IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 May 2022, accepted 3 July 2022, date of publication 20 July 2022, date of current version 26 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192642

RESEARCH ARTICLE

Reducing the Label Space a Predefined Ratio for a More Efficient Multilabel Classification

JOSE M. MOYANO, JOSE M. LUNA, AND SEBASTIAN VENTURA, (Senior Member, IEEE)
Department of Computer Science and Numerical Analysis, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Córdoba, 14071 Córdoba, Spain

Corresponding author: Sebastian Ventura (sventura@uco.es)

This research has been supported by Grant PID2020-115832GB-I00 funded by MICIN/AEI/10.13039/501100011033.

ABSTRACT The multi-label classification task has been widely used to solve problems where each of the instances may be related not only to one class but to many of them simultaneously. Many of these problems usually comprise a high number of labels in the output space, so learning a predictive model from such datasets may turn into a challenging task since the computational complexity of most algorithms depends on the number of labels. In this paper, we propose a methodology to reduce the label space a user predefined ratio of labels, aiming to improve the runtime of the multi-label classification algorithms. Obviously, such reduction should be done without producing a significant drop in their final predictive performance. The experimental analysis carried out over 25 well-known multi-label datasets, demonstrates a drastic reduction in the runtime. Besides, it is statistically proven that reducing 20% the number of labels does not lead to a decrease in the predictive performance of the multi-label algorithms using four well-known evaluation measures. Even more, in many cases, although reductions of up to 50% of the output space are made, the predictive performance of the algorithms is not significantly different from using the whole set of labels.

INDEX TERMS Algorithm efficiency, binary classification, dimensionality reduction, label space reduction, multi-label classification.

I. INTRODUCTION

Multi-label classification (MLC) is a supervised learning task, where, in contrast to classical supervised learning, each instance of the data may be associated with more than a single label simultaneously [1]. For years, MLC has been an area of great interest in the research community, due to its good applicability to a wide variety of real-world problems, such as text categorization [2], image classification [3] and medicine [4]. Nevertheless, such domains are usually inherently complex in the output space, i.e., they comprise a high number of labels. It should be noted that complex label spaces use to imply a high computational cost to build an accurate classifier, giving rise to one of the key challenges of MLC: dealing with high-dimensional or complex output spaces. Nowadays, this challenge is considered further aggravated with the emergence of problems where the number of labels is so extremely large that the classical methods cannot be applied successfully, leading to the so-called eXtreme MLC [5].

Many existing successful MLC methods are based on the decomposition of the multi-label problem into several single-label learning problems. Most common algorithms in this category are Binary Relevance (BR) and Label Powerset (LP) [6]. Whereas the former generates as many independent binary classification problems as labels exist in the original multi-label problem, the latter transforms the multi-label problem into a multi-class one, where each combination of labels is considered as a different class. As their complexity is dependent on the number of labels, they do not usually scale well for problems where the number of labels is extremely large. In such a situation, BR may require the training of a large number of different binary classifiers, whereas LP may generate a complex multi-class classification problem. Thus, dealing with domains that include a huge label space still tend to be a challenge for the vast majority of the MLC methods

The associate editor coordinating the review of this manuscript and approving it for publication was Julien Le Kernec.

that have been proposed so far, specially when considering that many of the existing MLC algorithms are designed upon the basis of BR and LP [7].

A possible solution to address the problems arisen when dealing with large label spaces lies in the design of methods that effectively reduce such spaces. Output dimensionality reduction in MLC has been addressed from different points of view, ranging from those that reduce both the label and the feature spaces simultaneously [7], [8], to those that only reduce the label space, either considering [9] or not [10], [11] the input features to achieve such objective. Focusing on the methods that reduce the label space disregarding the input set of features, existing approaches usually perform the following three main phases: (I) reduce the label space; (II) learn a function to predict the labels on the reduced dataset; (III) infer the whole label space of the original dataset. These methods, however, do not usually produce a predefined reduction ratio, and this ratio is not known till the whole process is performed. Thus, the user (domain expert) does not have any control on the reduction level to be performed and, as a result, the percentage of reduction might highly differ from dataset to dataset. Besides, some of the existing methods for label space reduction are presented as a MLC method itself, i.e., they do not allow to use different MLC algorithms to solve the problem.

Given the problems identified, the objective of this paper is to propose a methodology that enables the user to perform a predefined ratio of reduction in the label or output space of the multi-label data, allowing to run MLC algorithms more efficiently, while providing outputs for the entire original set of labels. Such reduction should improve the overall performance of MLC algorithms, not only reducing the runtime required by classifiers at both training and testing phases, but also without causing a significant reduction in their predictive ability. Of course, a huge reduction of the label space might give rise to a significant loss of information even when it may allow to run the MLC algorithm in a reasonable quantum of time.

In this way, the contributions of this research work can be summarized as follows:

1) A methodology that does not consider at all the input features, focusing only on removing those single labels that are better estimated by the rest is proposed. Unlike other methods in the literature, it allows both to reduce the label space a predefined ratio (which is not problem-dependent), and to run any MLC algorithm of the user's choice. As a result, accurate but computationally hard algorithms that could not be previously run on datasets with a large number of labels could be used.
2) The reduction phase is conceived as an iterative process, where one label is removed at a time. This process is therefore repeated as many times as the desired percentage of labels to be reduced. After any multi-label classification algorithm is applied on the reduced dataset, the inverse problem is carried out; that is, a series of labels are predicted (and added to the label space) from the reduced set of labels. Therefore, the original structure of the output space is maintained.
3) An extensive experimental study is carried out over 25 multi-label datasets, considering three different MLC algorithms, and using four evaluation measures. Reductions ranging from 10% of the label space, up to 50% have been considered. Both the operation of the label reduction methodology and the performance of the MLC algorithms on the reduced datasets are studied.

According to the study, focusing on the predictive performance of the binary classifiers used in the reduction phase, and as it was expected, results demonstrate a reduction in their performance when the number of labels is lower and lower. It denotes that is the user who decides and assumes the risk of significantly losing predictive performance for large reductions. Experimental results have also demonstrated that reductions of about 20% of the labels do not imply a significant reduction in the final predictive performance in any of the three different MLC algorithms considered. In some scenarios, it was possible to reduce up to 40% or 50% of the labels without a statistical difference in the final predictive performance, but drastically decreasing the required runtime. Situations in which the MLC algorithms obtained a better predictive performance with the reduced dataset than with the original one were also found.

The rest of the paper is organized as follows. Section II presents some background in MLC and label space reduction methods. Section III describes the proposed methodology for reducing the label space. Section IV includes the experimental study, describing the achieved results as well as the lesson learned. Finally, Section V presents the conclusions obtained from this work.

II. BACKGROUND

In this section, the MLC problem is formally defined as well as main approaches to address it. On the other hand, the label space dimensionality problem is also described, including methodologies proposed in the literature to tackle it.

A. MULTI-LABEL CLASSIFICATION

A multi-label problem comprises a d-dimensional feature space $\mathcal{F}$ and a q-dimensional label space $\mathcal{L} \in \{0, 1\}^q$, where d and q represent the number of input features and output labels, respectively. A multi-label example i can be represented as a tuple $\langle \mathbf{X}_i, \mathbf{Y}_i \rangle$, where $\mathbf{X}_i \in \mathcal{F}$ is the feature vector and $\mathbf{Y}_i \in \mathcal{L}$ the output or label vector of the example i. The output $\mathbf{Y}_i$ of a given example is usually represented as a binary vector that contains q components, where the component $Y_{i\ell}$ represents whether the example i is associated with the ℓth label (1) or not (0). The goal of any MLC problem is to learn a function $f : \mathcal{F} \rightarrow \mathcal{L}$, which can predict a label vector $\hat{\mathbf{Y}}$ given an unseen point $\mathbf{X}$ [6].

To date, many MLC algorithms have been proposed and they can be classified into three main categories: Problem Transformation Methods (PTMs), Algorithm Adaptation

Methods (AAMs), and Ensembles of Multi-Label Classifiers (EMLCs) [12]. PTMs decompose the multi-label problem into one or several single-label classification problems. A major advantage of PTMs is that classic and sophisticated existing single-label classifiers can be employed to solve the MLC problem. On the contrary, their main drawback is related to the computational complexity, which increases with the number of labels. For example, given a multi-label problem with q labels, BR require the construction of q binary classifiers, while LP compose a multi-class classification problem with up to 2^q distinct classes. Besides, it should be highlighted that BR and LP have been the basis of many other methods in the literature [12]–[14], so most of them share the aforementioned characteristics.

Focusing on AAMs, they comprise approaches designed to directly handle the multi-label data without having to transform the original problem. For these methods, the computational complexity is not so dependent on the number of labels as the case of PTMs; for example ML-kNN [15] adapts the popular k-Nearest Neighbors algorithm but retrieving information of the full label set for each example, while IBLR-ML [16] expands the feature space with the label information of the k nearest neighbors. Finally, EMLCs gather proposals that combine the output of several multi-label classifiers. The members of the ensemble can be either PTMs, AAMs, or a mixture of both. Therefore, the computational complexity of EMLCs depend on the the used base classifiers.

At this point, it should be highlighted that many of the existing MLC methods are impractical to be used on large-scale multi-label datasets, which comprise a considerable number of labels. Thus, a possible solution is to reduce the label space of the problem as it is described in the next subsection, where the main proposals that have appeared in the literature are analyzed.

B. LABEL SPACE DIMENSIONALITY PROBLEM

In traditional classification tasks, the problem of high dimensionality has been widely studied in the space of features with promising results [17]. Nevertheless, when dealing with MLC scenarios, the problem of dimensionality reduction becomes harder to tackle, and it can be analyzed from different perspectives: reducing the feature space [18], reducing the label space [19], or reducing both input and output spaces simultaneously [20].

While the dimensionality reduction on feature space has widely been studied on multi-label data [18], the problem of reducing dimensionality in the label space is not well-studied yet, even when recent studies can be found in the literature. In MLC, the number of labels is highly related to both the required runtime and the simplicity of the resulting model, so in cases with a high-dimensional output space, reducing the label space could be determinant both in terms of performance and runtime. Based on that, in this study we center our attention on the label space reduction methods.

Approaches to reduce the label space can be mainly categorized into two main groups: methods that consider the information carried by the features to reduce the label space, and those that do not pay attention to the input features for reducing the label space. The feature-dependent methods leverage the correlation that may exist between features and labels to reduce the label space. However, it is expected that, if a feature selection is not carried out while the reduction process is performed, then the estimation of the labels could be biased by redundant and noisy features. Also, it should be considered that the process for reducing the label space could have a high computational cost on datasets with tens of thousands of input features (which is very common). In this paper, we analyze the second group of methods; i.e. methods that reduce the label space disregarding the feature space.

One of the first feature-independent reduction method was proposed by Hsu *et al.* [19], where an approach based on a proven compression technique was designed. The so-called *Compressed Sensing* (CS) states that the complexity of a model with q labels can be reduced to the training of $O[log(q)]$ simpler models. This proposal requires a significant level of sparsity in the label space so it is really useful for problems with a large number of different labels, but with only a small subset of them appearing in each instance. In this proposal, the compression phase is made by projections of the original label space to obtain a representation in a real (non-binary) lower dimensionality space. Afterward, these projections are used to train a set of regression models and the outputs are decompressed to obtain the labels predicted for each sample. However, as previously stated, it considers that datasets have low density [21], but not all multi-label scenarios fulfill such condition.

Later, Tai and Lin [22] proposed the method named *Principal Label Space Transformation* (PLST), which relies on the popular singular value descomposition technique. In this case, the authors considered a hypercube view to perceive the label space of MLC problems geometrically and then to capture the correlation between labels. Instead of using label transformations, other works addressed the reduction problem by selecting a small subset of the labels that can approximately span the original label space. For example, Balasubramanian and Lebanon [23] proposed to train only on a small subset of the labels that is selected by solving a group-sparse learning problem. However, the formulated optimization problem by the authors is computationally expensive, especially in scenarios with a large number of labels.

Reducing the label space has also been tackled from a descriptive perspective. Charte *et al.* [24] proposed the mining of association rules [25] to capture the inter-label dependencies, later using such rules to reduce the label space. Once the prediction is done by a multi-label classifier on the reduced space, the previously mined association rules are responsible for inferring the rest of the labels to produce the final prediction. However, the main drawback of this method is related to the possibility of not finding any rule that reduces the label space; in particular, 37.5% of the datasets used in this study could not be reduced. That even enhances more the need to allow the user to reduce the label space a predefined ratio under its own control and risk, as the proposal presented in our study; thus, computationally complex algorithms could be executed over any dataset if desired.

More recently, Kumar *et al.* [10] assume the existence of groups of labels according to their sparsity. Therefore, the label embedding is made for these groups independently, which are obtained by a clustering algorithm. Later, the latent factor matrices are find to approximate the ground truth matrix and thus recover the original set of labels. However, they present their proposal as a multi-label classifier itself, not giving a chance to run any other MLC algorithm of the user's choice on the reduced dataset. It is also noteworthy that in this case, the clusters of labels are also used to reduce the dimensionality of the feature space, but the label space reduction is performed independently from the feature one. Besides, Ji *et al.* [11] proposed a label space reduction method in two phases: first, few uninformative labels are removed following an exact *Boolean Matrix Factorization* (BMF) procedure; and then, a number of informative labels are selected using a genetic algorithm, where the approximation ability of the reconstruction matrix is used as fitness value. In this case, a percentage of the informative labels could be reduced; but such percentage does not include the uninformative labels that can be reduced in the first phase.

Deep neural networks have been also applied to the label space reduction problem. The work of Liu *et al.* [26] proposes an algorithm where both the label and features spaces are reduced following different independent processes. For reducing the label space, they propose to learn a latent label matrix considering the correlation among labels, and later, a deep network is used to map the latent feature space and the label one. However, this process is proposed specifically for the text classification scenario, which share specific and common properties. Finally, Chen *et al.* [27] also propose a MLC method where both the feature and label spaces are independently reduced by using deep neural networks frameworks.

Although the aforementioned feature-independent methods tend to be less computationally complex than those that consider the input features to reduce the label space, the main drawback of this type of methods is that, to the best of our knowledge, none of the previously proposed methods fulfill both the following two criteria simultaneously: (I) providing the user a complete control on the ratio of reduced labels, thus ensuring that the number of labels removed is known a priori and not problem-dependent; and (II) enabling the user to run any MLC algorithm on the reduced dataset, thus making it possible to execute more complex methods on scenarios that would be prohibitive unless such reduction were made. Thus, it is important to highlight that any comparison among existing methodologies and our proposal (which meets both criteria and is described in the following section) would be unfair.

III. LABEL SPACE REDUCTION METHODOLOGY

In this section, the proposed methodology to reduce the label space in multi-label problems is presented. The proposed methodology comprises two stages: the first one reduces the label space before any multi-label classifier is executed, whereas the last one reconstructs the label space from the predictions obtained by the multi-label classifier trained on the reduced dataset.

A. LABEL SPACE REDUCTION

The first stage of the proposed methodology is performed before a multi-label classifier Φ is assessed on a multi-label dataset. Let us say $\mathcal{L}$ is a label space composed of q labels $\ell_1, \ell_2, \ldots, \ell_q$, where each label is a binary variable. Then, the proposed reduction process is based on the following hypothesis: *the label ℓ_k could be eliminated if it can be estimated from the other labels $\ell_i \in \mathcal{L} : 1 \leq i \leq q \wedge i \neq k$*. The hypothesis is based on the fact that labels are commonly related in multi-label problems [28] and, therefore, such dependencies can be effectively exploited to reduce the label space. In what follows, a formal description of our approach is portrayed.

Let us say $\mathcal{P}$ is the set of all the possible permutations of the q labels, where a particular permutation of labels (hereafter, label chain) is represented as $\zeta = (\ell_{\pi 1}, \ell_{\pi 2}, \ldots, \ell_{\pi q})$. On the other hand, let ρ_{ℓ_k} be a model that can estimate a distribution for label ℓ_k, denoted as $B(\ell_k | \ \ell_{k+1}, \ldots, \ell_q; \theta)$, given the labels $\ell_{k+1}, \ldots, \ell_q$, where θ is the set of parameters of the model. Furthermore, let us say that ρ_{ℓ_k} minimizes the following empirical risk by averaging the 0-1 loss function on a set of m examples

$$\text{error}(\rho_{\ell_k}) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}(y^i_{\ell_k},\ \rho_{\ell_k}(y^i_{\ell_{k+1}}, \ldots, y^i_{\ell_q})),$$

where $y^i_{\ell_k}$ indicates whether the instance i belongs (1) or not (0) to the label ℓ_k, and $\rho_{\ell_k}(y^i_{\ell_{k+1}}, \ldots, y^i_{\ell_q})$ is the prediction of the label ℓ_k made by the model; the model considers as input the actual values of the labels $\ell_{k+1}, \ldots, \ell_q$ for the instance i. Therefore, the goal of our approach is to find a label chain ζ that minimizes

$$\underset{\zeta \in \mathcal{P}}{\text{argmin}} \prod_{i=1}^{q-1} \text{error}(\rho_{\ell_{\pi i}}), \tag{1}$$

In other words, we want to find a label chain that produce the minimum expected error rate on the m examples, assuming that the label $\ell_{\pi i} : i < q$ can directly be estimated from the subsequent labels $\ell_{\pi j} : i < j \leq q$ of the chain ζ. However, minimizing empirical 0-1 loss is not computationally feasible and, therefore, other measures should be considered. In this regard, Cortes and Mohri demonstrated in [29] that a binary classifier minimizing the error rate also optimizes the area under ROC curve (AUC). Considering that each model $\rho_{\ell_{\pi i}} : 1 \leq i < q$ corresponds to a binary classifier that predicts a label given other labels, then the equation 1 can be transformed in

$$\underset{\zeta \in \mathcal{P}}{\operatorname{argmax}} \prod_{i=1}^{q-1} \mathrm{AUC}(\rho_{\ell_{\pi i}}), \qquad (2)$$

where $\mathrm{AUC}(\rho_{\ell_{\pi i}})$ computes the probability that has the classifier $\rho_{\ell_{\pi i}}$ in predicting correctly the label $\ell_{\pi i}$ for two examples that have been randomly chosen. Another justification in favor of using AUC as the evaluation measure of each binary classifier lies in the fact that if a multi-label problem is decomposed into several binary classification problems, then high imbalanced datasets can commonly be obtained and, therefore, it is recommendable to use an evaluation measure not so sensitive to this issue [30].

The formulated approach is optimistic because it assumes that we can always find a label chain that allows to attain good estimations for all the labels; in other words, it is assumed a priori that the inter-label correlations are strong enough in such way that each label could be predicted from the other ones. However, the joint distribution of the labels is often not known a priori and varies from one problem to another. Consequently, lower average expected values of AUC could be attained as the error rates propagate and increase along the classifier chain $\rho_{\ell_{\pi 1}}, \rho_{\ell_{\pi 2}}, \ldots, \rho_{\ell_{\pi q-1}}$. On the other hand, although the optimization problem presented by equation 1 has been relaxed, encountering the global solution to equation 2 is also computationally expensive in datasets with a large number of labels, since the size of the search space is $q!$. A possible solution to this issue is to find partial label chains instead of complete chains, i.e. label chains that do not include all the q labels but a percentage of the original label where the average error in the estimation is still acceptable. Therefore, in this work, a greedy algorithm (Algorithm 1) that can reach an approximate solution for the formulated optimization problem is proposed, where a partial label chain is generated; i.e. the algorithm can find a chain of a subset of labels.

Algorithm 1 follows an iterative process: (I) determine the label ℓ_k that could be better estimated from the rest of labels $\ell_i \in \mathcal{L} : 1 \leq i \leq q \wedge i \neq k$; (II) delete ℓ_k from the set $\mathcal{L}$; and (III) repeat the first step until the stop condition is reached. To decide whether a label ℓ_k is a candidate to be predicted from others, a binary classifier ρ is trained using a dataset where ℓ_k states for the target and the rest of labels remaining in $\mathcal{L}$ are used as the input variables.

An important aspect of the proposed algorithm is the stop condition. Two possible stop conditions are as follows: (I) to use a threshold value to decide when a label has been well estimated by the others, so labels whose estimation is poorer than the threshold are kept in the output space; and (II) to set the desired percentage of labels to reduce, independently of the quality of such estimations. Even when the first stop condition could be a feasible and great option, the use of a threshold value is problem dependent and therefore it should be necessary to know a priori a specific threshold value, or to tune it. Additionally, small reductions of the label space can be obtained if high thresholds are considered at the beginning of the process and, in the worse case, no reduction would be performed. Thus, in this work, the proposed algorithm considers the second stop condition, allowing users to define the desired level of reduction.

Algorithm 1 Label Space Reduction

```
Input
    $\mathcal{D}$: Multi-label dataset.
    $\rho$: Binary classifier.
    $\gamma$: Percentage of reduction.
Output
    $\mathcal{D}_r$: Reduced multi-label dataset.
    $P$: Stack of binary classifiers.

$\mathcal{F} \leftarrow$ extractFeatures($\mathcal{D}$)
$\mathcal{L} \leftarrow$ extractLabels($\mathcal{D}$)
$n_\ell \leftarrow \lfloor \gamma \times |\mathcal{L}| \rfloor$        ▷ Number of labels to eliminate
$P \leftarrow []$        ▷ Initialize stack
for $i \leftarrow 1$ to $n_\ell$ do
    for each $\ell_k \in \mathcal{L}$ do
        $\mathcal{F}_k \leftarrow \mathcal{L} \setminus \ell_k$        ▷ Consider the rest of labels as features
        $D_k \leftarrow$ construct ($\mathcal{F}_k, \ell_k$)        ▷ Create dataset for $\ell_k$
        $\rho_k \leftarrow$ crossvalidate ($\rho, D_k$)        ▷ Evaluate estimation of $\ell_k$
    end for
    $\ell_b \leftarrow \arg\max_{\ell_k \in \mathcal{L}} \mathrm{AUC}(\rho_k)$        ▷ Best estimated label
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \ell_b$
    push ($P, \rho_b$)        ▷ Push classifier to the stack
end for
$\mathcal{D}_r \leftarrow$ construct ($\mathcal{F}, \mathcal{L}$)
return $\mathcal{D}_r, P$
```

This stop condition allows users to define the desired level of reduction at their own risk; although it should be stressed that significant errors in the estimation of the labels might be obtained if a high level of reduction is applied by the user. Nevertheless, the reduction level that could be attained with the proposed approach, enables to run complex multi-label methods that, in normal conditions, are impractical to execute in scenarios with a large number of labels.

In summary, the algorithm receives as input a multi-label dataset $\mathcal{D}$ in which the label space reduction will be performed, a binary classifier ρ which will be used to estimate a given label from the rest (and therefore, to select which label is removed in each iteration), and the desired percentage of reduction γ. Finally. the algorithm returns the reduced multi-label dataset $\mathcal{D}_r$, and a stack of binary classifiers P that can subsequently be used to reconstruct the label space.

Regarding the computational complexity of the proposed algorithm, the biggest complexity lies in the nested loop where the binary classifiers are constructed and evaluated. In the first iteration, q binary classifiers are constructed, in the second iteration, $q-1$ classifiers, and so, in the last iteration, $q - n_\ell + 1$ classifiers (n_ℓ is the number of labels to delete). A total of $(2q - n_\ell + 1) \times n_\ell / 2$ classifiers are therefore required to be constructed in the whole process, that is a quadratic number since n_ℓ is a fraction of q. Let us say $f_{\mathcal{L}_{\ell_k} \times \ell_k}$ represents the cost function to train a binary classifier considering $\mathcal{L}_{\ell_k}$ as the input variables and ℓ_k as target variable. Therefore, the label reduction process is asymptotically upper bounded by the function $O(q^2 \times f_{\mathcal{L}_{\ell_k} \times \ell_k})$. It should be highlighted that the proposed algorithm might be computationally expensive in multi-label problems with huge label spaces. However, it should be stressed that this process is executed only once before assessing any multi-label classifier. Furthermore, this process can take advantage of distributed and parallel processing models as the binary classifiers constructed in the nested loop are totally independent.

This approach for the reduction of the label space implies that a multi-label classifier Φ can be built considering a smaller label space so, given a test example, the classifier Φ returns a subset of the labels; then, the rest of the labels are inferred (as explained in the next section). The use of binary models to decide whether a label can be deleted or not is an advantage due to the vast state-of-the-art on binary classification, including methods that range from those that detect linear relationships to the ones capable to model complex non-linear functions. Consequently, the performance of the proposed approach greatly depends on the ability of the binary classifier to exploit the dependencies among labels; more complex or accurate binary models would be expected to reduce the error rate in the reduction process, while simpler binary classifiers would increase such error.

B. LABEL SPACE RECONSTRUCTION

The objective of the second stage of the proposed methodology is to the reconstruct the label space that was previously reduced. For this end, it takes the predictions of a multi-label classifier that was trained on the reduced multi-label problem, and estimates the rest of the original labels. Here, suppose that the original label space $\mathcal{L}$ was reduced by eliminating the labels $\ell_{k_1}, \ell_{k_2}, \cdots, \ell_{k_n}$ in this very order. As a result, the first phase returned a chain $\rho_{\ell_{k_1}}, \rho_{\ell_{k_2}}, \cdots, \rho_{\ell_{k_n}}$ of binary classifiers, where the classifier ρ_{ℓ_k} considered the labels $\ell_j \in \mathcal{L} : j \neq k$ as inputs to predict ℓ_k. A reduced dataset $\mathcal{D}_r$, i.e., a multi-label dataset only comprising those labels that have not been removed, is also obtained from the first phase. Thus, given a test set $\mathcal{D}_{ts}$ with the same structure as the original dataset $\mathcal{D}$, and a multi-label classifier Φ, the second stage (see Algorithm 2) of the proposed methodology comprise the following steps: (I) train Φ on the reduced training set $\mathcal{D}_r$; (II) test Φ on $\mathcal{D}_{ts}$ only predicting the labels known by Φ; and (III) use the chain of binary classifiers in P to estimate the rest of the labels in the reverse order that they were removed.

Regarding the computational complexity to reconstruct the label space, let us say f_{tr}^{Φ} is the cost function to train the multi-label classifier Φ on the reduced dataset, f_{ts}^{Φ} the cost function to evaluate Φ on m test samples, and f_P represents the maximum cost to evaluate the stack of classifiers P. The computational complexity of Algorithm 2 is therefore $O(max(f_{tr}^{\Phi} + f_{ts}^{\Phi}, f_P))$. One drawback of this label space reconstruction process is that the chaining procedure cannot be parallelized. However, it can be serialized and, therefore, it would only require a single binary classification problem in memory at a time.

Algorithm 2 Label Space Reconstruction

```
Input
    $\mathcal{D}_{ts}$: Test data.
    $\mathcal{D}_r$: Reduced training data.
    $\Phi$: Multi-label classifier.
    $P$: Stack of binary classifiers.
Output
    $\mathcal{L}_e$: Estimated label space.

$\Phi \leftarrow$ train($\mathcal{D}_r$)                      ▷ Train MLC with the reduced set
$\mathcal{L}_e \leftarrow$ predict ($\Phi$, $\mathcal{D}_{ts}$)       ▷ Predict reduced set of labels in test set
while $P$ is not empty do
    $\rho_k \leftarrow$ pop ($P$)
    $\ell_k \leftarrow$ predict ($\rho_k$, $\mathcal{L}_e$)             ▷ Estimate each reduced label
    $\mathcal{L}_e \leftarrow \mathcal{L}_e \cup \ell_k$
end while
sortLabels($\mathcal{L}_e$, $\mathcal{D}_{ts}$)                  ▷ Put labels in original order
return $\mathcal{L}_e$
```

IV. EXPERIMENTAL STUDY

The objective of the experimental study is to evaluate the impact caused by the proposed methodology in the overall performance of the MLC algorithms. In this section, first the datasets and experimental settings are described. Then, the obtained results are analyzed, dividing the experimental study into three main parts: (I) analysis of the predictive performance obtained by the binary classifiers used to reduce the datasets at different reduction levels; (II) study of the impact of the reduction level in the MLC algorithms' required runtime; and (III) analysis of how the predictive performance of the MLC algorithms varies at different reduction levels. Finally, a discussion of the obtained experimental results is presented.

It should be noted that most results are summarized through boxplots, where the median is represented by a horizontal line inside the box, and the mean as a cross symbol ($\times$). Also note that outliers have not been included in any of the graphics in order to ease the plot interpretation. Due to the great amount of results collected and in order to make the paper more readable, only figures summarizing the obtained results are described in this paper.[1]

A. DATASETS AND SETTINGS

The experimental study has been carried out over a wide set of 25 well-known multi-label datasets. A summary of such datasets is shown in Table 1, denoting the number of examples (m), attributes (d), labels (q), and the average ratio of dependent label pairs ($rDep$) [21]. These datasets have been specifically selected according to the number of labels, which ranges from 19 (simple label spaces) to 233 labels (complex label spaces), as well as the ratio of dependent labels, ranging from values near to zero, meaning that almost all labels are

[1] The supplementary material available at http://www.uco.es/kdis/label-space-reduction/ includes detailed and raw results of the experimental study and the statistical tests. It also includes detailed information about the datasets.

TABLE 1. Multi-label datasets used in the experimental study. The datasets are sorted in ascending order by number of labels (*q*).

Dataset	m	d	q	*rDep*
Birds	645	260	19	0.123
20NG	19,300	1,006	20	0.984
Yahoo Entertainment	12,730	32,000	21	0.367
Slashdot	3,782	1,079	22	0.273
Yahoo Recreation	12,830	30,320	22	0.455
Ohsumed	13,930	1,002	23	0.526
Yahoo Arts	7,484	23,150	26	0.338
Genbase	662	1,186	27	0.157
Yahoo Society	14,510	31,800	27	0.382
Yahoo Business	11,210	21,920	30	0.209
Yahoo Health	9,205	30,610	32	0.192
Yahoo Reference	8,027	39,680	33	0.169
Yahoo Education	12,030	27,530	33	0.199
Yahoo Computers	12,440	34,100	33	0.364
Yahoo Social	12,110	52,350	39	0.189
Yahoo Science	6,428	37,190	40	0.196
Medical	978	1,449	45	0.039
Enron	1,702	1,001	53	0.141
Langlog	1,460	1,004	75	0.035
Reuters-K500	6,000	500	103	0.080
Stackex coffee	225	1,763	123	0.017
Bibtex	7,395	1,836	159	0.111
CAL500	502	68	174	0.192
Stackex chemistry	6,961	540	175	0.056
Stackex philosophy	3,971	842	233	0.040

independent among them, to higher values which denote that a significant amount of label pairs are dependent among them (in such cases, a better modeling of labels would be expected). Besides, it should be noted that great diversity in both the number of input attributes (ranging from 68 to 52,350) and the number of examples (ranging from 225 to 19,300) have been considered. With these 25 datasets we cover a wide range of possible characteristics of the data. Further, and as shown later in Section IV-C, some algorithms in specific datasets did not finish their execution for a low reduction ratio within a month. It leads us to the conclusion that no more complex datasets could be considered, since it would make the experimental study intractable in a reasonable period of time.

In order to construct the binary classifiers that correctly predict the labels, a Support Vector Machine (SVM) has been considered by the proposed methodology, since it has been demonstrated to work well at solving binary classification problems having all the input attributes defined in binary domains [31]. Besides, a linear kernel and various values for the parameter $C = \{10^{-5}, \cdots, 10^{5}\}$ have been taken into account, choosing in each case the one that performed the best. Nevertheless, it is noteworthy that any algorithm able to perform a binary classification could be used instead.

In order to assess how the label space reduction affects to both the runtime complexity and the predictive performance of the MLC algorithms, three well-known MLC algorithms have been considered: BR, LP, and IBLR-ML. The use of BR and LP is motivated by the fact that they are two of the simplest and more representative PTMs, which have been used as basis for many well-known MLC methods [12]. The reduction of the label space implies that BR builds fewer binary classifiers so its runtime complexity is also reduced. For example, both Classifier Chains and its ensemble version [32], which are one of the best-performing methods in MLC, are based on BR; so by reducing the complexity of BR, those methods that are based on it will also have their complexity reduced. In the case of LP, on the other hand, a smaller label space leads to consider a fewer number of combinations of labels, also reducing its runtime and complexity. Therefore, methods based on LP such as Ensemble of Pruned Sets [33] or RA*k*EL [34] will benefit from this reduction. As base classifier for both BR and LP, a C4.5 decision tree was considered with a confidence level of 0.25 and a minimum of 2 objects per node. It was used since it is widely used in the literature for these kind of algorithms [12].

On the other hand, an AAM such as IBLR-ML has been selected, since their computational complexity is usually less dependent on the number of labels. However, IBLR-ML adds the labels of the neighbors as extra input attributes, so then a reduction of the label space would have also an effect on its runtime. The parameters used by IBLR-ML are the same proposed in its original work [16], i.e. a number of neighbors equal to 10, and the Euclidean metric to calculate the distance between points. Finally, as EMLCs combine either PTMs or AAMs indifferently, their complexity is directly related to the one of their members. Therefore, EMLCs are not considered in this study, but their complexity is expected to reduce in the same order as their base members.

Additionally, a 10-folds cross-validation procedure has been followed on each of the reduced datasets. Note that the reduction of the label space is performed as a preprocessing stage and, therefore, it is performed only once per dataset. On the other hand, the reconstruction of the label space must be performed in each fold execution. All the algorithms have been executed over the same folds, and the results have been averaged along all the partitions. To estimate the impact of the label space reduction on the overall performance of the MLC algorithms, the average runtime to finish the execution of each fold has been collected. It includes the time required for training, testing and the label space reconstruction. As for the predictive performance, four evaluation measures which assess different aspects and have been widely used in MLC have been considered: Hamming loss (HL), Example-based FMeasure (ExF), Micro-averaged FMeasure (MiF), and Macro-averaged FMeasure (MaF) [1].

HL is one of the most widely used evaluation measures in MLC, which computes the average number of times that a label is incorrectly predicted. It considers both prediction errors (a negative label is predicted as positive), and omission errors (positive labels are not predicted as positive). It is calculated as in (3), where m is the number of test examples, and Δ represents the symmetric difference between the true ($\mathbf{Y}_i$) and predicted ($\hat{\mathbf{Y}}_i$) set of labels of the example i. Hereafter, the symbols ↓ and ↑ indicate that the measure is minimal and maximal, respectively.

$$\downarrow \text{HL} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{q} \times |\mathbf{Y}_i \Delta \hat{\mathbf{Y}}_i| \tag{3}$$

On the other hand, FMeasure is a widely used evaluation measure in traditional classification, and for MLC it can be calculated from different points of view. ExF calculates the measure for each instance, and then averages their value, as in (4). On the other hand, MiF, in (5), and MaF, in (6), calculate the measure based on the contingency matrix of each label. While the former gives more weight to more frequent labels in the calculation, the latter gives the same weight to all labels.

$$\uparrow \text{ExF} = \frac{1}{m}\sum_{i=1}^{m}\frac{2 \times \left|\mathbf{Y}_i \cap \hat{\mathbf{Y}}_i\right|}{|\mathbf{Y}_i| + \left|\hat{\mathbf{Y}}_i\right|} \tag{4}$$

$$\uparrow \text{MiF} = \frac{\sum_{\ell=1}^{q} 2 \times \text{tp}_\ell}{\sum_{\ell=1}^{q} 2 \times \text{tp}_\ell + \sum_{\ell=1}^{q} \text{fn}_\ell + \sum_{\ell=1}^{q} \text{fp}_\ell} \tag{5}$$

$$\uparrow \text{MaF} = \frac{1}{q}\sum_{\ell=1}^{q}\frac{2 \times \text{tp}_\ell}{2 \times \text{tp}_\ell + \text{fn}_\ell + \text{fp}_\ell} \tag{6}$$

where tp_ℓ, fp_ℓ, and fn_ℓ are the number of true positives, false positives and false negatives for the ℓth label, respectively.

Finally, a hypothesis testing by means of non-parametric statistical tests has been conducted with the aim of determining whether there exist significant differences in the overall performance of the algorithms at different reduction levels of the label spaces. The Friedman's test has been used to analyze the general differences, whereas the Shaffer's post-hoc test has been employed to perform all pairwise comparisons [35]. Finally, it should be highlighted that all the experiments have been performed on a machine with 6 Intel Xeon E5645 CPUs at 2.40 GHz and 24 GB of RAM.

B. PREDICTIVE PERFORMANCE OF BINARY CLASSIFIERS IN THE REDUCTION STAGE

The predictive performance of the MLC algorithms is directly related to the capacity of the binary classifiers to accurately reconstruct the label space. Hence, the performance levels attained by these binary classifiers are of paramount importance. This performance, in addition, is expected to be worse at high levels of reductions since those labels that are best predicted are removed at early stages.

In order to analyze this predictive performance, the AUC values used to decide which label is eliminated in each iteration of the reduction process were collected and averaged. Figure 1 presents the average AUC values when reducing up to 10% the number of labels; when reducing from 10% to 20%; and so on. The results demonstrate that the average and median of the AUC values decrease in almost a lineal way while the reduction level increases. Focusing on a reduction up to 10% of the labels, it is obtained that the AUC values for the binary classifiers are greater than 0.9 in most of the cases. Such high AUC values imply that the label space can be accurately reconstructed. When much higher reduction levels are considered, e.g. those greater than 40%, the resulting AUC values are significantly deteriorated, and in some cases, values near to 0.5 are obtained (meaning almost a random prediction). At this point, and given that the predictive performance tends to be reduced with the increase of the reduction level, there is no sense to go further than 50% of reduction.

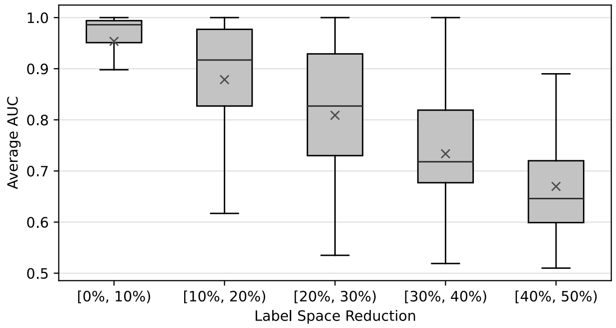


FIGURE 1. Average AUC values of the binary classifiers at the different reduction levels.

Finally, as previously commented in Section III, the stopping criterion is one of the most important aspects of the proposed methodology since it may affect the overall performance of the MLC algorithm. It is therefore important to remark here that the percentage of reduction is responsibility of the expert, who should decide and assume the risk of losing a significant grade of precision at high reduction levels of the label space, while allowing to execute complex algorithms over complex datasets.

C. ANALYSIS OF THE RUNTIME REQUIRED BY MLC ALGORITHMS

In this second analysis, the aim is to study the variation in runtime of the MLC algorithms when the label space is reduced. For this purpose, five reduction levels have been considered (10%, 20%, $\cdots$, 50%) to calculate the variation of the runtime for the MLC algorithms when they are applied on the original and the reduced multi-label dataset. It is calculated as $(\text{time}_\text{r} - \text{time}_o)/\text{time}_o$, being time_o the required runtime of the algorithm when the original dataset is used, and time_r the runtime using the subsequent reduced dataset. As a reduction in the runtime is expected, the rate of change will be negative for each reduction level (from 10% to 50%).

Results are summarized in Figure 2, illustrating the aforementioned rate of change in the runtime; it has been averaged across all the datasets, for each algorithm at different reduction levels. As illustrated, the runtime required by BR is reduced in a linear way for the different reduction levels. This linear reduction is completely awaited since the number of binary classifiers generated by BR is dependent on the number of labels in data. It should be noted that BR was the algorithm that reduced most the runtime in all cases.

It can be observed that, from the three algorithms, both BR and LP (as PTMs) are those that best improve their runtime

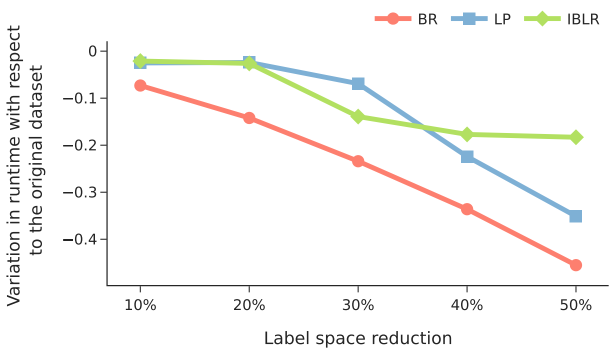


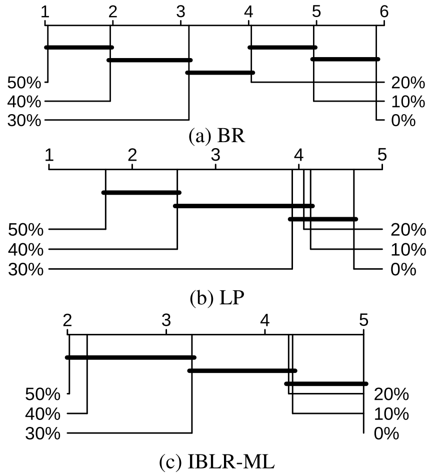
FIGURE 2. Rate of change in runtime at five reduction levels.

when the label reduction rate increases. In the case of LP, such runtime reduction rate is not so high until 30% of the labels are reduced; after that, the runtime starts to improve drastically. Such behavior is explained given that, at low reduction levels, the total number of possible label combinations in the data may vary very little; hence, the complexity of the multi-class problem that LP handles is maintained. On the other hand, at this point it is noteworthy that LP is such a complex algorithm that, under the available computational resources and after a month, it was not possible to finish its execution on three datasets: *Yahoo Business* and *Yahoo Computers* could execute only after 50% label reduction was made, while *Yahoo Education* finished for 40% and 50% reduction levels. Thus, it is corroborated that there are MLC algorithms that are impossible to execute in a reasonable quantum of time without considering, a priori, a significant reduction of the label space.

Finally, analyzing the behavior of IBLR-ML (see Figure 2), it is demonstrated that the number of labels do not affect so much in the runtime. The final reduction rate is almost half of the one obtained either by LP or BR. At a reduction level of 50%, IBLR-ML reduces its runtime in a rate of only $\sim$ 20%. In any case, our approach leads to a reduction, drastic in some cases, in the required runtime of the MLC algorithms.

The final step of this analysis of the runtime required by the MLC algorithms is to perform a statistical comparison to determine significant differences at different reduction levels. To conduct the statistical analysis, the raw runtime values were used instead of the rate of changes. In such cases where the MLC algorithm is not able to finish after a month, then the worst ranking value was assigned. The Friedman's test detected that there were general statistical differences in the three MLC algorithms at a significance level of $\alpha = 0.01$, rejecting the null hypothesis with a p-value smaller than 2.20E−16 for BR, p-value of 7.97E−09 for LP, and p-value of 4.35E−10 for IBLR-ML.

Then, the Shaffer's post-hoc test was performed to detect where these significant differences were located. The results for this post-hoc test, at a significance level of $\alpha = 0.01$, are summarized in Figure 3. This figure illustrates that BR at low reduction levels (20%) significantly reduces the runtime



(a) BR

(b) LP

(c) IBLR-ML

FIGURE 3. The results of the Shaffer's test after performing all pairwise comparisons between the required runtime at the different reduction levels.

required on the original dataset (0% of reduction). It can be observed that the reduction of the label space has a major impact in BR, identifying many significant differences in its runtime for the different reduction levels. As for the case of LP, the statistical difference in runtime versus using the whole label set, was obtained when the the reduction level is, at least, 40% of the label space. This result is caused by the complexity of LP, as it was previously explained. Finally, as for IBLR-ML, at least a reduction of 30% of the label space is required to significantly reduce the runtime.

So far, the results demonstrated that it is possible to significantly decrease the required runtime of MLC algorithms by reducing effectively the label space of large-scale datasets, supporting the first part of the aim and hypothesis of this work.

D. ANALYSIS OF THE PREDICTIVE PERFORMANCE OF MLC ALGORITHMS

In this third analysis, and once it has been demonstrated that the required runtime of the MLC algorithms is significantly improved by reducing the label space, we study the behavior of the predictive performance for each of the MLC algorithms when different reductions of the label space are considered.

Similarly to the previous study of the runtime, the variation rate for different evaluation measures (HL, ExF, MiF and MaF) is analyzed. HL is a minimal measure so its variation with respect to the original scenario is calculated as $(\mathrm{HL}_o - \mathrm{HL}_r)/\mathrm{HL}_o$, where HL_o states for the value of HL when the original dataset was used, and HL_r the HL value obtained in the reduced dataset. Here, negative ratios state for a drop in the predictive performance. On the other hand, FMeasure is a maximal measure and its variation with respect to the use of the original dataset is calculated as $(\mathrm{FMeasure}_r - \mathrm{FMeasure}_o)/\mathrm{FMeasure}_o$. Here, $\mathrm{FMeasure}_o$ is the performance of the algorithm on the original dataset, whereas $\mathrm{FMeasure}_\mathrm{r}$ is the performance obtained on the

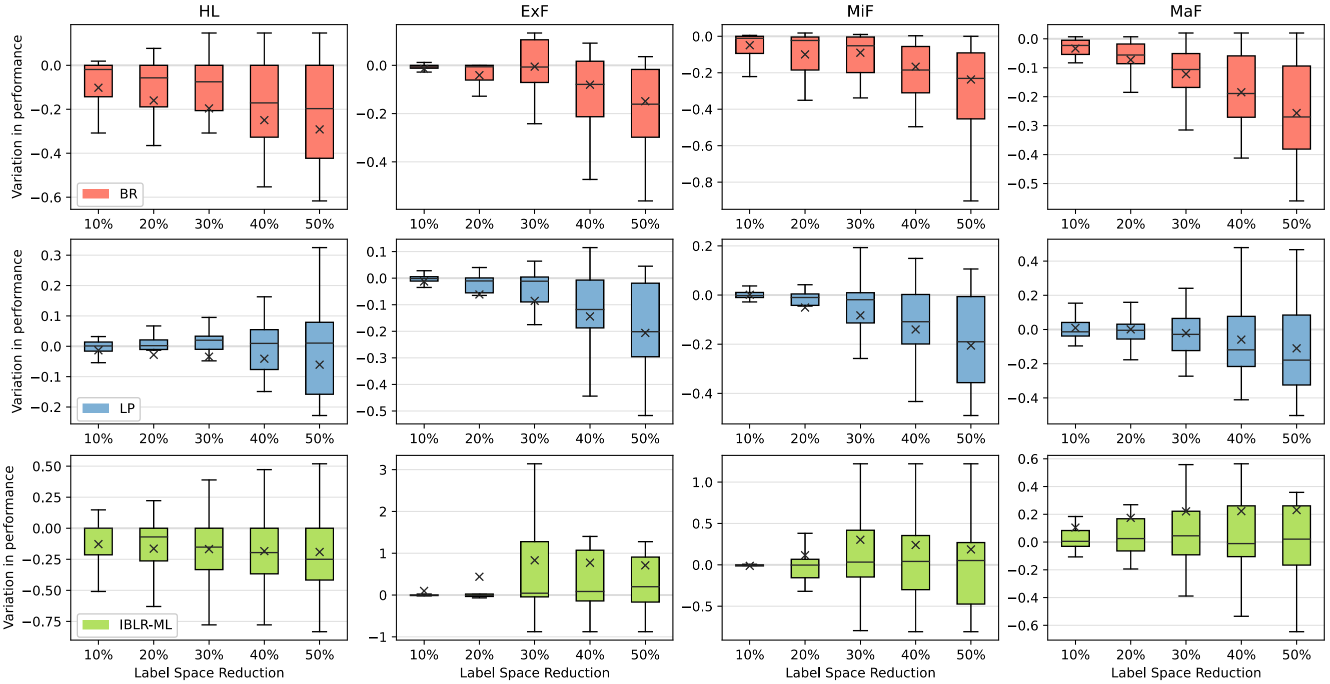


FIGURE 4. **Variation in the predictive performance of the three different MLC algorithms (BR, LP, and IBLR-ML) when the label space is reduced.**

corresponding reduced dataset; note that this equation is valid for any of the three ways to calculate FMeasure in multi-label scenarios (ExF, MiF, and MaF). As in HL, negative ratios still represent a loss of predictive performance.

In Figure 4, the variation in performance of the different MLC algorithms when the label space is reduced at different levels is presented. Analyzing the results for BR (upper row of Figure 4), it is obtained that it tends to reduce its predictive performance with the reduction of the number of labels in the output space, achieving a high reduction of performance when the number of labels is reduced at least a 40%. Note that the median ratio values of performance drop are always smaller than 0.10 at a reduction level of 30%, and even only of 0.01 for the ExF measure. In some specific cases, e.g. HL and MaF, a better predictive performance is obtained when the label space is further reduced. This interesting behavior is obtained by analyzing the upper limits of the boxplots.

The variation in the predictive performance is also analyzed for the LP algorithm (see middle row of Figure 4). Similarly to the previous analysis on BR, it is obtained that in most of the cases, the predictive performance tends to decrease with the reduction of the number of labels. When analyzing HL, on the contrary, the behavior is completely different to the rest of evaluation measures; in all cases the median variation is positive, meaning than at least in half of the cases, the predictive performance of LP was even better after the label reduction and reconstruction process. Focusing on the median value in all measures, the rate is never further from -0.03 when the number of labels is reduced up to a 30%, maintaining its performance at considerable reduction levels. When analyzing a reduction of 50% of the labels, it is obtained that the variation rate of the median of any of the evaluation measures is not higher than -0.20. It should be also noted that, analyzing the upper limits of the boxplots in all the evaluation measures, there exists some additional cases where a better predictive performance is obtained when reducing the number of labels (e.g. HL and MaF). Given that the complexity of LP would be much lower when the number of labels decreases, its output (over the reduced dataset) would also be much more precise, and thus the final performance is too. This, explains the behavior of the cases where LP performs better on the reduced dataset.

As for the IBLR-ML algorithm (see bottom row of Figure 4), results denote that the mean values for the HL measure are almost the same along the reduction levels. For the other three measures, however, a suprising behavior is obtained by IBLR-ML, achieving an improvement in the predictive performance when the percentage of reduction is increased. ExF and MiF evaluation measures obtained the best results when the dataset is reduced a 30% and, for 40% and 50% of reduction, these measures remain very similar in mean. The most interesting case is, perhaps, the MaF measure, which average value is increased in each of the reduction levels. Finally, it is also relevant to note the incredible performance of IBLR-ML in the ExF measure, where the average variation rate is about $+0.80$ when the number of labels are reduced, at least, a 30%.

The final step of the analysis of the predictive performance achieved by the MLC algorithms is to perform a statistical

TABLE 2. Level of reduction where the performance of the MLC algorithms is significantly different from the original dataset, according to Shaffer's test with $\alpha = 0.01$.

	HL	ExF	MiF	MaF
BR	50%	40%	30%	30%
LP	50%	-	-	50%
IBLR-ML	-	-	-	-

comparison to determine significant differences at different reduction levels. In this study, the Friedman's test determined that BR presented significant differences in the four evaluation measures at a significance level of $\alpha = 0.01$, rejecting the null hypothesis with p-values 1.24E−04 for HL, 4.23E−07 for ExF, 2.01E−13 for MiF, and 5.55E−16 for MaF. For LP, significant differences were only found in two evaluation measures, that is, ExF and MiF (rejecting the null hypothesis with p-values 9.91E−05 and 3.66E−04 respectively); whereas for the rest of measures, there is no statistical difference in the predictive performance obtained on the original dataset and the reduced dataset. As for IBLR-ML, the Friedman's test determined that no significant difference can be found in any performance measure, stating that the predictive performance of IBLR-ML is not harmed even when the number of labels is reduced at a high pace.

To detect particular significant differences in those cases where the Friedman's test detected statistical differences, the Shaffer's test was conducted. A summary of the results for the Shaffer's test, at a significance level of $\alpha = 0.01$, is illustrated in Table 2. In this table, the level of reduction for which significant differences were found with respect to the use of original dataset, is presented for the different MLC algorithms and evaluation measures. In the case of BR, it is obtained that the overall predictive performance is statistically the same for the original dataset and the reduced dataset up to 20%. As for LP, the overall performance does not present statistical differences with regard to the original dataset for up to 40% of reduction. In the case of IBLR-ML, there is no statistical difference in the overall predictive performance for any reduction level (up to 50% was analyzed).

To sum up, both LP and IBLR-ML maintained their overall predictive performance even when the label space was drastically reduced. By contrast, the predictive performance of BR decreases with the reduction of the label space; however, it is also noteworthy that, according to the previous analysis, BR is the one that most improves its runtime when the number of labels decreases.

E. DISCUSSION

The goal of this experimental study was to analyze the behavior of three MLC algorithms when the label space is reduced by means of the proposed methodology. First, it should be noted that this methodology enables the reduction level to be specified by the expert, considering that level according to the features of the problem at hand, or to the requirements of the particular situation. Besides, any MLC algorithm may be used. Before performing the experimental study, it was expected that high reduction levels would lead to a hard reconstruction of the label space and, therefore, that the overall predictive performance of the MLC algorithms would be harmed.

The experimental results showed that the required runtime of the MLC algorithms can be significantly improved by previously reducing the label space. For those MLC algorithms that work similarly to BR, a significant reduction in the runtime is achieved by only reducing $\sim$ 20% of the labels. However, for algorithms that works with PTMs like LP, or for AAMs algorithms like IBLR-ML, a higher reduction level is required to be applied if it is desired to obtain statistical differences in runtime. Additionally, it is relevant to note that there were various cases where the algorithm only finished its execution in a reasonable quantum of time if it works on a highly reduced dataset. It demonstrates the need of reducing the output space in large-scale multi-label problems.

As it is experimentally demonstrated, the runtime of PTMs is highly affected by the complexity of the label space. Regarding BR, if c labels are removed in the reduction process, its complexity is linearly reduced due to the need to construct only $q - c$ binary classifiers. As for the LP-based algorithms, their complexities are not directly related to the number of labels but to the number of distinct combinations of labels. The number of classes produced by the LP-based methods is upper bounded by 2^q and, therefore, deleting c labels would imply a reduction of 2^{q-c} classes and an improvement of the complexity of the multi-class problem generated by this type of methods. Focusing on IBLR-ML, not only the output space is reduced but also the input space, since it extends the input features with the labels of the k-nearest neighbors of each training instance.

Regarding the predictive performance achieved by the MLC algorithms, the obtained results demonstrated that, in some cases, it is possible to reduce the label space without significantly dropping their predictive performance. For example, the predictive performance achieved by BR is statistically the same when reducing the number of labels up to 20%. On the contrary, for the other two types of MLC algorithms (LP and IBLR-ML), significant reduction in the overall predictive performance was not observed up to a 40% of reduction. More specifically, IBLR-ML may reduce the label space in a 50% with no statistical differences in the overall predictive performance, which is considered an excellent result. Finally, it should be denoted that there were interesting cases where the predictive performance of a specific algorithm, e.g. IBLR-ML, was even improved when the percentage of labels to be removed increases. The explanation for this improvement is related to the expansion of the input space with labels. IBLR-ML reinforces the learning based on those labels that are afterwards used to reconstruct the label space, so the better these labels are predicted, the better is the reconstruction of such space.

V. CONCLUSION

In this paper, a novel methodology for the reduction of the label space in MLC problems have been proposed. This methodology first comprises the label space by building binary classifiers that learn some of the labels from the rest, before any MLC algorithm is executed. Later, the final prediction is obtained by reconstructing the label space from the predictions of the multi-label classifier that was built on the reduced dataset, by means of the binary classifiers obtained in the first stage. One of the major features of this methodology is that the user can reduce the label space as much as required, defining a ratio of labels to reduce. Nevertheless, higher reductions of the label space may lead to a significant loss of predictive performance.

The main aim of our proposal was to significantly reduce the required runtime of the MLC algorithms but without causing a significant reduction in the predictive performance. An extensive experimental study over 25 multi-label datasets with different label space complexities and using three different MLC algorithms demonstrated that it is possible to decrease the required runtime of the MLC algorithms but without significantly reducing their predictive performance. Besides, there are specific cases where the performance of the MLC algorithms increases as the label space is further reduced. Also, there are cases where, without reducing the label space, the algorithm could not be executed under the available computational resources in a reasonable quantum of time, so it reinforced the need of reducing the label space in some problems.

Finally, due to many well-known MLC algorithms were based on BR or LP, it would be also possible to reduce their runtime by means of the proposed methodology.

REFERENCES

[1] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 52:1–52:38, 2015.

[2] X. Zhang, J. Xu, C. Soh, and L. Chen, "LA-HCN: Label-based attention for hierarchical multi-label text classification neural network," *Exp. Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115922.

[3] S. Wen, W. Liu, Y. Yang, P. Zhou, Z. Guo, Z. Yan, Y. Chen, and T. Huang, "Multilabel image classification via feature/label co-projection," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 11, pp. 7250–7259, Nov. 2020.

[4] L. Zhou, X. Zheng, D. Yang, Y. Wang, X. Bai, and X. Ye, "Application of multi-label classification models for the diagnosis of diabetic complications," *BMC Med. Informat. Decis. Making*, vol. 21, no. 1, pp. 1–10, Dec. 2021.

[5] S. Khandagale, H. Xiao, and R. Babbar, "Bonsai: Diverse and shallow trees for extreme multi-label classification," *Mach. Learn.*, vol. 109, no. 11, pp. 2099–2119, Nov. 2020.

[6] E. Gibaja and S. Ventura, "Multi-label learning: A review of the state of the art and ongoing research," *Wiley Interdisci-Plinary Rev. Data Mining Knowl. Discovery*, vol. 4, no. 6, pp. 411–444, 2014.

[7] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label-specific features and class-dependent labels for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3309–3323, Dec. 2016.

[8] J. Huang, P. Zhang, H. Zhang, G. Li, and H. Rui, "Multi-label learning via feature and label space dimension reduction," *IEEE Access*, vol. 8, pp. 20289–20303, 2020.

[9] X. Wang, J. Li, and J. Xu, "A label embedding method for multi-label classification via exploiting local label correlations," in *Proc. Int. Conf. Neural Inf. Process.*, Sydney, NSW, Australia. Berlin, Germany: Springer, 2019, pp. 168–180.

[10] V. Kumar, A. K. Pujari, V. Padmanabhan, and V. R. Kagita, "Group preserving label embedding for multi-label classification," *Pattern Recognit.*, vol. 90, pp. 23–34, Jun. 2019.

[11] T. Ji, J. Li, and J. Xu, "A globally optimal label selection method via genetic algorithm for multi-label classification," in *Proc. Int. Conf. Database Exp. Syst. Appl.*, Virtual Event. Berlin, Germany: Springer, 2021, pp. 239–247.

[12] J. M. Moyano, E. L. Gibaja, K. J. Cios, and S. Ventura, "Review of ensembles of multi-label classifiers: Models, experimental study and prospects," *Inf. Fusion*, vol. 44, pp. 33–45, Nov. 2018.

[13] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains: A review and perspectives," *J. Artif. Intell. Res.*, vol. 70, pp. 683–718, Feb. 2021.

[14] R. Wang, S. Kwong, X. Wang, and Y. Jia, "Active K-labelsets ensemble for multi-label classification," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107583.

[15] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.

[16] W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Mach. Learn.*, vol. 76, nos. 2–3, pp. 211–225, 2009.

[17] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Jan. 2018.

[18] S. Kashef, H. Nezamabadi-Pour, and B. Nikpour, "Multilabel feature selection: A comprehensive review and guiding experiments," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 2, p. e1240, Mar. 2018.

[19] J. D. Hsu, M. S. Kakade, J. Langford, and T. Zhang, "Multi-label prediction via compressed sensing," 2009, *arXiv:0902.1284*.

[20] D. E. K. Mansouri and K. Benabdeslem, "Towards multi-label feature selection by instance and label selections," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, Virtual Event. Berlin, Germany: Springer, 2021, pp. 233–244.

[21] M. J. Moyano, L. E. Gibaja, and S. Ventura, "MLDA: A tool for analyzing multi-label datasets," *Knowl.-Based Syst.*, vol. 121, pp. 1–3, Apr. 2017.

[22] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Comput.*, vol. 24, no. 9, pp. 2508–2542, 2012.

[23] K. Balasubramanian and G. Lebanon, "The landmark selection method for multiple output prediction," 2012, *arXiv:1206.6479*.

[24] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "LI-MLC: A label inference methodology for addressing high dimensionality in the label space for multilabel classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1842–1854, Oct. 2014.

[25] S. Ventura and J. M. Luna, *Pattern Mining With Evolutionary Algorithms*. Cham, Switzerland: Springer, 2016.

[26] H. Liu, G. Chen, P. Li, P. Zhao, and X. Wu, "Multi-label text classification via joint learning from label embedding and label correlation," *Neurocomputing*, vol. 460, pp. 385–398, Oct. 2021.

[27] C. Chen, H. Wang, W. Liu, X. Zhao, T. Hu, and G. Chen, "Two-stage label embedding via neural factorization machine for multi-label classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3304–3311.

[28] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence and loss minimization in multi-label classification," *Mach. Learn.*, vol. 88, no. 1, pp. 5–45, Jul. 2012.

[29] C. Cortes and M. Mohri, "AUC optimization vs. Error rate minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 313–320.

[30] F. Charte, A. J. Rivera, M. J. del Jesús, and F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, pp. 3–16, Sep. 2015.

[31] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Eur. Conf. Mach. Learn. (ECML)*, 1998, pp. 137–142.

[32] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, p. 333, Dec. 2011.

[33] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 995–1000.

[34] G. Tsoumakas, I. Katakis, and L. Vlahavas, "Random K-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[35] S. Garcia and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, no. 12, pp. 1–18, 2008.

JOSE M. MOYANO received the B.Sc. and M.Sc. degrees in computer science from the University of Córdoba, Córdoba, Spain, in 2014 and 2016, respectively, and the Ph.D. degree in computer science from the University of Córdoba and Virginia Commonwealth University, Richmond, VA, USA, in 2020. He is currently an Assistant Professor with the University of Córdoba and a member of the Knowledge Discovery and Intelligent Systems Research Group. Up to date, he has published 16 articles in indexed journals and international scientific conferences. He has also participated in five national and regional research projects. His research interest includes ensemble methods for multilabel classification.

JOSE M. LUNA received the Ph.D. degree in computer science from the University of Granada, Spain, in 2014. He is currently an Associate Professor with the Department of Computer Science and Numerical Analysis, University of Córdoba, Spain. He is the author of two books related to pattern mining, published by Springer. He has published more than 30 articles in top ranked journals and international scientific conferences. He is the author of two book chapters. He has been also involved in four national and regional research projects. He has contributed to three international projects. His research interests include evolutionary computation and pattern mining.

SEBASTIAN VENTURA (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in sciences from the University of Córdoba, Spain, in 1989 and 1996, respectively. He is currently a Full Professor with the Department of Computer Science and Numerical Analysis, University of Córdoba, where he is also the Head of the Knowledge Discovery and Intelligent Systems Research Laboratory. He has published more than 220 papers in journals and scientific conferences. He has edited three books and several special issues in international journals. He has been also engaged in 16 research projects (being the coordinator of eight of them) supported by the Spanish and Andalusian Governments and the European Union. His main research interests include soft-computing, machine learning, data mining, and their applications. He is a Senior Member of the IEEE Computer Society, the IEEE Computational Intelligence Society, and the IEEE Systems, Man, and Cybernetics Society, as well as the Association for Computing Machinery (ACM).