_2$	1
$\mathbf{x}_3$	1
$\mathbf{x}_4$	0
$\mathbf{x}_5$	1
$\mathbf{x}_6$	0

CC)

$\mathbf{x}$	λ_1
$\mathbf{x}_1$	0
$\mathbf{x}_2$	1
$\mathbf{x}_3$	0
$\mathbf{x}_4$	0
$\mathbf{x}_5$	1
$\mathbf{x}_6$	1

$\mathbf{x}$	λ_2
$\mathbf{x}_1 \cup \hat{\lambda}_1$	1
$\mathbf{x}_2 \cup \hat{\lambda}_2$	0
$\mathbf{x}_3 \cup \hat{\lambda}_3$	1
$\mathbf{x}_4 \cup \hat{\lambda}_4$	1
$\mathbf{x}_5 \cup \hat{\lambda}_5$	0
$\mathbf{x}_6 \cup \hat{\lambda}_6$	1

$\mathbf{x}$	λ_3
$\mathbf{x}_1 \cup \hat{\lambda}_1 \cup \hat{\lambda}_2$	0
$\mathbf{x}_2 \cup \hat{\lambda}_1 \cup \hat{\lambda}_2$	1
$\mathbf{x}_3 \cup \hat{\lambda}_1 \cup \hat{\lambda}_2$	1
$\mathbf{x}_4 \cup \hat{\lambda}_1 \cup \hat{\lambda}_2$	0
$\mathbf{x}_5 \cup \hat{\lambda}_1 \cup \hat{\lambda}_2$	1
$\mathbf{x}_6 \cup \hat{\lambda}_1 \cup \hat{\lambda}_2$	0

Figure 1: Main problem transformations in MLC. For PS, labelsets appearing less than 2 times are pruned and reintroduced with most frequent subsets.

selection of instances in each BR provides diversity to the ensemble, but as BR, it still does not consider any of the characteristics of the data to build the model. Ensemble of Classifier Chains (ECC) [5] builds an ensemble of n CCs, each with a random chain and a random sample with replacement from the training dataset. The selection of several different chains reduces the risk of selecting a bad chain which could lead to a bad performance, however, they are all created randomly and not based on any of the characteristics of the data. Multi-Label Stacking (MLS) [25] is composed of two phases. In the first phase, q BR classifiers are learned, one for each label; while in the second phase, the input feature set is augmented with the predictions of each binary classifier from the first phase, training q new binary classifiers using the desired outputs as targets. MLS is able to model the relationship among labels thanks to the use of the predictions in the first phase to predict the labels in the second phase. Ensemble of Pruned Sets (EPS) [7] makes an ensemble of n PSs where each classifier is trained with a sample of the training set without replacement. The use of many PSs with different data subsets avoids overfitting effects of pruning instances, but as PS, in datasets with a high number of labels the complexity can be still very high.

Hierarchy Of Multi-label classifiERs (HOMER) [6] generates a tree of multi-label classifiers, where the root contains all labels and each leaf represents one label. At each node, the labels are split with a clustering algorithm, grouping similar labels into a meta-label. HOMER considers the relationship among labels to build the model, making it able to handle with smaller subsets of labels in each node, so that the dimensionality of the output space in each of them is reduced, also reducing the imbalance depending on the internal multi-label classifier used. Random Forest of Predictive Clustering Trees (RF-PCT) [26] builds an ensemble of n PCTs by selecting a random subset of the instances in each model. Further, each PCT selects at each node of the tree the best feature from a random subset of the original ones. As PCT, it considers the relationship among labels in the building phase. Finally, RAndom k-labELsets (RAkEL) [8] builds an ensemble of LP classifiers, where each is built over a random projection of the output space. In this way, RAkEL deals with the relationship among labels as LP does but in a much simpler way. RAkEL handles with the three main problems of the MLC: it is able to detect the compound dependencies among labels, it reduces the dimensionality of the output space by selecting small subsets of labels (a.k.a. k-labelsets), and also the imbalance of each of the methods is not usually high since the reduced number of labels in each of them. However, RAkEL selects the k-labelsets randomly, without considering any the characteristics of the data, which could lead to a poor performance.

A summary of the previously defined method is available in Table 1. This table indicates if each method deals with (D) and/or considers each of the characteristics of the data at building phase (B). Note that there are methods that are able to deal with any of the problems, but they do not consider the corresponding characteristics when building the model. For example RAkEL is able to model the relationship among labels, but it does not consider neither these relationships nor other of the characteristics of the data to select the k-labelsets, it simply creates them randomly. On the other hand, HOMER considers the relationship among labels when building the model, since it split the labelsets into smaller ones considering the relationship among the labels.

Table 1: Summary of state-of-the-art MLC methods. It is indicated with a 'D' if the method is able to deal with the corresponding problem (imbalance, relationships among labels, and high dimensionality of the output space), and with a 'B' if it considers this characteristic at building phase.

	Imbalance	**Relationships**	**Output Dim.**
BR			
CC		D	
LP		D	
PS	D, B	D	D, B
ChiDep	D	D, B	D, B
PCT		D, B	
MLkNN		D, B	
BP-MLL		D, B	
EBR			
ECC		D	
MLS		D	
EPS	D, B	D	D, B
HOMER	D	D, B	D, B
RF-PCT		D, B	
RAkEL	D	D	D, B

3. Evolutionary Multi-label Ensemble

In this section the evolutionary algorithm is presented, focusing on the encoding scheme, the fitness function, and the genetic operators. Then, the time complexity of EME is also presented.

3.1. Evolutionary algorithm

The evolutionary algorithm is based on a generational elitist algorithm [27, 28], that is, it ensures that the best individual in the last generation is the best individual of the evolution. Each individual of the evolutionary algorithm represents a full multi-label ensemble consisting of n multi-label classifiers, each of them modeling a k-labelset.

Figure 2 presents a flowchart of the evolutionary algorithm. At the beginning, the population p of *popSize* individuals is randomly created, considering the size of the k-labelsets and the number of classifiers in each individual (Section 3.2). Then, the initial population is evaluated using the multi-label dataset (Section 3.3). In each generation, *popSize* individuals are selected by tournament selection and stored in s. Each individual in s is considered for crossover and mutation based on their respective probabilities (Section 3.4). Once the genetic operators are applied, these new individuals are evaluated. To maintain elitism, the population in each generation keeps all new individuals, unless the best parent is better than all the children; in this case, the parent replaces the worst child. The *best(set)* and *worst(set)* methods return the best and the worst individual of a *set* respectively. At the end of the generations, the best individual in the last generation is returned as the best ensemble.

3.2. Individuals

The individuals, which are codified as binary arrays of $n \times q$ elements, represent a full multi-label ensemble formed by n multi-label classifiers and q labels. Each classifier of the ensemble is based on projections of the output space, built over a small subset of k labels. The parameter k is the same for all classifiers, so the number of labels in each classifier is fixed. Each fragment of size q in the individual represents the k-labelset of each classifier in the ensemble.

EME is implemented with the ability to use any multi-label classifier. However, LP is proposed as base classifier. Since EME has been designed to consider the dependencies among labels avoiding a high complexity, using a base classifier which does not consider the relationship among labels, such BR, makes no sense. Further, the use of LP has many advantages over other methods that also consider the relationship among labels, as for example CC. If k is small, as proposed, LP builds an unique model for each k-labelset able to model the dependencies among all labels at a time with a low complexity due to the reduced output space. On the other hand, CC needs to build k different binary models, and not all dependencies are considered in each model; for example the first label in the chain is modeled without considering the dependencies with the rest, the second is modeled considering the dependency with only the previous one, and so on. Besides, the use of CC would

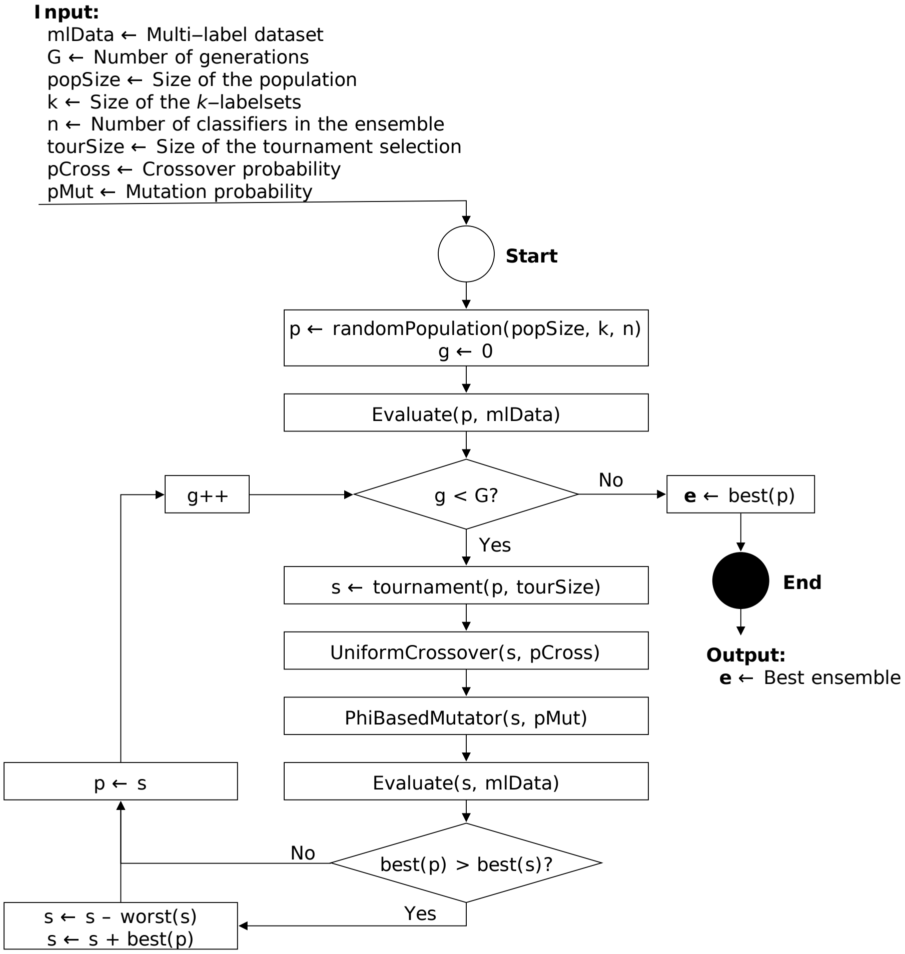

Input:
mlData ← Multi–label dataset
G ← Number of generations
popSize ← Size of the population
k ← Size of the k–labelsets
n ← Number of classifiers in the ensemble
tourSize ← Size of the tournament selection
pCross ← Crossover probability
pMut ← Mutation probability
Start
p ← randomPopulation(popSize, k, n)
g ← 0
Evaluate(p, mlData)
g++
g < G?
No
e ← best(p)
Yes
s ← tournament(p, tourSize)
End
UniformCrossover(s, pCross)
Output:
e ← Best ensemble
PhiBasedMutator(s, pMut)
p ← s
Evaluate(s, mlData)
No
best(p) > best(s)?
Yes
s ← s – worst(s)
s ← s + best(p)

Figure 2: Flowchart of the evolutionary process.

introduce a higher computational cost due to the increase in the number of different possible individuals by the different chains. The performance of EME, as well as that of the vast majority of multi-label methods, is biased by the performance of the single-label method used. Many ensemble methods in MLC have used decision trees as base classifier [5, 8, 29] with promising results, so the C4.5 decision tree (Weka's J48 [30]) is used as a single-label classifier. For the parameters of C4.5, we used a minimum number of objects per leaf of 2, and a pruning confidence of 0.25. Although we used these parameters, optimizing them for each specific problem would lead to a better performance, both in EME and in any other method that used C4.5.

The individuals in the initial population are generated by randomly choosing k bits to a value of 1 for each fragment representing a multi-label classifier. Then, with the evolutionary algorithm the individuals are crossed and mutated, evolving towards a more promising combination of multi-label classifiers, instead of being mere random selections. Figure 3 shows the genotype (represented as a one-dimensional array and as a matrix) and the phenotype of an individual. For example, the first, represented by $[0, 1, 1, 0, 1, 0]$ indicates that labels λ_2, λ_3 and λ_5 are included in the first classifier of the ensemble.

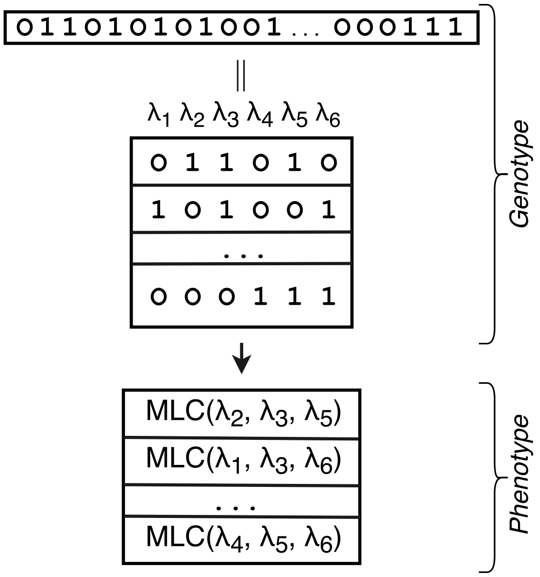
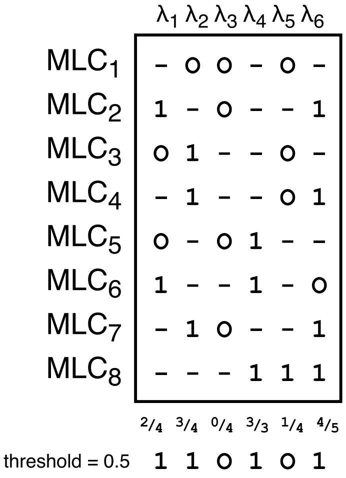


Figure 3: Genotype and phenotype of an individual.

At the time of evaluating each individual, first the corresponding multi-label ensemble must be generated. For each fragment of size q in the individual, the full training dataset is filtered keeping only the labels in its k-labelset, and the corresponding multi-label classifier is built over the filtered dataset. Then, for an unknown instance, each classifier of the ensemble provides prediction for the labels on its own k-labelset, as shown in Figure 4. In the example in Figure 3, the first classifier included labels λ_2, λ_3 and λ_5, so in Figure 4 the first classifier gives a prediction for only those labels. Finally, the ratio of positive predictions for each label is calculated. If this ratio is greater than or equal to a given threshold (in the example, threshold = 0.5), the final prediction is 1 (relevant label) and 0 (irrelevant) otherwise. As seen in Figure 4 for a certain example, label λ_5 obtains one of four possible votes, so the final prediction is 0, while for label λ_6, which obtains four of five positive votes, the final prediction is 1.



	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
MLC_1	-	0	0	-	0	-
MLC_2	1	-	0	-	-	1
MLC_3	0	1	-	-	0	-
MLC_4	-	1	-	-	0	1
MLC_5	0	-	0	1	-	-
MLC_6	1	-	-	1	-	0
MLC_7	-	1	0	-	-	1
MLC_8	-	-	-	1	1	1
	$2/4$	$3/4$	$0/4$	$3/3$	$1/4$	$4/5$
threshold = 0.5	1	1	0	1	0	1

Figure 4: Example of the voting process of the ensemble (prediction threshold = 0.5).

3.3. Fitness function

The fitness function measures both the performance of the classifier and the number of times that each label appears in the ensemble, thus leading the evolution towards high-performing individuals that also consider all labels the same number of times regardless of their frequency.

Many evaluation measures for MLC have been proposed in the literature, some of them identified as non-decomposable measures [31]. The non-decomposable measures evaluate the multi-label prediction as a whole, unlike others that evaluate the prediction for each label separately. As one of the objectives of EME is to consider the relationship among the labels in the ensemble, a evaluation measures which also considers this fact is used. The Example-based FMeasure (ExF) is an approach to calculate the widely used FMeasure for MLC, and it is defined in Equation 1. The ExF is calculated for each instance, and then, the value is averaged among all the instances. ExF is defined in the range $[0, 1]$; the higher the value the better the performance of the algorithm. In the following, $\downarrow$ and $\uparrow$ indicate if the measures are minimized or maximized respectively.

$$\uparrow \text{ExF} = \frac{1}{m} \sum_{i=1}^{m} \frac{2|\hat{Y}_i \cap Y_i|}{|\hat{Y}_i| \cup |Y_i|} \tag{1}$$

Further, a coverage ratio measure (c_r), which evaluates the number of times that each label appears in the ensemble, has been defined. This measure is shown in Equation 2, being v the vector of votes, i.e. a vector storing the number of times that each label appears in the ensemble, v_w a vector of votes in the worst case, and $stdv(v)$ the standard deviation of the vector v. The worst case is the one where the vector of votes is as imbalanced as possible, i.e., some labels appearing in all classifiers and the rest not being present at all. In the case where all labels appears the same number of times in the ensemble, the vector of votes is homogeneous and the standard deviation is 0. Therefore, c_r is to be minimized. The coverage ratio is divided by the worst case in order to have a measure in the range $[0, 1]$. If c_r were not taken into account in the fitness, labels that are easier to predict would tend to appear more frequently in the individuals, causing others barely appearing.

$$\downarrow c_r = \frac{stdv(v)}{stdv(v_w)} \tag{2}$$

As an example, c_r for the case in Figure 4 is shown in Equation 3:

$$c_r = \frac{stdv(4, 4, 4, 3, 4, 5)}{stdv(8, 8, 8, 0, 0, 0)} = 0.1443 \tag{3}$$

Since both measures are in the range $[0, 1]$, but ExF is maximized and c_r is minimized, the fitness function is defined as the linear combination of them, as shown in Equation 4.

$$\uparrow fitness = \frac{\text{ExF} + (1 - c_r)}{2} \tag{4}$$

As all the multi-label ensembles of the population must be generated to calculate their fitness, evaluation is the process that consumes the most time. In order to reduce the runtime of the algorithm, two structures are created: one storing the fitness of each evaluated individual and other storing each multi-label classifier that was built. Thus, if an individual appears more than once, regardless of the order of its multi-label classifiers, the fitness is directly obtained from this structure, avoiding to evaluate a full ensemble. Further, if an individual which is going to be built contains a classifier that was previously built for other individual, this multi-label classifier does not have to be built again but it is directly obtained from the structure.

3.4. Genetic operators

In this section the crossover and mutation operators used in the evolutionary algorithm are described. Tournament selection is used to determine the individuals that form the set of parents. Then, each of these individuals is crossed or mutated based on crossover and mutation probabilities. The crossover and mutation operators are not mutually exclusive, i.e., an individual could be crossed and mutated in the same generation.

3.4.1. Uniform crossover operator

The uniform crossover operator swaps fragments of genotype of size q corresponding to multi-label classifiers between two parents. For each of the n fragments, the operator decides based on a probability (by default, 0.5) if the fragments in the same position in both parents are swapped. Figure 5 shows an example of the crossover operator, where the first and third classifiers are swapped between the parents. This operator makes each ensemble explore new combinations of classifiers that were already present in other individuals. The new individuals will always be valid, because neither the number of active bits of each classifier nor the number of classifiers are modified.

3.4.2. Phi-based mutation operator

We have proposed a phi-based mutation operator as a contribution of the paper. This operator swaps two bits of different value for each fragment corresponding to a base classifier in an individual, making each classifier of

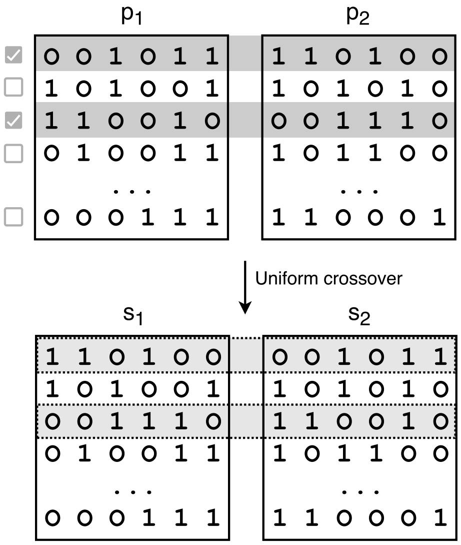


Figure 5: Uniform crossover operator.

the individual cease to classify one label to classify other. The bit swapping is performed considering the relationships among the labels, favoring the mutation to combinations of more related labels. In order to evaluate the relationships among the labels, the phi (ϕ) coefficient [32] that identifies the relationship between label pairs, is used. The phi coefficient is in the range $[-1, 1]$, 1 meaning total direct correlation, -1 total indirect correlation, and 0 no correlation.

Figure 6 shows an example of the phi-based mutation operator for a fragment of an individual. First, a random position corresponding to an active label is randomly selected (Figure 6a). Then, mutation weights w_b of each position b corresponding to each inactive label are calculated as shown in Equation 5. The weights are calculated by accumulating the values of ϕ between the corresponding labels and each label in A, being A the set of remaining active labels (Figure 6b). As the purpose is to evaluate the dependencies among the labels, regardless of whether positive or negative, the absolute value of phi is used. Also, a small value of ε is used to assign a small probability of mutating to the labels that are not correlated with the other active labels.

$$w_b = \varepsilon + \sum_{l \in A} |\phi_{b,l}| \tag{5}$$

Based on these weights, one of the inactive labels is selected to mutate (Figure 6c), where labels with a higher weight are more likely to be selected. Finally, the two selected positions are swapped (Figure 6d). Thereby, subsets of more related labels are more likely to be selected, but also keeping a small probability to search for less related combination of labels. The mutated individuals are always valid, because the number of active bits remains constant.

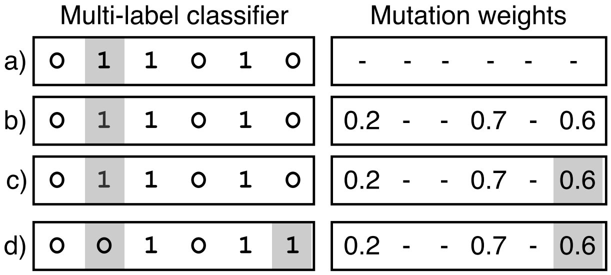


Figure 6: Phi-based mutation operator.

3.5. Time complexity

As previously stated, the most consuming process of the whole evolutionary algorithm is the evaluation of the individuals, since it requires to build each of the multi-label classifiers. The individuals are based on the use of C4.5 classifier, which complexity is $O(m \times d^2)$ [33], where m is the number examples and d is the number of features of the dataset. The complexity of EME is upper bounded by the total number of C4.5 classifiers that it has to evaluate. In each of the G generations, a total of *popSize* individuals, each composed by n C4.5 classifiers are evaluated. However, each base classifier that has been ever built is stored and EME does not have to build it again to evaluate an individual, so the number of classifier to build is usually drastically reduced. Besides, note that the number of possible C4.5 classifier to build is the same as the number of possible combinations of k labels given q. Thereby, the time complexity of EME is upper bounded by $O(m \times d^2 \times n_T)$, being n_T the number of C4.5 classifiers that could be build, defined as $n_T = \min(n \times popSize \times G, \binom{q}{k})$. Nevertheless, this asymptotic time complexity is reduced in the reality, since each individual that appears repeated in the population in any generation, is not evaluated and its fitness is directly obtained, avoiding to build an entire individual. Also note that the complexity of EME is directly related to the complexity of the base classifier used; using a different single-label classifier its complexity would vary.

4. Experimental studies

The purpose of the experimental studies is to compare EME to other state-of-the-art algorithms in multi-label classification over a wide range of datasets and evaluation measures. In this section the multi-label datasets and the evaluation measures used in the experiments are first presented, and then, the experimental settings are explained.

4.1. Datasets

The experiments were performed over a wide set of 14 reference datasets[1] from different domains, such as text categorization, multimedia, chemistry and biology. Table 2 lists the datasets along with their main characteristics, such as domain, number of instances (m), number of labels (q), number of features (d), cardinality ($card$, mean number of labels per instance) and density ($dens$, cardinality divided by the number of labels). The datasets are ordered by number of labels. The MLDA tool [34] was used for the characterization of the datasets.

4.2. Evaluation measures

In order to evaluate multi-label classification methods, many evaluation measures that take into account all labels were proposed [43]. We have based on the study of correlation among evaluation measures carried out in [44] to select the measures for the experiments.

Given its wide use in the evaluation of MLC methods in the literature, Hamming loss (HL) has been selected. It is a minimized measure which computes the average number of times that a label is incorrectly predicted. HL is defined in Equation 6, being Δ the symmetric difference between two binary sets. Subset Accuracy (SA) is a very strict evaluation measure which requires

[1] All the datasets and their descriptions are available at the repository in `http://www.uco.es/kdis/mllresources/`

Table 2: Datasets and their characteristics.

Dataset	Domain	m	q	d	*card*	*dens*	Ref
Emotions	Audio	72	6	72	1.868	0.311	[6]
Reuters1000	Text	294	6	1000	1.126	0.188	[35]
Guardian1000	Text	302	6	1000	1.126	0.188	[35]
Bbc1000	Text	352	6	1000	1.125	0.188	[35]
3s-inter3000	Text	169	6	3000	1.142	0.190	[35]
Gnegative	Biology	1392	8	440	1.046	0.131	[36]
Plant	Biology	948	12	440	1.080	0.089	[36]
Water-quality	Chemistry	1060	16	14	5.072	0.362	[37]
Yeast	Biology	2417	14	103	4.237	0.303	[38]
Human	Biology	3108	14	440	1.190	0.084	[36]
Birds	Audio	645	19	260	1.014	0.053	[39]
Slashdot	Text	3782	22	1079	1.180	0.053	[40]
Genbase	Biology	662	27	1186	1.252	0.046	[41]
Medical	Text	978	45	1449	1.245	0.028	[42]

the full multi-label prediction, including relevant and irrelevant labels, to be correctly predicted. SA is defined in Equation 7, where $[\![\pi]\!]$ returns 1 if predicate π is true and 0 otherwise. On the other hand, usually the labels that are most important, interesting or difficult to predict are the minority labels. Therefore, the macro approach, which gives the same importance to all labels in the evaluation has been selected to calculate label-based measures such as precision (MaP), recall (MaR) and specificity (MaS). These measures are formally described in Equations 8, 9, and 10 respectively, being tp_i, tn_i, fp_i, and fn_i the true positives, true negatives, false positives, and false negatives for the i-th label. Precision and recall are both based on measuring the prediction of relevant labels. The study in [44] did not consider the specificity measure; however, we consider that measuring the ratio of correctly predicted irrelevant labels should be also interesting.

$$\downarrow \text{HL} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{q}|Y_i \Delta \hat{Y}_i| \tag{6}$$

$$\uparrow \text{SA} = \frac{1}{m}\sum_{i=1}^{m}[\![Y_i = \hat{Y}_i]\!] \tag{7}$$

$$\uparrow \text{MaP} = \frac{1}{q}\sum_{i=1}^{q}\frac{tp_i}{tp_i + fp_i} \tag{8}$$

$$\uparrow \text{MaR} = \frac{1}{q}\sum_{i=1}^{q}\frac{tp_i}{tp_i + fn_i} \tag{9}$$

$$\uparrow \text{MaS} = \frac{1}{q}\sum_{i=1}^{q}\frac{tn_i}{tn_i + fp_i} \tag{10}$$

4.3. Experimental settings

The experimental study carried out was divided in two parts. First, EME was compared to other classic MLC methods such as BR, LP, CC, PS, and ChiDep. Then, in order to perform a more complete experimental study, EME was compared to other state-of-the-art EMLCs such as EBR, ECC, MLS, EPS, RAkEL, HOMER, and RF-PCT.

To compare the performance of the algorithms, the Friedman's test [45] was used for each evaluation measure. In cases where the Friedman's test indicated that there were significant differences in the performance of the algorithms with a 95% confidence, the Holm's post-hoc test [46] for comparisons of multiple classifiers involving a control method was performed. The adjusted p-values were used in the analysis, since they consider the fact of performing multiple comparisons without a significance level, providing more statistical information [47].

The experiments were carried out using a random 5-fold cross-validation and using 10 different seeds for those which use random numbers, such as CC, EBR, ECC, EPS, RAkEL, and RF-PCT. The default parameters, as originally recommended by their authors, were used in different algorithms employed. All methods use C4.5 as a single-label classifier. PS prunes the instances with labelsets occurring less than 3 times, and keeps the top two best ranked subsets when reintroducing the pruned instances. EPS is composed of 10 classifiers and sampling is done without replacement, keeping the rest of parameters as PS. RAkEL is composed of $2q$ classifiers and each with a subset of $k = 3$ labels. Both EBR and ECC are composed of 10 classifiers and use sampling with replacement. HOMER generates 3 clusters at each node and uses the *balanced k-means* clustering method. RF-PCT uses 10 trees in the ensemble, each with the full set of the training instances.

For C4.5 decision tree we used a minimum number of objects per leaf of 2, and a pruning confidence of 0.25. It should be noted that if the parameters of C4.5 were tuned for each specific case, the performance of EME should be improved. However, this improvement should be the same for the rest of state-of-the-art methods, so tuning the parameters of C4.5 is not the objective of this paper. In this way, we carry out a fair comparison among methods that use the same parameters of C4.5.

EME was implemented using JCLEC [48] and Mulan [49] frameworks, and the code is publicly available in a GitHub repository[2]. A brief study was carried out first in order to select the parameters of the evolutionary algorithm, such as the population size, number of generations, and crossover and mutation probabilities. For the parameters of the multi-label classifier in EME, they are similar to those proposed for RAkEL, each member of the ensemble has a subset of $k = 3$ labels, the ensemble is composed of $2q$ classifiers, and the prediction threshold is 0.5.

5. Results and discussion

In this section we present the experimental results. First the experimental study to select the parameters of EME is introduced, and then, the analysis and discussion of the two experiments carried out are presented, including the statistical tests performed for each of them. Te supplementary material available at the KDIS Research Group webpage[3] includes tables with the detailed results of all the experiments, including those of the selection of parameters of EME, more evaluation measures and runtime for the following experiments, and the average rankings and p-values of statistical tests.

5.1. Selection of parameters of EME

A brief study to select the parameters of the evolutionary algorithm in EME was carried out first. For these experiments, four datasets of different size were selected: Scene, Flags, PlantGO, and EukaryotePseAAC[4].

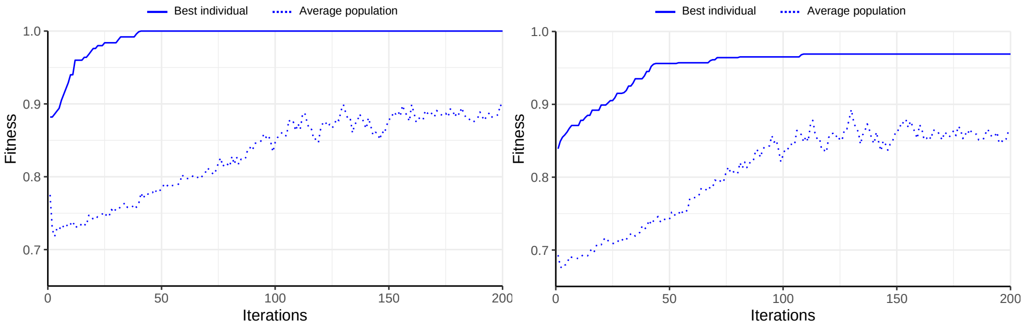
For the study of the size of the population and the number of generations, we analyzed the fitness value of the best individual in each generation,

[2] https://github.com/kdis-lab/EME

[3] http://www.uco.es/kdis/eme/

[4] All these datasets are different from those used in the rest of the experimental study, and are available at the repository in http://www.uco.es/kdis/mllresources/

as well as the average value of fitness of the whole population. For the smaller datasets (i.e., Scene and Flags, with 6 and 7 labels respectively), 50 individuals and a total of 200 generations were used. Figures 7a and 7b show the fitness value of the best individual and the average value of the whole population for Scene and Flags datasets respectively. For Scene dataset, we could see as the algorithm converged soon, obtaining the best value of fitness in early generations (less than 50); however, for Flags dataset the algorithm converged over the iteration 110, moment in which also the average fitness value of the population stabilized.

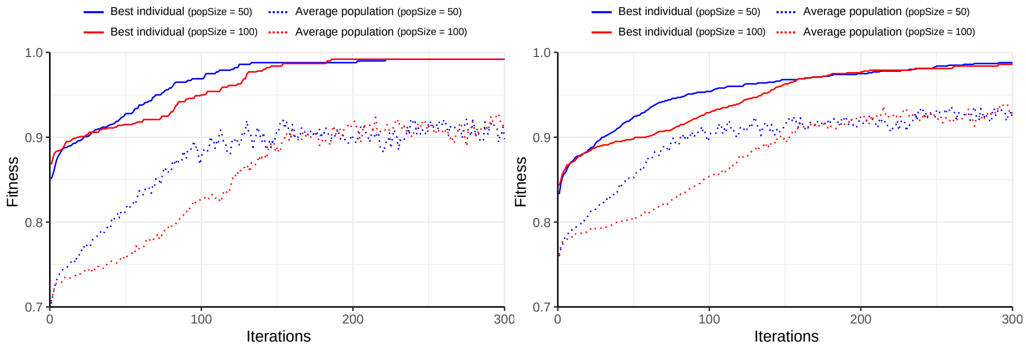


(a) Scene

(b) Flags

Figure 7: Variation in fitness of the best individual and the average value of fitness of the population for Scene and Flags datasets.

Secondly, for PlantGO and EukaryotePseAAC datasets, which have 14 and 22 labels respectively, the experiments were executed with 50 and 100 individuals in the population, and with a total of 300 generations, allowing enough time for the algorithm to stabilize. Figures 8b and 8a show the fitness value of the best individual and average population with both configurations for PlantGO and EukaryotePseAAC datasets respectively. For PlantGO, we can see that in early generations the configuration with 50 individuals achieved better fitness values, however, at the end of the evolution both configurations reached the same fitness for the best individual. On the other hand, the behavior for EukaryotePseAAC dataset is similar to PlantGO, but in this case the algorithm reached the best value in the last generations, where the configuration with 50 individuals obtained slightly better results. For both cases, the configuration with 50 individuals obtained the same or slightly better results, obtaining also a lower execution runtime.



(a) PlantGO (b) EukaryotePseAAC

Figure 8: Variation in fitness of the best individual and the average value of fitness of the population for PlantGO and EukaryotePseAAC datasets.

Given these results, for datasets with a small number of labels (≤ 8 labels), 50 individuals and a total of 110 generations were used. Further, for datasets where the label space is more complex (> 8 labels) and therefore the search space is much wider, also 50 individuals and a total of 300 generations were used.

On the other hand, the probability of crossover and mutate an individual could have a direct effect on the final performance of the algorithm, since their variation would vary the diversity of the population and could lead to a premature convergence of the algorithm or to never converge; therefore, an experimental study to select the optimal values for both probabilities was performed. For that, values of $pCross = \{0.7, 0.8, 0.9\}$ and $pMut = \{0.1, 0.2, 0.3\}$ were used. For both Scene and Flags datasets, the best results were obtained with $pCross = 0.9$, and $pMut = 0.2$, so we selected this configuration for small datasets. On the other hand, EukaryotePseAAC obtained the best performance with $pCross = 0.8$ and $pMut = 0.2$, while PlantGO also obtained competitive results with this configuration, so we selected it as default configuration for bigger datasets. The results of these experiments are fully available at the KDIS Research Group webpage[5].

5.2. Experiment 1: Comparing EME with other classic MLC algorithms

In this first experiment, EME is compared to other classic state-of-the-art MLC algorithms. The results of EME and the rest of state-of-the-art

[5] http://www.uco.es/kdis/eme/

algorithms over all datasets are shown in Tables 3, 4, 5, 6 and 7 for HL, SA, MaP, MaR, and MaS evaluation measures, respectively.

Table 3: Results of classic MLC algorithms for HL ↓ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**BR**	**LP**	**CC**	**PS**	**ChiDep**
Emotions	**0.220±0.016**	0.254±0.022	0.263±0.016	0.262±0.019	0.273±0.016	0.252±0.017
Reuters1000	**0.229±0.013**	0.257±0.004	0.268±0.019	0.284±0.022	0.276±0.025	0.257±0.004
Guardian1000	**0.228±0.015**	0.265±0.030	0.279±0.021	0.287±0.018	0.274±0.014	0.265±0.030
Bbc1000	**0.216±0.016**	0.263±0.015	0.264±0.013	0.284±0.016	0.270±0.010	0.267±0.017
3s-inter3000	**0.265±0.014**	0.308±0.026	0.312±0.009	0.311±0.030	0.314±0.030	0.308±0.026
Gnegative	**0.091±0.007**	0.120±0.011	0.119±0.007	0.122±0.009	0.118±0.011	0.123±0.011
Plant	**0.102±0.004**	0.139±0.008	0.141±0.006	0.141±0.005	0.144±0.005	0.139±0.006
Water-quality	**0.299±0.007**	0.310±0.007	0.375±0.008	0.334±0.009	0.337±0.011	0.315±0.012
Yeast	**0.210±0.007**	0.249±0.007	0.283±0.006	0.268±0.008	0.279±0.007	0.274±0.007
Human	**0.090±0.002**	0.121±0.002	0.126±0.002	0.122±0.003	0.123±0.002	0.121±0.001
Birds	**0.047±0.004**	0.052±0.010	0.063±0.004	0.052±0.008	0.054±0.006	0.052±0.010
Slashdot	**0.041±0.001**	0.043±0.001	0.054±0.001	0.052±0.007	0.053±0.001	0.042±0.001
Genbase	**0.001±0.001**	**0.001±0.001**	0.002±0.001	**0.001±0.000**	0.004±0.001	**0.001±0.001**
Medical	**0.010±0.001**	0.011±0.001	0.013±0.002	0.010±0.001	0.013±0.002	0.010±0.001

Table 4: Results of classic MLC algorithms for SA ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**BR**	**LP**	**CC**	**PS**	**ChiDep**
Emotions	**0.248±0.038**	0.170±0.047	0.226±0.034	0.218±0.040	0.209±0.056	0.191±0.036
Reuters1000	0.111±0.026	0.092±0.047	**0.207±0.055**	0.163±0.051	0.204±0.074	0.092±0.047
Guardian1000	0.086±0.035	0.069±0.040	0.166±0.050	0.147±0.049	**0.192±0.045**	0.069±0.040
Bbc1000	0.120±0.034	0.071±0.037	**0.207±0.040**	0.175±0.039	0.202±0.024	0.071±0.037
3s-inter3000	0.040±0.023	0.089±0.041	0.094±0.024	0.105±0.050	**0.106±0.057**	0.089±0.041
Gnegative	0.487±0.030	0.397±0.035	**0.522±0.031**	0.503±0.035	0.520±0.049	0.422±0.038
Plant	0.113±0.017	0.099±0.009	**0.189±0.033**	0.188±0.021	0.172±0.022	0.101±0.013
Water-quality	0.014±0.008	0.008±0.006	0.005±0.003	0.010±0.007	**0.015±0.011**	0.008±0.007
Yeast	0.137±0.013	0.070±0.009	0.135±0.014	**0.138±0.014**	0.131±0.007	0.113±0.011
Human	0.159±0.014	0.115±0.012	0.175±0.011	**0.192±0.014**	0.187±0.017	0.122±0.011
Birds	**0.496±0.048**	0.471±0.069	0.429±0.058	0.476±0.049	0.462±0.045	0.471±0.069
Slashdot	0.323±0.015	0.308±0.019	0.410±0.015	0.344±0.027	**0.412±0.021**	0.328±0.019
Genbase	**0.966±0.015**	0.965±0.015	0.965±0.016	0.965±0.013	0.937±0.019	0.965±0.015
Medical	0.649±0.037	0.635±0.045	0.661±0.044	0.664±0.040	**0.666±0.029**	0.665±0.035

For HL, EME performed the best in all cases, including a tie with BR, CC and ChiDep for Genbase dataset. For SA, the best results are more spread, where EME, LP, CC, and PS achieved the best result in many datasets. In the case of MaP, EME again shown the best performance in 11 out of 14 datasets. MaP and MaR are opposite measures, so good results in one of them usually lead to bad results in the other; for MaR LP achieved the best performance in seven datasets, while EME was the best in four, BR and

Table 5: Results of classic MLC algorithms for MaP ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**BR**	**LP**	**CC**	**PS**	**ChiDep**
Emotions	**0.657±0.039**	0.596±0.041	0.569±0.029	0.578±0.027	0.560±0.032	0.593±0.022
Reuters1000	0.235±0.068	0.215±0.051	**0.287±0.070**	0.211±0.073	0.256±0.106	0.215±0.051
Guardian1000	**0.284±0.088**	0.205±0.053	0.221±0.053	0.230±0.074	0.248±0.048	0.205±0.053
Bbc1000	**0.362±0.061**	0.262±0.045	0.293±0.058	0.244±0.079	0.291±0.043	0.256±0.050
3s-inter3000	0.107±0.048	0.167±0.086	0.174±0.038	**0.177±0.064**	0.154±0.055	0.167±0.086
Gnegative	**0.509±0.104**	0.317±0.029	0.366±0.125	0.316±0.040	0.335±0.056	0.300±0.027
Plant	**0.183±0.046**	0.142±0.021	0.144±0.014	0.142±0.028	0.123±0.036	0.143±0.016
Water-quality	**0.558±0.023**	0.521±0.027	0.446±0.014	0.500±0.024	0.354±0.048	0.510±0.024
Yeast	**0.510±0.029**	0.403±0.009	0.377±0.021	0.394±0.014	0.377±0.012	0.381±0.010
Human	**0.220±0.038**	0.163±0.014	0.129±0.007	0.143±0.011	0.132±0.019	0.158±0.014
Birds	0.396±0.071	**0.398±0.082**	0.318±0.086	0.386±0.090	0.318±0.048	**0.398±0.082**
Slashdot	**0.529±0.045**	0.518±0.055	0.430±0.029	0.500±0.053	0.434±0.043	0.514±0.071
Genbase	**0.929±0.050**	**0.929±0.056**	0.915±0.061	**0.929±0.050**	0.760±0.084	**0.929±0.056**
Medical	**0.651±0.054**	0.644±0.053	0.615±0.059	0.646±0.049	0.621±0.068	0.643±0.056

Table 6: Results of classic MLC algorithms for MaR ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**BR**	**LP**	**CC**	**PS**	**ChiDep**
Emotions	**0.592±0.025**	0.547±0.028	0.561±0.022	0.568±0.037	0.553±0.023	0.571±0.032
Reuters1000	0.131±0.034	0.178±0.062	**0.275±0.072**	0.201±0.053	0.235±0.069	0.178±0.062
Guardian1000	0.133±0.038	0.132±0.032	**0.223±0.054**	0.196±0.055	0.217±0.009	0.132±0.032
Bbc1000	0.164±0.039	0.136±0.055	**0.291±0.059**	0.202±0.036	0.263±0.033	0.129±0.057
3s-inter3000	0.078±0.042	0.173±0.108	**0.206±0.085**	0.191±0.081	0.187±0.092	0.173±0.108
Gnegative	0.352±0.083	0.343±0.078	**0.368±0.104**	0.340±0.067	0.327±0.052	0.330±0.078
Plant	0.082±0.016	**0.151±0.018**	0.137±0.018	**0.151±0.026**	0.125±0.049	**0.151±0.017**
Water-quality	**0.469±0.018**	0.429±0.023	0.451±0.021	0.445±0.026	0.179±0.009	0.440±0.027
Yeast	0.361±0.008	0.384±0.006	0.375±0.014	0.387±0.017	0.362±0.017	**0.388±0.011**
Human	0.095±0.011	**0.162±0.017**	0.130±0.011	0.148±0.017	0.134±0.018	0.155±0.017
Birds	0.228±0.048	0.269±0.053	**0.292±0.068**	0.264±0.059	0.226±0.025	0.269±0.053
Slashdot	0.319±0.027	0.313±0.031	**0.400±0.024**	0.336±0.029	0.398±0.024	0.314±0.033
Genbase	**0.934±0.045**	**0.934±0.050**	0.902±0.059	**0.934±0.045**	0.751±0.080	**0.934±0.050**
Medical	**0.650±0.056**	0.644±0.057	0.605±0.062	0.645±0.051	0.600±0.047	0.645±0.058

Table 7: Results of classic MLC algorithms for MaS ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**BR**	**LP**	**CC**	**PS**	**ChiDep**
Emotions	**0.858±0.014**	0.829±0.022	0.811±0.013	0.808±0.014	0.800±0.020	0.820±0.011
Reuters1000	**0.914±0.015**	0.865±0.018	0.833±0.016	0.822±0.020	0.834±0.019	0.865±0.018
Guardian1000	**0.918±0.017**	0.864±0.037	0.826±0.011	0.823±0.016	0.833±0.007	0.864±0.037
Bbc1000	**0.922±0.018**	0.864±0.022	0.831±0.009	0.824±0.012	0.833±0.011	0.862±0.024
3s-inter3000	**0.883±0.023**	0.801±0.026	0.799±0.009	0.795±0.032	0.801±0.018	0.801±0.026
Gnegative	**0.961±0.006**	0.922±0.013	0.922±0.005	0.919±0.005	0.923±0.007	0.917±0.009
Plant	**0.968±0.004**	0.916±0.009	0.916±0.003	0.914±0.004	0.915±0.001	0.916±0.008
Water-quality	0.786±0.016	0.782±0.016	0.687±0.008	0.750±0.023	**0.899±0.009**	0.770±0.020
Yeast	**0.803±0.006**	0.745±0.006	0.735±0.010	0.743±0.013	0.743±0.009	0.730±0.010
Human	**0.968±0.002**	0.924±0.002	0.923±0.001	0.924±0.002	0.925±0.002	0.925±0.003
Birds	**0.989±0.003**	0.982±0.005	0.968±0.004	0.982±0.003	0.982±0.003	0.982±0.005
Slashdot	**0.992±0.001**	0.991±0.001	0.972±0.001	0.978±0.009	0.973±0.001	0.991±0.002
Genbase	**1.000±0.000**	**1.000±0.000**	0.999±0.001	**1.000±0.000**	0.999±0.001	**1.000±0.000**
Medical	**0.995±0.001**	**0.995±0.001**	0.993±0.001	**0.995±0.001**	0.994±0.001	**0.995±0.001**

ChiDep in three each, and finally CC was the best in two datasets. MaP and MaR measures are both focused on relevant labels; on the other hand MaS measures the ratio of correctly predicted irrelevant labels. For MaS, EME performed the best in 13 out of 14 datasets, being the best method so far. Despite the opposition of evaluation measures, EME was able to achieve great performance in all of them.

The results of the Friedman's test for all evaluation measures, including the Friedman's statistics and the p-values are shown in Table 8. In four measures the Friedman's test determined that significant differences exists in the performance of the algorithms at 95% confidence, so the Holm's post-hoc test was also performed, and the adjusted p-values are shown in Table 9.

Table 8: Friedman's test results for the comparison with classic MLC algorithms. Values in bold indicate that there exist significant differences in the performance of the algorithms at 95% confidence.

	Statistic	***p*-value**
HL	45.42	**0.0000**
SA	27.61	**0.0000**
MaP	19.97	**0.0013**
MaR	8.81	0.1171
MaS	37.72	**0.0000**

For the four evaluation measures where the Friedman's test indicated that there were significant differences in the performance of the algorithms,

Table 9: Adjusted p-values of the Holm's test for the comparison with classic MLC algorithms. Algorithms marked with "-" are the control algorithm in each measure and values in bold indicates that there are significant differences with the control algorithm at 95% confidence.

	EME	**BR**	**LP**	**CC**	**PS**	**ChiDep**
HL	-	**0.0299**	**0.0000**	**0.0001**	**0.0000**	**0.0267**
SA	≥ 0.2	**0.0003**	≥ 0.2	≥ 0.2	-	**0.0160**
MaP	-	0.0617	**0.0041**	**0.0120**	**0.0002**	**0.0128**
MaS	-	**0.0339**	**0.0000**	**0.0000**	**0.0008**	**0.0080**

EME had the better performance in three of them. For HL and MaS, EME performed significantly better than the rest of methods, while for MaP it performed statistically better than all except BR. Further, for SA, where EME was not the control algorithm, it performed statistically equal than the control algorithm. These results showed that our algorithm has statistically better performance than the rest of classic MLC algorithms for all evaluation measures.

5.3. Experiment 2: Comparing EME with other EMLCs

Although EME has already shown that performs significantly better than classic state-of-the-art MLC algorithms, its performance was also compared to other state-of-the-art EMLCs. The results of EME and the rest of EMLCs over all datasets are shown in Tables 10, 11, 12, 13, and 14, again for HL, SA, MaP, MaR, and MaS measures.

Table 10: Results of EMLCs for HL ↓ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**ECC**	**EBR**	**RA*k*EL**	**EPS**	**HOMER**	**MLS**	**RF-PCT**
Emotions	0.220±0.016	**0.200±0.017**	0.202±0.015	0.225±0.015	0.209±0.012	0.253±0.015	0.259±0.013	0.213±0.017
Reuters1000	0.229±0.013	0.227±0.017	0.214±0.011	0.237±0.014	0.231±0.017	0.317±0.030	0.256±0.012	**0.205±0.012**
Guardian1000	0.228±0.015	0.225±0.016	0.210±0.014	0.239±0.022	0.226±0.016	0.288±0.018	0.264±0.019	**0.202±0.011**
Bbc1000	0.216±0.016	0.216±0.012	0.202±0.011	0.222±0.017	0.221±0.015	0.286±0.016	0.257±0.007	**0.197±0.010**
3s-inter3000	0.265±0.014	0.246±0.020	0.220±0.013	0.279±0.019	0.243±0.016	0.302±0.022	0.315±0.031	**0.207±0.014**
Gnegative	0.091±0.007	**0.082±0.007**	**0.082±0.007**	0.094±0.007	0.086±0.006	0.117±0.010	0.117±0.008	0.092±0.006
Plant	0.102±0.004	0.097±0.004	**0.093±0.003**	0.107±0.007	0.095±0.003	0.140±0.005	0.137±0.005	0.096±0.003
Water-quality	0.299±0.007	0.295±0.007	**0.290±0.007**	0.311±0.008	0.324±0.008	0.341±0.015	0.337±0.016	0.314±0.008
Yeast	0.210±0.007	0.210±0.006	**0.207±0.008**	0.225±0.008	0.210±0.007	0.263±0.008	0.273±0.005	0.219±0.007
Human	0.090±0.002	0.088±0.002	**0.085±0.002**	0.097±0.005	0.087±0.002	0.121±0.004	0.118±0.003	0.090±0.003
Birds	0.047±0.004	**0.043±0.006**	**0.043±0.006**	0.048±0.004	0.046±0.007	0.062±0.009	0.049±0.007	0.046±0.005
Slashdot	**0.041±0.001**	0.043±0.002	0.042±0.001	0.042±0.001	0.044±0.001	0.048±0.002	0.043±0.001	0.043±0.001
Genbase	**0.001±0.001**	**0.001±0.000**	**0.001±0.000**	**0.001±0.000**	0.004±0.001	**0.001±0.001**	**0.001±0.001**	0.046±0.003
Medical	**0.010±0.001**	**0.010±0.001**	0.011±0.001	0.011±0.001	0.012±0.001	0.011±0.002	0.011±0.001	0.025±0.001

For HL, EBR performed the best in seven datasets, while ECC in five, RF-PCT in four, EME in three and both RAkEL, HOMER and MLS in

Table 11: Results of EMLCs for SA ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**ECC**	**EBR**	**RA*k*EL**	**EPS**	**HOMER**	**MLS**	**RF-PCT**
Emotions	0.248±0.038	**0.297±0.036**	0.274±0.037	0.250±0.032	0.292±0.031	0.182±0.041	0.186±0.039	0.284±0.037
Reuters1000	0.111±0.026	0.064±0.031	0.040±0.026	**0.129±0.029**	0.115±0.035	0.078±0.031	0.112±0.010	0.045±0.022
Guardian1000	0.086±0.035	0.063±0.030	0.037±0.020	0.092±0.036	**0.130±0.040**	0.069±0.036	0.076±0.055	0.037±0.026
Bbc1000	0.120±0.034	0.086±0.034	0.057±0.024	0.134±0.028	**0.142±0.042**	0.102±0.051	0.088±0.024	0.045±0.022
3s-inter3000	0.040±0.023	0.050±0.029	0.025±0.026	0.037±0.022	0.044±0.035	0.042±0.027	**0.077±0.044**	0.033±0.032
Gnegative	0.487±0.030	**0.548±0.032**	0.497±0.031	0.493±0.030	0.513±0.027	0.421±0.023	0.397±0.037	0.470±0.030
Plant	0.113±0.017	**0.140±0.024**	0.089±0.020	0.127±0.029	0.095±0.018	0.094±0.015	0.109±0.028	0.101±0.018
Water-quality	0.014±0.008	**0.017±0.010**	0.016±0.009	0.013±0.008	0.015±0.009	0.004±0.004	0.008±0.004	0.012±0.008
Yeast	0.137±0.013	**0.171±0.016**	0.131±0.014	0.112±0.015	0.168±0.015	0.076±0.011	0.051±0.008	0.145±0.014
Human	0.159±0.014	**0.174±0.011**	0.141±0.013	0.167±0.018	0.140±0.013	0.105±0.004	0.122±0.007	0.127±0.012
Birds	0.496±0.048	**0.522±0.054**	0.516±0.055	0.490±0.045	0.515±0.055	0.457±0.049	0.491±0.053	0.503±0.057
Slashdot	0.323±0.015	0.330±0.021	0.303±0.016	0.314±0.024	**0.399±0.008**	0.309±0.021	0.310±0.020	0.252±0.013
Genbase	0.966±0.015	0.968±0.013	0.967±0.013	0.965±0.014	0.937±0.018	**0.970±0.009**	0.967±0.016	0.000±0.000
Medical	0.649±0.037	0.671±0.030	0.650±0.025	0.641±0.040	**0.674±0.024**	0.654±0.052	0.637±0.044	0.085±0.038

Table 12: Results of EMLCs for MaP ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**ECC**	**EBR**	**RA*k*EL**	**EPS**	**HOMER**	**MLS**	**RF-PCT**
Emotions	0.657±0.039	0.685±0.033	**0.704±0.034**	0.640±0.032	0.673±0.029	0.588±0.017	0.588±0.026	0.647±0.033
Reuters1000	0.235±0.068	0.170±0.087	0.134±0.090	0.243±0.048	**0.244±0.089**	0.165±0.030	0.208±0.061	0.137±0.099
Guardian1000	**0.284±0.088**	0.166±0.070	0.133±0.083	0.250±0.076	0.272±0.114	0.242±0.049	0.202±0.070	0.120±0.111
Bbc1000	**0.362±0.061**	0.216±0.102	0.214±0.123	0.353±0.077	0.359±0.098	0.248±0.052	0.267±0.069	0.179±0.117
3s-inter3000	0.107±0.048	0.139±0.092	0.094±0.090	0.144±0.061	0.090±0.063	**0.165±0.074**	0.157±0.080	0.117±0.107
Gnegative	**0.509±0.104**	0.495±0.094	0.499±0.082	0.476±0.091	0.473±0.094	0.369±0.087	0.349±0.050	0.406±0.087
Plant	0.183±0.046	0.178±0.051	0.189±0.057	**0.190±0.054**	0.170±0.061	0.145±0.028	0.157±0.023	0.135±0.041
Water-quality	0.558±0.023	0.556±0.024	**0.573±0.024**	0.536±0.024	0.281±0.049	0.500±0.023	0.498±0.029	0.522±0.022
Yeast	0.510±0.029	0.495±0.037	**0.515±0.034**	0.463±0.029	0.505±0.054	0.389±0.014	0.392±0.007	0.465±0.035
Human	0.220±0.038	0.222±0.035	**0.224±0.040**	0.206±0.033	0.211±0.053	0.143±0.012	0.171±0.018	0.182±0.043
Birds	0.396±0.071	0.431±0.078	0.420±0.082	0.398±0.086	0.330±0.066	0.318±0.049	0.408±0.127	**0.432±0.078**
Slashdot	0.529±0.045	0.522±0.044	0.521±0.051	0.524±0.045	**0.535±0.028**	0.463±0.052	0.506±0.034	0.477±0.024
Genbase	**0.929±0.050**	0.923±0.053	0.921±0.053	0.925±0.053	0.768±0.078	0.921±0.071	**0.929±0.056**	0.217±0.104
Medical	**0.651±0.054**	0.645±0.055	0.647±0.063	0.645±0.049	0.625±0.061	0.627±0.055	0.646±0.057	0.379±0.067

Table 13: Results of EMLCs for MaR ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**ECC**	**EBR**	**RA*k*EL**	**EPS**	**HOMER**	**MLS**	**RF-PCT**
Emotions	0.592±0.025	0.585±0.033	0.590±0.029	0.627±0.034	0.621±0.025	0.606±0.048	0.575±0.024	**0.667±0.031**
Reuters1000	0.131±0.034	0.084±0.042	0.045±0.026	0.158±0.032	0.130±0.034	0.158±0.029	**0.162±0.044**	0.044±0.022
Guardian1000	0.133±0.038	0.077±0.029	0.041±0.019	0.163±0.040	0.142±0.040	**0.226±0.048**	0.141±0.040	0.040±0.025
Bbc1000	0.164±0.039	0.086±0.028	0.057±0.023	0.177±0.038	0.163±0.039	**0.235±0.052**	0.164±0.032	0.039±0.019
3s-inter3000	0.078±0.042	0.069±0.038	0.031±0.026	0.098±0.050	0.053±0.036	0.165±0.060	**0.166±0.094**	0.035±0.031
Gnegative	0.352±0.083	0.341±0.071	0.299±0.067	**0.386±0.087**	0.298±0.060	0.374±0.060	0.360±0.083	0.255±0.061
Plant	0.082±0.016	0.069±0.014	0.051±0.015	0.101±0.020	0.046±0.013	0.140±0.023	**0.165±0.032**	0.042±0.009
Water-quality	0.469±0.018	0.519±0.014	0.465±0.015	0.525±0.022	0.148±0.011	0.579±0.042	0.453±0.017	**0.587±0.018**
Yeast	0.361±0.008	0.389±0.009	0.351±0.009	0.405±0.017	0.358±0.008	**0.406±0.017**	0.394±0.020	0.402±0.010
Human	0.095±0.011	0.086±0.008	0.066±0.007	0.118±0.016	0.064±0.006	0.141±0.013	**0.157±0.026**	0.057±0.005
Birds	0.228±0.048	0.222±0.055	0.201±0.048	0.246±0.050	0.198±0.047	**0.295±0.057**	0.271±0.074	0.217±0.043
Slashdot	0.319±0.027	0.321±0.023	0.305±0.022	0.317±0.029	**0.399±0.018**	0.325±0.028	0.314±0.035	0.232±0.020
Genbase	0.934±0.045	0.929±0.052	0.926±0.049	0.931±0.048	0.759±0.075	0.912±0.065	**0.935±0.050**	0.216±0.104
Medical	**0.650±0.056**	0.646±0.054	0.641±0.060	0.645±0.052	0.600±0.059	0.598±0.063	0.646±0.055	0.335±0.058

Table 14: Results of EMLCs for MaS ↑ measure and standard deviations. Values in bold indicate the best results for each dataset.

	EME	**ECC**	**EBR**	**RA*k*EL**	**EPS**	**HOMER**	**MLS**	**RF-PCT**
Emotions	0.858±0.014	0.861±0.014	**0.881±0.012**	0.834±0.016	0.856±0.012	0.805±0.013	0.810±0.012	0.828±0.014
Reuters1000	0.914±0.015	0.924±0.017	0.952±0.011	0.896±0.014	0.910±0.017	0.793±0.042	0.867±0.014	**0.964±0.014**
Guardian1000	0.918±0.017	0.929±0.016	0.958±0.013	0.897±0.021	0.910±0.017	0.822±0.019	0.863±0.016	**0.969±0.011**
Bbc1000	0.922±0.018	0.936±0.015	0.964±0.010	0.911±0.018	0.912±0.014	0.817±0.020	0.865±0.010	**0.974±0.011**
3s-inter3000	0.883±0.023	0.907±0.023	0.952±0.016	0.863±0.024	0.918±0.015	0.811±0.026	0.794±0.023	**0.967±0.014**
Gnegative	0.961±0.006	0.964±0.004	**0.973±0.004**	0.949±0.008	0.968±0.004	0.922±0.007	0.922±0.009	0.964±0.004
Plant	0.968±0.004	0.974±0.004	**0.985±0.002**	0.959±0.012	0.982±0.003	0.915±0.007	0.917±0.003	0.979±0.004
Water-quality	0.786±0.016	0.759±0.018	0.800±0.017	0.735±0.021	**0.940±0.007**	0.662±0.046	0.741±0.029	0.685±0.018
Yeast	0.803±0.006	0.774±0.006	**0.804±0.005**	0.761±0.013	0.786±0.005	0.727±0.017	0.743±0.010	0.746±0.005
Human	0.968±0.002	0.969±0.002	**0.981±0.002**	0.954±0.008	0.975±0.002	0.927±0.003	0.927±0.003	0.972±0.003
Birds	0.989±0.003	0.993±0.002	**0.995±0.001**	0.986±0.003	0.992±0.003	0.970±0.006	0.985±0.005	0.991±0.002
Slashdot	0.992±0.001	0.989±0.002	0.992±0.001	0.992±0.001	0.985±0.001	0.984±0.002	0.990±0.001	**0.997±0.001**
Genbase	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	0.999±0.001	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
Medical	0.995±0.001	0.994±0.001	0.995±0.001	0.995±0.001	0.994±0.001	0.995±0.001	0.995±0.001	**0.999±0.001**

one tied. As can be shown, the performance of EME in HL is better for datasets where the number of labels is greater. That means that when the label space is wider, EME tends to predict correctly, in average, a greater number of labels than the rest of methods. This is given by the fact that for cases where a greater number of different possible combinations of k-labelsets are available, EME is able to obtain a good combination of subsets of labels with a great performance. It can be also seen as EME obtained a better performance than RAkEL in all cases, enhancing the need for optimizing the combination of k-labelsets instead of only making a random selection. In the case of SA, EME did not achieved the best results in any dataset for such a strict measure, in which ECC was the best in seven datasets. SA evaluates the ratio of multi-label predictions where both the relevant and irrelevant labels were exactly predicted. Although it is an interesting evaluation measure in some cases, it must be interpreted cautiously since it does not consider partially correct predictions. For example, a method with a low value of SA could be predicting the rest of instances with almost all relevant labels, while a method with a higher value of SA could be predicting completely bad the rest of instances. For MaP, EME was the best in five datasets, followed by EBR being the best in four. ECC, which achieved better results in other measures, was not the best in any case for MaP. Further, although MaP and MaR are opposite measures, ECC did not achieve great results in MaR either, being the best in only one case. On the other hand, EME was the best in only one case in MaR but achieved better results for MaP, which is an expected behavior. Both ECC and EBR were not the best in any case for this measure. Finally, for MaS the better results were spread between EBR and RF-PCT, being the best in seven datasets each, while the rest in only one.

As in the previous experiment, first Friedman's test was performed in order to know if there were significant differences on the performance of the algorithms. The results of Friedman's test are shown in Table 15, indicating that significant differences exist for all measures at 95% confidence. Therefore, the post-hoc Holm's test was performed for all the measures. The results, including the adjusted p-values are shown in Table 16.

Table 15: Friedman's test results for the comparison with state-of-the-art EMLCs. Values in bold indicate that there exist significant differences in the performance of the algorithms at 95% confidence.

	Statistic	***p*-value**
HL	52.99	**0.0000**
SA	31.65	**0.0000**
MaP	26.74	**0.0004**
MaR	42.49	**0.0000**
MaS	50.84	**0.0000**

Table 16: Adjusted p-values of the Holm's test for the comparison among state-of-the-art EMLCs. Algorithms marked with "-" are the control algorithm in each measure and values in bold indicates that there are significant differences with the control algorithm at 95% confidence.

	EME	**ECC**	**EBR**	**RA*k*EL**	**EPS**	**HOMER**	**MLS**	**RF-PCT**
HL	0.1016	≥ 0.2	-	**0.0014**	**0.0436**	**0.0000**	**0.0000**	0.1016
SA	0.1613	-	**0.0081**	0.1613	≥ 0.2	**0.0013**	**0.0052**	**0.0003**
MaP	-	≥ 0.2	≥ 0.2	≥ 0.2	0.1636	**0.0007**	**0.0308**	**0.0007**
MaR	≥ 0.2	**0.0436**	**0.0001**	≥ 0.2	**0.0030**	-	≥ 0.2	**0.0002**
MaS	0.0758	0.0758	-	**0.0011**	0.0745	**0.0000**	**0.0000**	≥ 0.2

Although EME was the best performing algorithm for only one evaluation measure, it was the only one that did not have significant differences with the control algorithm in any measure. ECC and EBR, which achieved great results in some evaluation measures, being the control algorithm in one and two cases respectively, also had a significantly poor performance than the control algorithm in some cases, such as for SA and MaR. These results showed that EME is more consistent in overall performance than other state-of-the-art EMLCs over all measures, and did not perform significantly worse than the rest in any case. EME achieved high predictive performance compared not only with classic MLC algorithms, but also when compared with other EMLCs.

Further, EME had a better overall performance than RAkEL in four of the five measures, including HL and MaS, where RAkEL performed significantly worse than the control algorithm. This indicates that the fact of not only selecting the k-labelsets randomly as RAkEL does, but also evolving towards a more promising combination of k-labelsets in the ensemble makes the model to achieve a better predictive performance.

6. Conclusions

In this paper we presented an evolutionary algorithm for the automatic generation of ensembles of multi-label classifiers based on projections of labels, taking into account the relationships among the labels but avoiding a high complexity. Each individual in the evolutionary algorithm encodes an ensemble of multi-label classifiers, which are evaluated taking into account both the predictive performance of the individual and the number of times that each label appears in the ensemble. The evolutionary algorithm helps to obtain a promising and high-performing combination of multi-label classifiers into an ensemble.

The experiments over a wide set of fourteen datasets and five evaluation measures showed that our algorithm performed statistically better than classic MLC methods and also had a more consistent performance than other other state-of-the-art EMLCs. EME obtained the best results in several cases, and although not being always the first algorithm in the ranking, EME was the only algorithm that did not perform significantly worse than the rest in any case. Further, the experimental results also show that the fact of evolving the individuals toward more promising combinations of multi-label classifiers achieves better results than just selecting them randomly, as RAkEL does.

As future work, we aim to extend EME to use a variable number of labels (k) in each of the classifiers of the ensemble and to explore other ways to combine the predictions of the classifiers to create the final ensemble prediction. Further, we we aim to perform an optimization and tuning of the parameters of the single-label classifier in order to improve the performance of the final multi-label classifier. Finally, we aim to explore some multi-objective fitness functions that may improve the performance of EME, instead of selecting only one.

Acknowledgements

This research was supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund, project TIN2017-83445-P. This research was also supported by the Spanish Ministry of Education under FPU Grant FPU15/02948.

References

[1] L. Tang, H. Liu, Scalable learning of collective behavior based on sparse social dimensions, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 09), 2009, pp. 1107–1116.

[2] G. Nasierding, A. Kouzani, Image to text translation by multi-label classification, in: Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence, Vol. 6216, 2010, pp. 247–254.

[3] E. Loza, J. Fürnkranz, Efficient multilabel classification algorithms for large-scale problems in the legal domain, in: Semantic Processing of Legal Texts, Vol. 6036, 2010, pp. 192–215.

[4] 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.

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

[6] 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), Vol. 21, 2008, pp. 53–59.

[7] J. Read, A pruned problem transformation method for multi-label classification, in: Proceedings of the NZ Computer Science Research Student Conference, 2008, pp. 143–150.

[8] 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.

[9] K. Laghmari, C. Marsala, M. Ramdani, An adapted incremental graded multi-label classification model for recommendation systems, Progress in Artificial Intelligence 7 (1) (2018) 15–29.

[10] R. Sousa, J. Gama, Multi-label classification from high-speed data streams with adaptive model rules and random rules, Progress in Artificial Intelligence 7 (3) (2018) 177–187.

[11] J. M. Moyano, E. L. Gibaja, K. J. Cios, S. Ventura, Review of ensembles of multi-label classifiers: Models, experimental study and prospects, Information Fusion 44 (2018) 33 – 45.

[12] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Information Fusion 6 (1) (2005) 5 – 20.

[13] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2) (2003) 181–207.

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

[15] 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.

[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] K. Dembczynski, W. Cheng, E. Hüllermeier, Bayes optimal multilabel classification via probabilistic classifier chains, in: ICML, Vol. 10, 2010, pp. 279–286.

[18] E. Goncalves, A. Plastino, A. A. Freitas, Simpler is better: a novel genetic algorithm to induce compact multi-label chain classifiers., in: 2015 Conference on Genetic and Evolutionary Computation Conference (GECCO-2015), 2015, pp. 559–566.

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Ensembles of multi-objective decision trees, in: European conference on machine learning, Springer, 2007, pp. 624–631.

[27] K. Deb, An introduction to genetic algorithms, Sadhana 24 (4) (1999) 293–315.

[28] D. Thierens, Selection schemes, elitist recombination, and selection intensity, in: Proceedings of the 7th International Conference on Genetic Algorithms, 1998, pp. 152–159.

[29] L. Rokach, A. Schclar, E. Itach, Ensemble methods for multi-label classification, Expert Systems with Applications 41 (16) (2014) 7507 – 7523.

[30] 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.

[31] K. Dembczyński, W. Waegeman, W. Cheng, E. Hüllermeier, On label dependence and loss minimization in multi-label classification, Machine Learning 88 (1) (2012) 5–45.

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

[33] J. Su, H. Zhang, A fast decision tree learning algorithm, in: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1, AAAI'06, 2006, pp. 500–505.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

[42] 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.

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

[44] R. B. Pereira, A. Plastino, B. Zadrozny, L. H. Merschmann, Correlation analysis of performance measures for multi-label classification, Information Processing & Management 54 (3) (2018) 359 – 369.

[45] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Statist. 11 (1) (1940) 86–92.

[46] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian journal of statistics (1979) 65–70.

[47] 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.

[48] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, C. Hervás, JCLEC: a java framework for evolutionary computation, Soft Computing 12 (4) (2008) 381–392.

[49] 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.