UNIVERSIDAD DE CÓRDOBA

Escuela Internacional de Doctorado

Nuevos Modelos de Aprendizaje Híbrido para Clasificación y Ordenamiento Multi-Etiqueta

Memoria de Tesis presentada por

Oscar Gabriel Reyes Pupo

como requisito para optar al grado

de Doctor en Informática

Director

Dr. Sebastián Ventura Soto

Córdoba Octubre de 2016

UNIVERSITY OF CÓRDOBA

International Doctoral School

New Hybrid Learning Models for Multi-label Classification and Label Ranking

A Thesis presented by

Oscar Gabriel Reyes Pupo

as a requirement to aim for the degree of

Ph.D. in Computer Science

Advisor

Dr. Sebastián Ventura Soto

Córdoba October, 2016

TÍTULO DE LA TESIS: Nuevos Modelos de Aprendizaje Híbrido para Clasificación y Ordenamiento Multi-Etiqueta.

DOCTORANDO/A: Oscar Gabriel Reyes Pupo

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

En su tesis, D. Oscar Gabriel Reyes Pupo ha abordado tres temas en el contexto del aprendizaje multi-etiqueta: la estimación de atributos, el aprendizaje basado en instancias y el aprendizaje activo.

En el primer tema, se propusieron un total de cinco métodos de estimación de atributos, dos de ellos basados en la aplicación de algoritmos evolutivos. En el segundo tema, se diseñó un nuevo algoritmo de vecindad inspirado en los principios de clasificación basada en gravitación de datos. En el tercer tema, se desarrollaron dos estrategias de aprendizaje activo, y se construyó una librería de clases que favorece la implementación de métodos de aprendizaje activo y la experimentación en esta área de estudio. Además, se propusieron dos aproximaciones que permiten evaluar de una manera más adecuada el rendimiento de las técnicas de aprendizaje activo.

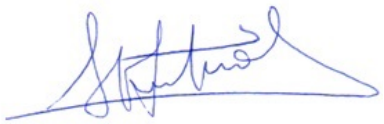
A partir de los resultados alcanzados en esta tesis, se lograron varias publicaciones en revistas internacionales de impacto y conferencias internacionales, lo que muestra la calidad científica del trabajo realizado. Por otra parte, las líneas de investigación desarrolladas en esta memoria no están aún agotadas, existiendo algunas líneas de trabajo futuro que considero pueden también dar lugar a varias publicaciones científicas de calidad.

En conclusión, considero que la memoria presentada por D. Oscar Gabriel Reyes Pupo reúne, en mi opinión, las condiciones necesarias para su defensa.

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 18 de octubre de 2016

Firma del director



Fdo.:_____

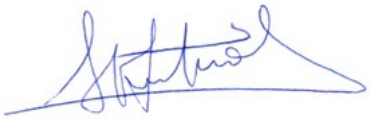
La memoria de Tesis Doctoral titulada "*Nuevos Modelos de Aprendizaje Híbrido para Clasificación y Ordenamiento Multi-Etiqueta*", que presenta Oscar Gabriel Reyes Pupo para optar al grado de Doctor, ha sido realizada dentro del Programa Oficial de Doctorado "Computación Avanzada, Energía y Plasmas" de la Universidad de Córdoba, España, bajo la dirección del Dr. Sebastián Ventura Soto, cumpliendo, en su opinión, los requisitos exigidos a este tipo de trabajos.

Córdoba, Octubre de 2016

El Doctorando

Fdo: Oscar Gabriel Reyes Pupo

El Director



Fdo: Dr. Sebastián Ventura Soto

Tesis Doctoral parcialmente subvencionada por el Ministerio de Economía y Competitividad, proyecto **TIN2014-55252-P**.

Agradecimientos

Esta tesis de doctorado no es el resultado del esfuerzo de una sola persona, sino de un conjunto de personas que de una manera u otra han contribuido a la realización de la misma.

Ante todo quiero agradecerle a Dios por la ayuda y fuerza que me ha dado para llevar a cabo esta empresa. A mis amados padres que desde los comienzos de mis estudios me han apoyado incondicionalmente y me han exhortado a seguir superándome. A mi amada esposa por su amor, consejos, aliento y comprensión. Ciertamente, sin el apoyo y sacrificio de mi familia esta tesis no se hubiera podido realizar. Le doy gracias a mi familia por su apoyo, y a la vez le pido perdón por el tiempo robado para realizar esta tesis, tiempo robado que nunca volverá, en especial los momentos que he dejado de ver crecer a mi pequeña y amada hija.

Quiero agradecerle especialmente a mi director Sebastián Ventura, por haber confiado en aquel muchacho desconocido que le solicitó un día que le dirigiera su tesis, y por el apoyo incondicional que me ha prestado a lo largo de todo este tiempo. Le agradezco al Dr. Carlos Morell de la Universidad de Las Villas, Cuba, por sus valiosos comentarios y colaboración en los trabajos realizados.

También quiero agradecerle a mis amigos de la Universidad de Holguín, Cuba, por el tiempo y momentos que compartimos juntos. A los colegas del grupo KDIS de la Universidad de Córdoba, España, por su apoyo en mis estancias realizadas.

Gracias de todo corazón.

Resumen

En la última década, el aprendizaje multi-etiqueta se ha convertido en una importante área de investigación, debido en gran parte al creciente número de problemas reales que contienen datos multi-etiqueta. En esta tesis se estudiaron dos problemas sobre datos multi-etiqueta, la mejora del rendimiento de los algoritmos en datos multi-etiqueta complejos y la mejora del rendimiento de los algoritmos a partir de datos no etiquetados.

El primer problema fue tratado mediante métodos de estimación de atributos. Se evaluó la efectividad de los métodos de estimación de atributos propuestos en la mejora del rendimiento de los algoritmos de vecindad, mediante la parametrización de las funciones de distancias empleadas para recuperar los ejemplos más cercanos. Además, se demostró la efectividad de los métodos de estimación en la tarea de selección de atributos. Por otra parte, se desarrolló un algoritmo de vecindad inspirado en el enfoque de clasificación basada en gravitación de datos. Este algoritmo garantiza un balance adecuado entre eficiencia y efectividad en su solución ante datos multi-etiqueta complejos.

El segundo problema fue resuelto mediante técnicas de aprendizaje activo, lo cual permite reducir los costos del etiquetado de datos y del entrenamiento de un mejor modelo. Se propusieron dos estrategias de aprendizaje activo. La primer estrategia resuelve el problema de aprendizaje activo multi-etiqueta de una manera efectiva y eficiente, para ello se combinaron dos medidas que representan la utilidad de un ejemplo no etiquetado. La segunda estrategia propuesta se enfocó en la resolución del problema de aprendizaje activo multi-etiqueta en modo de lotes, para ello se formuló un problema multi-objetivo donde se optimizan tres medidas, y el problema de optimización planteado se resolvió mediante un algoritmo evolutivo.

Como resultados complementarios derivados de esta tesis, se desarrolló una herramienta computacional que favorece la implementación de métodos de aprendizaje activo y la experimentación en esta área de estudio. Además, se propusieron dos aproximaciones que permiten evaluar el rendimiento de las técnicas de aprendizaje activo de una manera más adecuada y robusta que la empleada comúnmente en la literatura.

Todos los métodos propuestos en esta tesis han sido evaluados en un marco experimental adecuado, se utilizaron numerosos conjuntos de datos y se compararon los rendimientos de los algoritmos frente a otros métodos del estado del arte. Los resultados obtenidos, los cuales fueron verificados mediante la aplicación de test estadísticos no parámetricos, demuestran la efectividad de los métodos propuestos y de esta manera comprueban las hipótesis planteadas en esta tesis.

Abstract

In the last decade, multi-label learning has become an important area of research due to the large number of real-world problems that contain multi-label data. This doctoral thesis is focused on the multi-label learning paradigm. Two problems were studied, firstly, improving the performance of the algorithms on complex multi-label data, and secondly, improving the performance through unlabeled data.

The first problem was solved by means of feature estimation methods. The effectiveness of the feature estimation methods proposed was evaluated by improving the performance of multi-label lazy algorithms. The parametrization of the distance functions with a weight vector allowed to recover examples with relevant label sets for classification. It was also demonstrated the effectiveness of the feature estimation methods in the feature selection task. On the other hand, a lazy algorithm based on a data gravitation model was proposed. This lazy algorithm has a good trade-off between effectiveness and efficiency in the resolution of the multi-label lazy learning.

The second problem was solved by means of active learning techniques. The active learning methods allowed to reduce the costs of the data labeling process and training an accurate model. Two active learning strategies were proposed. The first strategy effectively solves the multi-label active learning problem. In this strategy, two measures that represent the utility of an unlabeled example were defined and combined. On the other hand, the second active learning strategy proposed resolves the batch-mode active learning problem, where the aim is to select a batch of unlabeled examples that are informative and the information redundancy is minimal. The batch-mode active learning was formulated as a multi-objective problem, where three measures were optimized. The multi-objective problem was solved through an evolutionary algorithm.

This thesis also derived in the creation of a computational framework to develop any active learning method and to favor the experimentation process in the active learning area. On the other hand, a methodology based on non-parametric tests that allows a more adequate evaluation of active learning performance was proposed.

All methods proposed were evaluated by means of extensive and adequate experimental studies. Several multi-label datasets from different domains were used, and the methods were compared to the most significant state-of-the-art algorithms. The results were validated using non-parametric statistical tests. The evidence showed the effectiveness of the methods proposed, proving the hypotheses formulated at the beginning of this thesis.

Table of Contents

List of Acronyms

AAM Algorithm Adaptation Methods

AL Active Learning

AUC Area Under the learning Curve

BMAL Batch-Mode Active Learning

BR Binary Relevance

CC Classifier Chain

CMA-ES Covariance Matrix Adaptation Evolution Strategy

CVIRS Category Vector Inconsistency and Ranking of Scores

DGC Data Gravitation Classification

DM Data Mining

ESBMAL Evolutionary Strategy for Batch-Mode Multi-Label Active Learning

FS Feature Selection

FW Feature Weighting

GA Genetic Algorithm

JCLAL Java Class Library for Active Learning

KDD Knowledge Discovery in Databases

KNN K-Nearest Neighbours

LPS Label Power Set

LR Label Ranking

ML Machine Learning

MLAL Multi-label Active Learning

MLC Multi-label Classification

MLL Multi-label Learning

NSGA-II Non-dominated Sorting Genetic Algorithm II

PPT Pruned Problem Transformation

PTM Problem Transformation Methods

RPC Ranking by Pair-wise Comparison

SSL Semi-Supervised Learning

SVM Support Vector Machine

TP True performance of a selection strategy

Part I: Ph.D. Dissertation

1 Introduction

In the last two decades, the volume of data stored in the Internet has exponentially grown. Currently, it is common to find datasets that contain million of examples[1] and thousands (even millions) of features that describe these examples. Nowadays, the making decision process faces new challenges that arise not only of the complexity of the problem to resolve, but also of the complexity of data that must be processed. The Knowledge Discovery in Databases (KDD) is an important tool in the making decision process through large datasets.

KDD is the process of discovering useful, nontrivial, implicit, and previously unknown knowledge from a collection of data [1]. In KDD, the Data Mining (DM) is a crucial step, where the aim is to discover, through advanced data analysis tools, valid patterns and relationships in datasets [2]. DM uses data analysis tools such as statistical models, mathematical methods, and machine learning algorithms. Machine learning (ML) is a branch of artificial intelligence that focus on the construction of computer algorithms that can learn from data [3].

In the last decade, Multi-label Learning (MLL) has become a popular area of study due to the increasing number of real-world problems that contain multi-label data [4]. The multi-label problems involve examples that belong to a set of

[1] Also known as objects or instances.

labels at the same time. Particular problems involving multi-label data include text categorization [5, 6], semantic annotation of images [7–9], classification of music and videos [10, 11], classification of protein function and gene function [12, 13], chemical data analysis [14] and many more.

Generally speaking, multi-label datasets contain a large number examples and features that describe the examples, e.g. description of texts, images, proteins and genes. Datasets with a large number of examples and features affect in several ways the performance of learning algorithms. For instance, datasets with high dimensionality have a highly negative impact in the efficiency[2], efficacy[3] and effectiveness[4] of the most learning algorithms, know as "The curse of dimensionality" in the literature.

The goal of the MLL paradigm is to learn a model that correctly generalizes unseen multi-label data. On the MLL context two problems have been studied, Multi-label Classification (MLC) and Label Ranking (LR). MLC divides the set of labels into relevant and irrelevant sets, whereas the LR provides an ordering of the labels for a given query example [4, 15].

To date, several MLL algorithms have been proposed. The multi-label algorithms can be divided into two main categories [4, 15]: Problem Transformation Methods (PTM) and Algorithm Adaptation Methods (AAM). The PTM methods transform multi-label datasets into one or more single-label datasets. Then, for each transformed dataset, a single-label classifier is executed, and an aggregation strategy is performed. The Binary Relevance (BR) [15] trains a single-label classifier for each label. The Classifier Chain (CC) [16] is similar to BR, but the dependency between labels is considered. The Ranking by Pair-wise Comparison (RPC) method [17, 18] trains a single-label classifier for each pair of labels. The Label Power Set (LPS) [15] method constructs a new multi-class dataset, where each unique combination of labels is considered as a class of the new multi-class dataset. In studies [19–21], other sophisticated methods based on LPS approach were proposed.

On the other side, the AAM category comprises algorithms that are designed to directly handle multi-label data. In study [22], the Predictive Clustering Trees

[2]The efficiency refers to the amount of computational resources (space and time) used by an algorithm.

[3]The efficacy is related to the probability that has an algorithm to reach an optimal solution.

[4]The effectiveness, or exactness, represents the quality of the solutions found by the algorithms.

method, that has been applied to the MLC task, was proposed. In case [23], an adaptation of the well-known C4.5 algorithm was proposed. Several adaptations of the Artificial Neural Networks have appeared in the literature [24, 25]. In case [26], an extension of the popular AdaBoost algorithm appeared. Several lazy algorithms have been also proposed [27–31].

Despite the large number of studies that exist around the MLL context, there are some open issues to the scientific community. The challenges that arise of learning process from multi-label data inspire the development of new algorithms, mainly focusing in their efficiency, efficacy and effectiveness. Next, the different issues that were faced in the dissertation are introduced, providing their motivation and justification.

Improving performance on complex multi-label data

Generally speaking, multi-label datasets contain a large number of features that describe the examples, e.g. description of texts, images, proteins and genes [5, 6, 32, 33]. The irrelevant, interacting, redundant and noisy features have a highly negative impact in the performance of the learning algorithms. Moreover, the number of features is much bigger than the number of examples in several multi-label applications [5, 32, 33]. On the other hand, in some domains the number of possible labels scales up to hundreds (even thousands) and the distribution of examples per label can be showed in a non-uniform way [11, 12, 32, 34, 35]. Consequently, some multi-label algorithms, specifically lazy algorithms, present a poor performance with regard to time efficiency and effectiveness [36].

Preprocessing techniques have demonstrated to be an important step of KDD process [1, 2]. Feature engineering techniques such as Feature Weighting (FW) and Feature Selection (FS) can significantly improve the performance of learning algorithms [37–39]. FW task assigns a weight to each feature representing the usefulness of the feature to distinguish pattern classes [38]. A weight vector can be used to improve the performance of the lazy algorithms by means of parameterizing the distance function used to retrieve the k-nearest neighbors of a given query example [38]. Furthermore, a weight vector can be used as a ranking of features to guide the search of the best subset of features [40–42]. FS task can be seen as a specific

case of the FW process, where the feature weights are binary values representing whether a feature is removed or conserved. FS tries to reduce the dimensionality, which has a positive effect on the efficiency, effectiveness and comprehensibility of machine learning [37, 43, 44].

A large number of studies related to FW and FS tasks on single-label data have been proposed. However, far less studies related to FW and FS tasks on multi-label context have appeared. In studies [45–52], several feature estimation methods were proposed, all of them focused on the FS task. Generally speaking, the feature estimation process in multi-label data is carried out by means of a PTM. However, these approaches have several limitations. First, the performance of a PTM generally depends on the number of labels of the dataset. Consequently, they are very expensive for domains that contain a moderate number of labels. Second, a drawback of some PTM is that they do not consider label correlations. As a result of the above situations, nowadays the designing process of FW and FS methods faces several challenges and it is an open field of research.

In general, the lazy algorithms do not construct a model from the training set, postponing almost all the process until classification. In this family of algorithms, the K-Nearest Neighbors (KNN) [53] algorithms are the simplest and easiest to understand. The KNN algorithms have shown be useful in several domains [54]. However, the main drawback of these algorithms is that they severely deteriorate in data with high dimensionality, imbalanced data, or when the classes are non-separable or they overlap. In case [36], an extensive experimental study was carried out, where the most significant multi-label algorithms were compared. The results showed that the multi-label lazy algorithms obtained the worst performance for almost all the evaluation metrics considered.

In studies [27–31, 55–57], the most significant lazy approaches to multi-label data have appeared. These previous works are important contributions to MLL. However, it is still necessary the development of lazy methods that do not deteriorate their performance on multi-label data with a large number of features, labels, imbalanced data, etc. The multi-label lazy algorithms consider any feature equally important for classifying a query; yet irrelevant, interacting, redundant and noisy features have a highly negative impact in the effectiveness of these algorithms. In this sense, FW methods can help to improve the performance of lazy algorithms

by an adequate fitting of the feature weights. Examples with relevant set of labels for the classification of a query example can be retrieved by parameterizing the distance function with a weight vector, leading to a superior performance of the lazy algorithms.

On the other hand, there are other interesting lazy approaches that have been successfully applied on single-label data, and these approaches can be easily adapted to multi-label context. For instance, the Data Gravitation Classification (DGC) approach may be effective in the resolution of multi-label problems. DGC is an approach that applies the principles of the universal law of gravitation to resolve ML problems [58]. One advantage of DGC, compared to other techniques, is that it is based on simple principles with high performance levels [59].

Improving performance through unlabeled data

Generally speaking, the majority of the multi-label problems originate from domains where a huge amount of data is commonly available [6, 7, 20, 60–63]. Data labeling is a very expensive process that requires expert handling. In multi-label settings, experts must label each example several times, as each example belongs to various categories. The situation is further complicated when a multi-label dataset with a large number of examples and label classes is analyzed. Consequently, several real scenarios nowadays contain a small number of labeled data and a large number of unlabeled data simultaneously.

The challenges that arise from problems that contain labeled and unlabeled data at the same time motivate the creation of new computational methods capable of using all these information. Most multi-label algorithms that have been proposed in the literature are designed for working on supervised learning environments, i.e. scenarios where all training examples are labeled. Therefore, the multi-label algorithms can have a poor performance in these scenarios which have a small number of labeled examples.

To date, there are two main areas that are concerned with learning models from labeled and unlabeled data, known as Semi-Supervised Learning (SSL) [64] and Active Learning (AL) [65]. SSL and AL attack the same problem, but from different directions. SSL tries to exploit the latent structure of unlabeled data with the goal

of improving label predictions. On the other side, AL is concerned with learning better classifiers by choosing which instances are labeled for training, reducing the costs of data labeling and training an accurate model. AL methods are involved in the acquisition of their own training data. A selection strategy iteratively selects examples from the unlabeled set that seem to be the most informative. Afterwards, an oracle (e.g. a human annotator) annotates the selected examples and they are inserted in the set of labeled data [65].

After more than a decade, an important number of AL methods for single-label data have been proposed (for an interesting survey, see [65]). However, compared to single-label AL, the AL problem within a multi-label context is far less studied. The main challenge in performing AL on multi-label data (MLAL, Multi-label Active Learning) is designing effective strategies that measure the unified informative potential of unlabeled examples across all labels. The most relevant works related to MLAL have appeared in studies [66–70, 70–84].

Most state-of-the-art MLAL strategies employ the Binary Relevance (BR) approach [15] to break down a multi-label problem into several binary classification problems. Consequently, some of these methods are computationally expensive. MLAL strategies are generally tested on the MLC task. However, their performances with regard to the LR task have not been considered. On the other hand, several MLAL strategies have been designed to work with BR-SVM (Binary Relevance with binary Support Vector Machines) as a base classifier. Therefore, the adaptation of these MLAL strategies for working with other type of base classifiers is a hard task to accomplish. In this sense, it would be interesting the development of new MLAL strategies not restricted to a type of base classifier, to directly estimate the utility of the unlabeled examples without using a PTM.

Most state-of-the-art MLAL strategies were designed to select one unlabeled instance at a time. However, in several domains, such as in the speeding up of the process of inducting classifiers with slow training procedures or in systems where a parallel annotation environment is available, the selection of a batch of unlabeled examples is preferred. Batch-mode AL (BMAL) selects a batch of k unlabeled examples in each iteration, in such a way that the selected instances are informative

and the overlapping of information between them is minimal [85]. The most significant works related to performing BMAL on multi-label data appeared in [74, 81]. However, it is considered that this research line has not been studied in depth.

2

Objectives

Due to the complexity and importance of the multi-label learning, in this thesis we formulated the following **scientific problem**: How to increase the possibilities to resolve the multi-label classification and label ranking tasks, in order to obtain significantly better solutions than state-of-the-art multi-label algorithms?

The **general objective** of this thesis was to develop new algorithms with a high performance in the resolution of the multi-label classification and label ranking tasks.

The following **specific objectives** were pursued to successfully accomplish this aim:

- $\mathbf{O}_1$: Develop new feature estimation methods that allow to improve the performance of multi-label algorithms.
- $\mathbf{O}_2$: Design a new multi-label lazy algorithm with a good trade-off between efficiency and effectiveness in its solution to learn from complex multi-label data.
- $\mathbf{O}_3$: Develop new MLAL strategies not restricted to a type of base classifier and they directly handle the multi-label data.

After an extensive bibliographic review, the following **hypotheses** were formulated:

- $\mathbf{H}_1$: If a feature estimation method developed a similarity function more effective in the determination of nearest examples associated to relevant label sets for classification, a significant improvement on the performance of the multi-label lazy algorithms would be achieved.
- $\mathbf{H}_2$: If a feature estimation method guided the search of relevant subsets of features, the performance of multi-label algorithms would improve.
- $\mathbf{H}_3$: If a multi-label lazy algorithm based on a data gravitation model was proposed, it would be competitive with the state-of-the-art multi-label lazy algorithms, and it would also provide a good trade-off between efficiency and effectiveness in its solution.
- $\mathbf{H}_4$: If a MLAL strategy measured the uncertainty on the predictions of the base classifier and the inconsistency of the predicted label sets, it would obtain better solutions than state-of-the-art MLAL strategies.
- $\mathbf{H}_5$: If the BMAL problem on multi-label data was formulated as a multi-objective problem, and it was resolved by means of an evolutionary algorithm, a significant improvement in the solution of the multi-label BMAL problem would be achieved.

In order to achieve the specific objectives and to test the hypotheses formulated, the following **research tasks** were accomplished:

- Analyze the basis of MLL and review the state-of-the-art multi-label algorithms, identifying open problems in MLC and LR.
- Design and implement new feature estimation methods on multi-label data.
- Validate the effectiveness of the feature estimation methods proposed in the improvement of the performance of multi-label lazy algorithms.
- Validate the effectiveness of the feature estimation methods proposed in the multi-label FS task.

- Design and implement a multi-label lazy algorithm based on DGC principles.
- Validate the effectiveness of the multi-label lazy algorithm proposed by means of comparing with the most relevant state-of-the-art lazy algorithms.
- Design and implement a MLAL strategy not restricted to a type of base classifier and that directly handles the multi-label data.
- Design and implement an evolutionary strategy to resolve the multi-label BMAL problem.
- Validate the effectiveness of the MLAL strategies proposed by means of comparing with the most relevant state-of-the-art MLAL strategies.

In the execution of these tasks, the following **scientific methods** were used:

- General methods: the hypothetico-deductive method was used to elaborate the hypotheses and to propose research lines from partial results. The systematic method for the development of computational tools. The bibliographic revision method for the analysis of previous works.
- Logic methods: the method of analysis and synthesis to decompose the information in logical and related parts, simplifying the information to process. The modeling method in the designing of algorithms and computational tools.
- Empirical methods: the experimentation to assess the methods proposed.
- Mathematical methods: statistical tests to validate the quality of the results. The statistical comparisons between algorithms were carried out by means of non-parametric statistical tests as proposed in [86–88].

3

Methodology

This chapter summarizes the methods, tools and dataset used for the development and evaluation of the algorithms proposed in this thesis. Detailed information about the methodology employed in each of the experimental studies is provided in their respective article's documentation.

Multi-label datasets

The multi-label datasets used in all experiments were obtained from the repository of real-world multi-label problems of MULAN library[1] [89]. Multi-label datasets with different scale and from different application domains were included to analyze the behavior of the methods proposed in this thesis.

Table 3.1 shows some statistics of the multi-label datasets. The values of the properties of the Corel16k dataset were averaged over all ten samples used. The label cardinality is the average number of labels per example. The label density is the label cardinality divided by the total number of labels. The label cardinality, label density and different subsets of labels are measures that represent the complexity of a multi-label dataset. The datasets vary in size: from 194 up to 43,907 examples

[1] http://mulan.sourceforge.net/datasets-mlc.html

(n), from 19 up to 52,350 features (d), from 6 up to 374 labels (q), from 15 up to 6555 different subset of labels (d_s), from 1.014 up to 26.044 label cardinality (l_c), and from 0.009 up to 0.485 label density (l_d).

Dataset	**Domain**	**Source**	n	d	q	d_s	l_c	l_d
Arts	Text	[32]	7484	23146	26	599	1,654	0,064
Bibtex	Text	[6]	7395	1836	159	2856	2,402	0,015
Birds	Audio	[90]	645	260	19	133	1,014	0,053
Business	Text	[32]	11214	21924	30	233	1,599	0,053
Cal500	Music	[11]	502	68	174	502	26,044	0,150
Computers	Text	[32]	12444	34096	33	428	1,507	0,046
Corel16k (10 samples)	Image	[7]	13811	500	161	4937	2,867	0,018
Corel5k	Image	[91]	5000	499	374	3175	3,522	0,009
Education	Text	[32]	12030	27534	33	511	1,463	0,044
Emotions	Music	[92]	593	72	6	27	1,869	0,311
Enron	Text	[93]	1702	1001	53	753	3,378	0,064
Entertainment	Text	[32]	12730	32001	21	337	1,414	0,067
Flags	Image	[9]	194	19	7	54	3,392	0,485
Genbase	Biology	[33]	662	1186	27	32	1,252	0,046
Health	Text	[32]	9250	30605	32	335	1,644	0,051
Mediamill	Video	[61]	43907	120	101	6555	4,376	0,043
Medical	Text	[5]	978	1449	45	94	1,245	0,028
Recreation	Text	[32]	12828	30324	22	530	1,429	0,065
Reference	Text	[32]	8027	39679	33	275	1,174	0,035
Scene	Image	[10]	2407	294	6	15	1,074	0,179
Science	Text	[32]	6428	37187	40	457	1,450	0,036
Social	Text	[32]	12111	52350	39	361	1,279	0,033
Society	Text	[32]	14512	31802	27	1054	1,670	0,062
TMC2007-500	Text	[60]	28596	500	22	1341	2,16	0,098
Yeast	Biology	[17]	2417	103	14	198	4,237	0,303

Table 3.1: Some statistics of the benchmark datasets.

Software

The MULAN library [89] was used for the implementation of the algorithms proposed and the existing methods in the literature. MULAN is a Java library which contains several methods for MLL, and it is constructed over the popular data mining tool WEKA [94]. On the other hand, the JCLEC library [95], which is a framework for evolutionary computation, was used to implement those methods that use evolutionary techniques.

Performance evaluation

In all experiments, a stratified 10-fold cross validation method [96] was carried out. To stratify the multi-label data, the methods proposed in [97] were used. Owing to the random nature of the evolutionary techniques, for each experiment, several runs were executed and the average value was calculated. In the experiments that involved lazy algorithms, the best number of neighbors was determined for each classifier on each dataset. In the experiments related to AL, a pool-based scenario [98] was employed.

Several evaluation measures proposed in [4, 15, 36] were used to assess the effectiveness of the multi-label algorithms. The formulation of these measures, as also their interpretations, can be consulted in the articles derived from this thesis.

In all experiments, the results were statistically validated to analyze if there were significant differences between the algorithms compared. The comparisons between algorithms were carried out using non-parametric statistical tests as proposed in [86–88]. The Wilcoxon's test [99] was conducted to compare a pair of algorithms. The Friedman's test [100] was used to perform multiple comparisons. In case that Friedman's test detected significant differences, the Bergmann-Hommel [101] and Shaffer [102] tests were used to perform all pairwise comparisons, and the Hommel procedure [103] was employed to conduct multiple comparisons with a control method.

4

Results

This chapter summarizes the different methods proposed and briefly presents the results achieved in regard to the objectives aimed in this thesis.

Improving performance on complex multi-label data

The performance of learning algorithms, specifically the lazy algorithms, is affected in datasets which have a high dimensionality. Generally speaking, lazy algorithms consider any feature equally important for classifying a query; yet irrelevant, interacting, redundant and noisy features have a highly negative impact in the performance of these algorithms.

In study [104], a feature estimation method for multi-label data was proposed. In this work, a heuristic based on a similarity measure to compute an adequate weight vector was designed. The proposal takes as premise that the similarity between label sets is a good heuristic to estimate the similarity between examples in the feature space. Given a subset of training examples (validation set), a weight vector is estimated. For each validation example, two rankings of examples are calculated, a ranking of examples based on feature space and another ranking of examples

based on label space. The aim is to learn a weight vector that minimizes the distance between the two rankings associated to each validation example. For solving the optimization problem, a Genetic Algorithm (GA) with a real codification was designed.

The best weight vector found by the GA is used to improve the effectiveness of the multi-label lazy algorithms. The weight vector allows the distance function to have a greater probability to recover examples with labels sets more relevant for classification. The effectiveness of the method proposed was tested with the MLkNN [27], BRkNN [28] and IBLRML [30] lazy algorithms. Several evaluation metrics related to MLC and LR tasks were used to assess the effectiveness of the feature estimation method proposed. A statistical test validated that the weighted lazy algorithms, which parameterize their distance functions using the weight vectors learned, obtained significantly better results than the original versions (non-weighted) of the lazy algorithms.

In study [105], a more sophisticated feature estimation method was presented. A new way of computing the rankings of examples is proposed, where only the k nearest neighbors of each validation example are considered. On the other hand, a new metric to compute the distance between the rankings of examples is formulated. In this work, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [106, 107] algorithm was employed for the resolution of the optimization problem. CMA-ES optimizes the metric defined as a heuristic to estimate an adequate weight vector. As in case [104], the main goal was to examine the benefit of feature estimation methods to improve the performance of multi-label lazy algorithms. The effectiveness of the method proposed was tested with the MLkNN [27], BRkNN [28], DMLkNN [29], IBLRML [30] and MLCWkNN [57] algorithms.

In study [108], an extension of the well-known ReliefF algorithm [40] to multi-label data was proposed. The method proposed, named ReliefF-ML, directly estimates the utility of the features, i.e. it does not use any PTM. The concepts *Hits* and *Misses* used by the classic ReliefF algorithm were redefined. ReliefF-ML can be considered a generalization of the classic ReliefF, where the equation used to update the weights was modified. The effectiveness of the method proposed was validated on the improvement of the performance of three multi-label lazy algorithms, MLkNN [27], DMLkNN [29] and MLCWkNN [57].

In study [109], two another extensions of the ReliefF algorithm for multi-label data were proposed. The first extension proposed, named PPT-ReliefF, uses the Pruned Problem Transformation (PPT) method [19] to convert the original multi-label dataset into a new multi-class dataset. PPT has the power of LPS approach, where the correlation among labels is implicitly taken into account, but PPT only considers the most important label relationships. PPT approach reduces the scarcity of labels and the over-fitting of data. The second extension proposed, named RReliefF-ML, is based on the well known ReliefF adaptation to regression problems [110]. RReliefF-ML does not use a PTM for the estimation of the feature weights, it retrieves only k-nearest neighbors for each sampling example. In this work, the two extensions proposed PPT-ReliefF and RReliefF-ML, and the extension ReliefF-ML that was proposed in [108], were compared to other existing state-of-the-art ReliefF extensions. The experimental study showed that the three methods significantly improved the performance of the multi-label lazy algorithms.

On the other hand, the effectiveness of the three methods (PPT-ReliefF, RReliefF-ML and ReliefF-ML) was evaluated in FS task. The weight vectors were converted into feature rankings, the features are ordered according their relevance, and these rankings guided the search of the best subset of features. The evidence suggested that the distributions of the relevant features on the top of the rankings determined by PPTReliefF, ReliefF-ML and RReliefF-ML were better than the distributions of the relevant features determined by the other ReliefF extensions considered in the comparison. The study showed that the baseline classifiers can obtain formidable results on complex multi-label datasets considering a small number of features.

In study [111], a multi-label lazy algorithm based on the principles of DGC approach was proposed. The method proposed, named MLDGC, directly handles multi-label data, and considers each example as an atomic data particle. Considering each example as an atomic data particle, the problems that arise in the creation of artificial particles from several examples are avoided. In this work, the concept of *Neighborhood-based Gravitation Coefficient* was introduced, which is used in the calculation of the gravitation forces. MLDGC has an acceptable computational complexity. MLDGC was compared to 12 multi-label lazy algorithms, confirming the effectiveness of this data gravitation model for better multi-label lazy learning.

The publications associated to this part of the dissertation are:

O. Reyes, C. Morell and S. Ventura. ***Learning Similarity Metric to improve the performance of Lazy Multi-label Ranking Algorithms***. In Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA'2012). IEEE, pp. 246-251, 2012.

O. Reyes, C. Morell and S. Ventura. ***ReliefF-ML: an extension of ReliefF algorithm to multi-label learning***. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. LNCS, Springer, vol. 8259, pp. 528-535, 2013.

O. Reyes, C. Morell and S. Ventura. ***Evolutionary feature weighting to improve the performance of multi-label lazy algorithms***. Integrated Computer-Aided Engineering, vol. 21, no. 4, pp. 339-354, 2014.

O. Reyes, C. Morell and S. Ventura. ***Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context***. Neurocomputing, vol. 161, pp. 168-182, 2015.

O. Reyes, C. Morell and S. Ventura. ***Effective lazy learning algorithm based on a data gravitation model for multi-label learning***. Information Sciences, vol. 340-341, pp. 159-174, 2016.

Improving performance through unlabeled data

The main challenge in performing AL on multi-label data is designing effective strategies that measure the unified informative potential of unlabeled examples across all labels. On the other hand, developing efficient strategies is a crucial point in scenarios where a base classifier which has a costly training process is used, or the time that the expert can wait to label the unlabeled examples is limited.

In study [112], a MLAL strategy, named Category Vector Inconsistency and Ranking of Scores (CVIRS), was proposed. Two uncertainty measures based on the

predictions of the base classifier and the inconsistency of a predicted label set regarding to the label dimension of the labeled dataset, respectively, were defined to select the most uncertain examples. Given an unlabeled example, the difference margins[1] in predictions of classifier with respect to whether the example belongs or does not belong to each label is computed. An example with large margin value on a label means that the classifier has small error in differentiating whether the example belongs or does not belong to this label. On the other hand, an example with small margin value on a label means that it is more ambiguous for the current classifier to predict whether the example belongs or does not belong to this label. The calculus of the unified uncertainty that has the classifier with respect to an unlabeled example was formulated as an rank aggregation problem. A simple and efficient positional method was used to resolve the rank aggregation problem formulated.

On the other hand, a measure that represents the inconsistency of a predicted label set was defined. This measure is based on the premise that as the labeled set and unlabeled set are drawn from the same underlying distribution, is expected that predicted label sets and the label sets of labeled examples share common properties. The inconsistency of a predicted label set is calculated by means of the Hamming and entropic distances between two binary vectors. Based on the two measures defined (uncertainty and inconsistency), CVIRS iteratively selects the unlabeled examples that have high uncertainty levels and, at the same time, high inconsistency in their predicted label sets. This approach can be used with any base classifier which can obtain proper probability estimates from its outputs. The proposal is not restricted to base classifiers that use PTM, it can also be used with multi-label algorithms that belong to AAM category. CVIRS was compared to seven state-of-the-art MLAL strategies, confirming the effectiveness of the proposal for better MLAL.

Most state-of-the-art MLAL strategies were designed to select one unlabeled example at a time. This type of AL strategy can be easily used to select a batch of unlabeled examples, e.g. by selecting the k best instances in a greedy manner, but the information overlapping between the selected instances is not considered. The most significant works related to performing batch-mode AL on multi-label

[1]The difference margin is defined as the difference between the probabilities that an example belongs or does not belong to a particular label.

data appeared in [74, 81]. In these previous works, the batch selection task is commonly formulated as a NP-hard integer programming problem. However, the use of this type of methods is difficult, practically speaking, for their application to large-scale multi-label datasets. On the other hand, most MLAL strategies only use informativeness-based[2] criteria to select the most useful unlabeled examples, leading to a sub-optimal performance [82]. Other types of selection criteria, such as representativeness[3] and diversity[4], have been rarely considered in the MLAL context. Few works have combined two selection criteria to select the best unlabeled examples [74, 78, 79, 81, 82], notably informativeness and representativeness, or informativeness and diversity. To date, to the best of our knowledge, a MLAL strategy that combines the three criteria (informativeness, representativeness and diversity) had not been proposed.

In study [113], a MLAL strategy, named Evolutionary Strategy for Batch-Mode Multi-Label Active Learning (ESBMAL), was proposed. ESBMAL formulates the BMAL problem as a multi-objective optimization problem, and the optimization problem is solved by the well-known NSGA-II algorithm [114]. The evolutionary algorithm tries to optimize three measures based on informativeness, diversity and representativeness, respectively. An individual of the population represents a candidate batch of examples. In each AL iteration, ESBMAL aims to select a set of unlabeled examples which are usually informative across all labels, diverse between each other, and representative of the underlying distribution. ESBMAL can be used with any base multi-label classifier which can obtain proper probability estimates from its outputs. ESBMAL is more efficient, in computational terms, than state-of-the-art multi-label BMAL strategies. The experimental study showed the effectiveness of the proposal for better multi-label BMAL.

Next, complementary results derived of this thesis are briefly exposed:

- Currently, there are several software tools which assist the experimentation process and development of new algorithms in DM and ML areas, such as Rapid Miner, WEKA, Scikit-learn, Orange and KEEL. However, these tools are focused to supervised and unsupervised learning problems. To date,

[2]Informativeness measures the effectiveness of an example by reducing the uncertainty of a model.

[3]Representativeness measures whether an unlabeled example is representative of the underlying distribution.

[4]Diversity measures the information redundancy that exist among a set of examples.

there has been insufficient effort towards the creation of a computational tool mainly focused to AL.

The above situation motivated the development of the JCLAL[5] (Java Class Library for Active Learning) framework [115]. JCLAL is an open source software for researchers and end-users to develop AL methods. JCLAL aims to bring the benefits of open source software to people working in the area of AL. It includes the most relevant strategies that have been proposed in single-label and multi-label learning paradigms. It provides the necessary interfaces, classes and methods to develop any AL method.

JCLAL is an open source project under the GNU General Public License (GPL). It has an architecture that follows strong principles of object-oriented programming, where it is common and easy to reuse code. Through a flexible class structure, the library provides the possibility of including new AL methods, as well as the ability to adapt, modify or extend the framework according to developer's needs.

- Despite the call made by the ML community for a rigorous and correct statistical analysis of published results, the use of statistical tests for analyzing the performance of AL methods has not been rigorous. Through an extensive bibliographic review of works published in the AL area, we observed that many excellent and innovative AL papers end by drawing conclusions by means of visually comparing learning curves.

  The visual comparison of learning curves is effective when a small number of AL strategies are compared, and their performances differ sufficiently so that the learning curves do not overlap greatly. Conversely, the visual comparison of several learning curves can be very confusing, as the learning curves may intersect at many points. If several active learning strategies are compared over multiple datasets, and their performances are similar over multiple datasets, the resulting graphs may be very difficult to interpret, and the visual analysis of the AL performance may be a very difficult task to accomplish. Consequently, conclusions from questions such as, which is the AL strategy that delivers the best performance?, are not possible, or very difficult to draw.

[5] http://jclal.sourceforge.net

In study [116], two approaches, based on the use of non-parametric statistical tests, to statistically compare AL strategies over multiple datasets were proposed. The first approach is based on the analysis of the Area Under learning Curve (AUC) and the rate of performance change. The concept *True Performance of a selection strategy* (TP) is defined. A TP score can be interpreted as a general view of the performance of an AL strategy. After computing the TP scores of the AL strategies on each dataset, a statistical analysis can be carried out, and then final considerations could be given with a statistical support.

The second approach, instead of only considering the final results (TP scores), analyzes the intermediate results generated in each iteration of the AL process. This can reveal very significant information when AL strategies are compared, especially in cases where TP scores are statistically similar. The application of both approaches was illustrated by means of an experimental study, demonstrating the usefulness of the proposal for improving analysis of AL performance.

The publications associated to this part of the dissertation are:

O. Reyes, C. Morell and S. Ventura. ***Effective active learning strategy for multi-label learning***. Neurocomputing, submitted, 2015.

O. Reyes and S. Ventura. ***Evolutionary Strategy to perform Batch-Mode Active Learning on Multi-label Data***. ACM Transactions on Intelligent Systems and Technology, submitted, 2016.

O. Reyes, E. Pérez, M. C. Rodríguez Hernández, H. M. Fardoun and S. Ventura. ***JCLAL: A Java Framework for Active Learning***. Journal of Machine Learning Research, vol. 17 (95), pp. 1-5, 2016.

O. Reyes, A. H. Altahi and S. Ventura. ***Statistical Comparisons of Active Learning Strategies over Multiple Datasets***. Information Sciences, submitted, 2016.

5

Conclusions and future work

This chapter briefly summarizes the concluding remarks obtained from the research of the thesis and provides research lines for future work.

Conclusions

This Ph.D. thesis focused on MLL paradigm. The bibliographic study allowed to detect open problems, and formulate the hypotheses that guided this research. The work developed cannot be considered to be concluded, due to the extent of the topics treated and their possible application to other areas.

Improving performance on complex multi-label data

The first part of the thesis focused on the improvement of the performance of multi-label algorithms, specifically the lazy algorithms, on complex multi-label data.

In studies [104, 105, 108, 109], five feature estimation methods were proposed. Two methods are based on the application of evolutionary algorithms to estimate an adequate weight vector. The three other methods proposed are extensions of the

well-known ReliefF algorithm. The methods based on ReliefF approach are more computationally efficient than the two other based on evolutionary techniques.

The results showed that it is possible to obtain a good estimation of the feature weights by directly handling the multi-label data, i.e. without using a PTM. The parameterization of the distance functions with a weight vector allowed to recover examples with relevant label sets for classification. The evidence showed that the methods proposed significantly improve the performance of the multi-label lazy algorithms on complex multi-label data, proving the hypothesis $\mathbf{H}_1$ formulated in this thesis.

On the other hand, a weight vector can be useful to guide the search process of the best subsets of features. In case [109], it is showed how by converting the weight vectors into feature rankings, it is possible to select small subsets of features efficiently, leading to a significant improvement of the performance of multi-label algorithms. The evidence showed that the methods perform well on the FS task, proving the hypothesis $\mathbf{H}_2$ formulated in this thesis.

In study [111], a multi-label lazy algorithm based on DGC approach was proposed. Considering each example as an atomic data particle avoided the problems that arise in the creation of artificial particles from several examples. The introduction of the new concept *Neighborhood-based Gravitation Coefficient* achieved better levels of classification. This coefficient strengthens or weakens the effect that a particle has over a test example. Two particles at the same distance of a test example, but with different levels of purity in their neighborhood, will exert different gravitational forces. The evidence showed that the method proposed significantly outperformed the state-of-the-art multi-label lazy algorithms. The method provides a good trade-off between efficiency and effectiveness in its solution, proving the hypothesis $\mathbf{H}_3$ formulated in this thesis.

In general, the specific objectives $\mathbf{O}_1$ and $\mathbf{O}_2$, declared at the beginning of this thesis, were fulfilled through the results obtained in the works [104, 105, 108, 109, 111].

Improving performance through unlabeled data

The second part of the thesis focused on the development of MLAL strategies not restricted to a type of base classifier, and they directly handle the multi-label data. The multi-label BMAL problem was also analyzed.

In study [112], an efficient MLAL strategy was proposed. Two measures to select the unlabeled examples were defined. The first measure is related to the uncertainty in the predictions of the base classifier. A rank aggregation problem was formulated to compute the unified uncertainty of an unlabeled example, and this problem was solved by an efficient positional method. The second measure is related to the inconsistency of the predicted labels sets. The combination of these two measures allowed to select examples that not only are informative for the current model, but they are also not representative of the data distribution of the labeled set. The method proposed can be used with any base classifier which can obtain proper probability estimates from its outputs. The evidence showed that the method significantly outperformed several state-of-the-art MLAL strategies, proving the hypothesis $\mathbf{H}_4$ formulated in this thesis.

In study [113], a multi-label BMAL strategy was proposed. Three measures based on informativeness, representativeness and diversity were defined, respectively. The multi-label BMAL problem was formulated as a multi-objective problem, and the optimization problem was solved by an evolutionary algorithm. The solutions reached by the evolutionary algorithm represent sets of examples which are usually informative across all labels, diverse between each other, and representative of the underlying distribution. The results showed that the multi-objective problem formulated is a good heuristic to resolve the multi-label BMAL problem, as also that the evolutionary algorithms are effective in the resolution of this type of problem. The method is more computationally efficient than other existing approaches, which commonly formulate the multi-label BMAL problem as a complex integer programming problem. The method can be used with any base multi-label classifier which can obtain proper probability estimates from its outputs. The evidence showed that the method significantly outperformed the state-of-the-art multi-label BMAL strategies, proving the hypothesis $\mathbf{H}_5$ formulated in this thesis.

As complementary results derived of this thesis, a computational tool that allows the implementation of AL methods in a simple manner and favors the experimentation on this area was developed [115]. This framework was announced to AL community in November 2014 and has had a good acceptation.

Finally, in study [116], two approaches to assess the performance of AL methods were proposed. The first approach is based on the analysis of the AUC and the rate of performance change. The second approach analyses the intermediate results derived from AL iterations. The second approach is more robust and powerful than the first one, it is able to detect less significant differences. The evidence showed the usefulness of non-parametric tests in the evaluation of AL performance.

In general, the specific objective $\mathbf{O}_3$, declared at the beginning of this thesis, was fulfilled through the results obtained in the works [112, 113].

Future work

In this section, some remarks for future lines of research that arise from the studies developed in this thesis are provided.

To date, there are still some issues that remain far less studied in the MLL paradigm. Through the bibliographic study carried out in the development of this work, the following promising research lines were detected:

- Instance selection algorithms for multi-label data. To date, very few works have been proposed in this sense.
- Imbalanced learning techniques for multi-label data. In last years, this line of research has gained the attention of the scientific community due to multi-label datasets commonly show a non-uniform distribution of examples per label.
- Algorithms for multi-label data streams. In this sense, it is important the development of incremental multi-label algorithms.
- Dimensionality reduction in the label space. To date, very few proposals have been presented in this sense.
- SSL algorithms for multi-label data. SSL algorithms for multi-label data, in comparison to single-label data, have been far less studied. On the other hand, the combination of AL and SSL approaches is an interesting research line.

On the other hand, the following research lines are also proposed from the results obtained in this work:

- Define new heuristics to learn similarity metrics in order to improve the performance of multi-label lazy algorithms.
- Propose feature estimation methods based on evolutionary techniques that allow to effectively select subsets of relevant features on multi-label data.
- Propose other models based on DGC approach for better multi-label lazy learning.

- Design new AL strategies based on evolutionary techniques to effectively resolve the multi-label BMAL problem.
- Extend the AL methods proposed in this thesis to select example-label pairs, instead of consulting all possible labels of the selected examples. The selection of example-label pairs, taking into account the dependence between labels, can lead to a considerable reduction on the data labeling cost.
- Adapt the methods proposed in this thesis to the multi-instance multi-label problem. In multi-instance multi-label problems, examples are described by multiple instances and they are associated with multiple class labels.
- Extend JCLAL library by including other AL strategies, for instance AL strategies for multi-instance learning and multi-instance multi-label learning. On the other hand, it would be interesting the development of a module that allows the distributed computation of AL strategies, thus enabling the use of the library in Big Data area.

Bibliography

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37–54, 1996.

[2] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, 2nd ed. New Jersey, United States of America: John Wiley & Sons, 2014.

[3] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

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

[5] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch, "A shared task involving multi-label classification of clinical free text," in *Proceedings of the Workshop on Biological, Translational, and Clinical Language Processing (BioNLP'2007)*. Stroudsburg, PA, United States of America: Association for Computational Linguistics, 2007, pp. 97–104.

[6] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proceedings of the ECML/PKDD Discovery Challenge*, vol. 75, 2008.

[7] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan, "Matching words and pictures," *Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[8] S. Yang, S. Kim, and Y. Ro, "Semantic home photo categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 324–335, 2007.

[9] E. Correa, A. Plastino, and A. Freitas, "A Genetic Algorithm for Optimizing the Label Ordering in Multi-Label Classifier Chains," in *Proceedings of the 25th International Conference on Tools with Artificial Intelligence (ICTAI'2013)*. IEEE, 2013, pp. 469–476.

[10] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[11] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.

[12] F. Otero, A. Freitas, and C. Johnson, "A hierarchical multi-label classification ant colony algorithm for protein function prediction," *Memetic Computing*, vol. 2, no. 3, pp. 165–181, 2010.

[13] M. G. Larese, P. Granitto, and J. Gómez, "Spot defects detection in cDNA microarray images," *Pattern Analysis and Applications*, vol. 16, no. 3, pp. 307–319, 2013.

[14] E. Ukwatta and J. Samarabandu, "Vision based metal spectral analysis using multi-label classification," in *Canadian Conference on Computer and Robot Vision (CRV'2009)*. IEEE, 2009, pp. 132–139.

[15] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Data Mining and Knowledge Discovery Handbook*, 2nd ed. New York, United States of America: Springer-Verlag, 2010, ch. Mining Multi-label Data, pp. 667–686.

[16] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, 2011.

[17] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2001, pp. 681–687.

[18] J. Furnkranz, E. Hullermeier, E. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[19] J. Read, "A pruned problem transformation method for multi-label classification," in *Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRS'2008)*, 2008, pp. 143–150.

[20] G. Tsoumakasa, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proceedings of the ECML/PKDD Workshop on Mining Multidimensional Data (MMD'2008)*, 2008, pp. 30–44.

[21] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1079–1089, 2011.

[22] H. Blockeel, L. Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the 15th International Conference on Machine Learning*, 1998, pp. 55–63.

[23] A. Clare and R. King, "Knowledge discovery in multi-label phenotype data," in *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'2001)*. Springer, 2001, pp. 42–53.

[24] K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," *Journal of Machine Learning Research*, vol. 3, pp. 1025–1058, 2003.

[25] M. L. Zhang and Z. H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 1338–1351, 2006.

[26] R. Schapire and Y. Singer, "Boostexter: a boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2, pp. 135–168, 2000.

[27] M. L. Zhang and Z. H. Zhou, "ML-kNN: A lazy learning approach to multilabel learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[28] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An empirical study of lazy multi-label classification algorithms," in *Artificial Intelligence: Theories, Models and Applications*. Springer, 2008, pp. 401–406.

[29] Z. Younes, F. Abdallah, and T. Denceux, "Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies," in *Proceedings of the 16th Eropean Signal Processing Conference (EUSIPCO'2008)*. IEEE, 2008, pp. 1–5.

[30] W. Cheng and E. Hullermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Machine Learning*, vol. 76, no. 2-3, pp. 211–225, 2009.

[31] Z. Younes, F. Abdallah, and T. Denceux, "An Evidence-Theoretic K-Nearest Neighbor Rule for Multi-label Classification," in *Scalable Uncertainty Management*, ser. LNAI, I. Godo and A. Pugliese, Eds., vol. 5785. Springer, 2009, pp. 297–308.

[32] N. Ueda and K. Saito, "Parametric mixture models for multi-labeled text," in *Proceedings of Advances in Neural Information Processing Systems (NIPS'2015)*. MIT Press, 2002, pp. 737–744.

[33] S. Diplarisa, G. Tsoumakas, P. Mitkas, and I. Vlahavas, "Protein classification with multiple algorithms," in *Proceedings of the 10th Panhellenic Conference on Informatics (PCI'2005)*, ser. LNCS, vol. 3746. Springer, 2005, pp. 448–456.

[34] S. Dendamrongvit, P. Vateekul, and M. Kubat, "Irrelevant attributes and imbalanced classes in multi-label text-categorization domains," *Intelligent Data Analysis*, vol. 15, no. 6, pp. 843–859., 2011.

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

[36] G. Madjarov, D. Kocev, and D. Gjorgjevikj, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, pp. 3084–3104, 2012.

[37] K. Kira and L. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth International Workshop on Machine learning*. Morgan Kaufmann, 1992, pp. 249–256.

[38] D. Wettschereck, D. W. Aha, and T. Mohri, "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms," *Artificial Intelligence Review*, vol. 11, pp. 273–314, 1997.

[39] A. Abraham, E. Corchado, and J. Corchado, "Hybrid learning machines," *Neurocomputing*, vol. 72, pp. 2729–2730, 2009.

[40] I. Kononenko, "Estimating attributes: analysis and extensions of ReliefF," in *Proceedings of the European Conference on Machine Learning (ECML'1994)*. Catania, Italy: Springer, 1994, pp. 171–182.

[41] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53 (1-2), pp. 23–69, 2003.

[42] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Heuristic search over a ranking for feature selection," in *Proceedings of IWANN'2005, Computational intelligence and bioinspired systems*, ser. LNCS, vol. 3512. Springer, 2005, pp. 742–749.

[43] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," in *Proceedings of the 20th International Conference on Machine Learning (ICML'2000)*, Washington DC, 2003, pp. 856–863.

[44] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[45] M. L. Zhanga, J. M. Peña, and V. Robles, "Feature selection for multi-label naive bayes classification," *Information Sciences*, vol. 179, pp. 3218–3229, 2009.

[46] Q. Gu, Z. Li, and J. Han, "Correlated multi-label feature selection," in *Proceedings of the 20th ACM international Conference on Information and knowledge Management (CIKM'2011)*. Scotland, United Kingdom: ACM, 2011.

[47] N. Spolaôr, E. Cherman, and M. Monard, "Using ReliefF for multi-label feature selection," in *Proceedings of the Conferencia Latinoamericana de Informática*, Brazil, 2011, pp. 960–975.

[48] N. Spolaôr, E. Cherman, M. Monard, and H. Lee, "Filter approach feature selection methods to support multi-label learning based on ReliefF and Information Gain," in *Proceedings of the Advances in Artificial Intelligence (SBIA'2012)*, ser. LNCS, vol. 7589. Springer, 2012, pp. 72–81.

[49] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label ReliefF and F-statistic feature selections for image annotation," in *Proceedings of Computer Vision and Pattern Recognition (CVPR'2012)*. IEEE, 2012, pp. 2352–2359.

[50] J. Lee and D. W. Kim, "Feature selection for multi-label classification using multivariate mutual information," *Pattern Recognition Letters*, vol. 34, pp. 349–357, 2013.

[51] N. Spolaôr, E. Alvares, M. Carolina, and H. Diana, "A comparison of multi-label feature selection methods using the problem transformation approach," *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135–151, 2013.

[52] G. Doquire and M. Verleysen, "Mutual information-based feature selection for multilabel classification," *Neurocomputing*, 2013.

[53] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[54] I. Kononenko and M. Kukar, *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Cambridge, United Kingdom: Horwood Publishing, 2007.

[55] K. Brinker and E. Hüllermeier, "Case-based multilabel ranking," in *Proceedings of the 20th International Conference on Artificial Intelligence (IJCAI'2007)*, 2007, pp. 702–707.

[56] Z. Younes, F. Abdallah, and T. Denoux, "Fuzzy multi-label learning under veristic variables," in *Proceedings of International Conference on Fuzzy Systems*. IEEE, 2010, pp. 1–8.

[57] J. Xu, "Multi-label weighted k-nearest neighbor classifier with adaptive weight estimation," in *Proceedings of the ICONIP'2011, Neural Information Processing*, ser. LNCS, vol. 7073. Springer, 2011, pp. 79–88.

[58] L. Peng, B. Peng, Y. Chen, and A. Abraham, "Data gravitation based classification," *Information Sciences*, vol. 179, no. 6, pp. 809–819, 2009.

[59] A. Cano, A. Zafra, and S. Ventura, "Weighted Data Gravitation Classification for Standard and Imbalanced Data," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1672–1687, 2013.

[60] A. Srivastava and B. Zane-Ulman, "Discovering recurring anomalies in text reports regarding complex space systems," in *Proceedings of the Aerospace Conference*. IEEE, 2005, pp. 55–63.

[61] C. Snoek, M. Worring, J. van Gemert, J. Geusebroek, and A. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proceedings of the 14th annual ACM International Conference on Multimedia*. Santa Barbara, United States of America: ACM, 2006, pp. 421–430.

[62] E. L. Mencía and J. Furnkranz, "Efficient pairwise multi-label classification for large-scale problems in the legal domain," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'2008)*. Antwerp, Belgium: Springer-Verlag, 2008, pp. 50–65.

[63] T. S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. T. Zheng, "NUS-WIDE: A Real-World Web Image Database from National University of Singapore," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. Greece: ACM, 2009.

[64] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, 2009.

[65] B. Settles, *Active Learning*, 1st ed., ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012.

[66] X. Li, L. Wang, and E. Sung, "Multi-label SVM active learning for image classification," in *Proceedings of the International Conference on Image Processing (ICIP'2004)*, vol. 4. IEEE, 2004, pp. 2207–2210.

[67] K. Brinker, *From Data and Information Analysis to Knowledge Engineering.* Springer, 2006, ch. On Active Learning in Multi-label Classification, pp. 206–213.

[68] G. Qi, X. Hua, Y. Rui, J. Tang, and H. Zhang, "Two-dimensional active learning for image classification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'2008).* IEEE, 2008, pp. 1–8.

[69] ——, "Two-dimensional multi-label active learning with an efficient online adaptation model for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, no. 1, 2009.

[70] B. Yang, J. Sun, T. Wang, and Z. Chen, "Effective multi-label active learning for text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* Paris, France: ACM, 2009, pp. 917–926.

[71] X. Zhang, J. Cheng, C. Xu, H. Lu, and S. Ma, "Multi-view multi-label active learning for image classification," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'2009).* IEEE, 2009, pp. 258–261.

[72] A. Esuli and F. Sebastiani, "Active Learning Strategies for Multi-Label Text Classification," in *Advances in Information Retrieval.* Springer, 2009, pp. 102–113.

[73] M. Singh, E. Curran, and P. Cunningham, "Active learning for multi-label image annotation," in *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*, 2009, pp. 173–182.

[74] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, "Optimal Batch Selection for Active Learning in Multi-label Classification," in *Proceedings of the 19th ACM international conference on Multimedia (MM's2011).* Scottsdale, Arizona, United States of America: ACM, 2011, pp. 1413–1416.

[75] C. W. Hung and H. T. Lin, "Multi-label active learning with auxiliary learner," in *Proceedings of the Asian Conference on Machine Learning.* JMLR, 2011, pp. 315–330.

[76] P. Wang, P. Zhang, and L. Guo, "Mining multi-label data streams using ensemble-based active learning," in *Proceedings of the 12th SIAM International Conference on Data Mining*, 2012, pp. 1131–1140.

[77] J. Tang, Z.-J. Zha, D. Tao, and T.-S. Chua, "Semantic-gap-oriented active learning for multilabel image annotation," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2354–2360, 2012.

[78] X. Li and Y. Guo, "Active Learning with Multi-Label SVM Classification," in *Proceedings of the 23th International joint Conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1479–1485.

[79] S. Huang and Z. Zhou, "Active query driven by uncertainty and diversity for incremental multi-label learning," in *Proceedings of 13th International Conference on Data Mining*. IEEE, 2013, pp. 1079–1084.

[80] J. Wu, V. Sheng, J. Zhang, P. Zhao, and Z. Cui, "Multi-label active learning for image classification," in *Proceedings of the International Conference on Image Processing*. IEEE, 2014, pp. 5227–5231.

[81] B. Zhang, Y. Wang, and F. Chen, "Multilabel image classification via high-order label correlation driven active learning," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1430–144, 2014.

[82] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 1936–1949, 2014.

[83] D. Vasisht and A. Damianou, "Active learning for sparse bayesian multilabel classification," in *Proceedings of the 20th SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 472–481.

[84] S. Huang, S. Chen, and Z. Zhou, "Multi-label active learning: Query type matters," in *Proceedings of the 24th International Conference on Artificial Intelligence*. AAI Press, 2015, pp. 946–952.

[85] Y. Fu, X. Zhu, and A. K. Elmagarmid, "Active learning with optimal instance subset selection," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, 2013.

[86] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[87] S. García and F. Herrera, "An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[88] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, pp. 2044–2064, 2010.

[89] G. Tsoumakas, E. Spyromitros-Xioufi, J. Vilcek, and I. Vlahavas, "MULAN: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[90] F. Briggs and et. al., "The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment," in *Proceedings of the International Workshop on Machine Learning for Signal Processing (MLSP'2013)*. IEEE, 2013.

[91] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proceedings of the 7th European Conference on Computer Vision*, ser. LNCS, vol. 2353. Springer, 2002, pp. 97–112.

[92] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multilabel classification of music into emotions," in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'2008)*, 2008, pp. 325–330.

[93] B. Klimt and Y. Yang, "The Enron corpus: a new dataset for email classification research," in *Proceedings of the 15th European Conference on Machine Learning (ECML'2004)*. Springer, 2004, pp. 217–226.

[94] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," in *SIGKDD Explorations*, vol. 11, no. 1. ACM, 2009, pp. 10–18.

[95] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás, "JCLEC: A java framework for evolutionary computation," *Soft Computing*, vol. 12, pp. 381–392, 2008.

[96] R. Kohavi, "A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, 1995, pp. 1137–1143.

[97] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 145–158.

[98] D. Lewis and W. Gale, "A sequential algorithm for training text classifier," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Dublin, Ireland: Springer, 1994, pp. 3–12.

[99] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.

[100] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, pp. 86–92, 1940.

[101] G. Bergmann and G. Hommel, "Improvements of general multiple test procedures for redundant systems of hypotheses," *Multiple Hypotheses Testing*, pp. 100–115, 1988.

[102] J. Shaffer, "Modified sequentially rejective multiple test procedures," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 826–831, 1986.

[103] G. Hommel, "A stagewise rejective multiple test procedure based on a modified bonferroni test," *Biometrika*, vol. 75, no. 2, pp. 383–386, 1988.

[104] O. Reyes, C. Morell, and S. Ventura, "Learning similarity metric to improve the performance of lazy multi-label ranking algorithms," in *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA'2012)*. IEEE, 2012, pp. 246–251.

[105] ——, "Evolutionary feature weighting to improve the performance of multi-label lazy algorithms," *Integrated Computer-Aided Engineering*, vol. 21, no. 4, pp. 339–354, 2014.

[106] N. Hansen and A.Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[107] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2. IEEE, 2005, pp. 1769–1776.

[108] O. Reyes, C. Morell, and S. Ventura, "ReliefF-ML: an extension of ReliefF algorithm to multi-label learning," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. LNCS. Springer, 2013, vol. 8259, pp. 528–535.

[109] ——, "Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context," *Neurocomputing*, vol. 161, pp. 168–182, 2015.

[110] M. Robnik-Šikonja and I. Kononenko, "An adaptation of Relief for attribute estimation in regression," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, 1997, pp. 296–304.

[111] O. Reyes, C. Morell, and S. Ventura, "Effective lazy learning algorithm based on a data gravitation model for multi-label learning," *Information Sciences*, vol. 340-341, pp. 159–174, 2016.

[112] ——, "Effective active learning strategy for multi-label learning," *Neurocomputing, submitted*, 2015.

[113] O. Reyes and S. Ventura, "Evolutionary strategy to perform batch-mode active learning on multi-label data," *ACM Transactions on Intelligent Systems and Technology, submitted*, 2016.

[114] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[115] O. Reyes, E. Pérez, M. C. Rodríguez-Hernández, H. M. Fardoun, and S. Ventura, "JCLAL: A Java Framework for Active Learning," *Journal of Machine Learning Research*, vol. 17, pp. 1–5, 2016.

[116] O. Reyes, A. H. Altahi, and S. Ventura, "Statistical comparisons of active learning strategies over multiple datasets," *Information Sciences, submitted*, 2016.

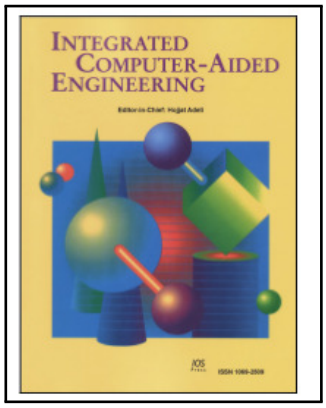
Part II: Journal Publications

TITLE:

Evolutionary feature weighting to improve the performance of multi-label lazy algorithms

AUTHORS:

O. Reyes, C. Morell, and S. Ventura



Integrated Computer-Aided Engineering, *Volume 21, pp. 339-354, 2014*

RANKING:

Impact factor (JCR 2014): 4.698

Knowledge area:

Computer Science, Interdisciplinary Applications: 2/102

Computer Science, Artificial Intelligence: 5/123

DOI: 10.3233/ICA-140468

Integrated Computer-Aided Engineering 21 (2014) 339–354
DOI 10.3233/ICA-140468
IOS Press

Evolutionary feature weighting to improve the performance of multi-label lazy algorithms

Oscar Reyes[a], Carlos Morell[b] and Sebastián Ventura[c,d,*]
[a]*Computer Science Department, University of Holguín, Holguín, Cuba*
[b]*Computer Science Department, Universidad Central de Las Villas, Santa Clara, Cuba*
[c]*Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain*
[d]*Information Systems Department, King Abdulaziz University, Jeddah, Saudi Arabia*

Abstract. In the last decade several modern applications where the examples belong to more than one label at a time have attracted the attention of research into machine learning. Several derivatives of the k-nearest neighbours classifier to deal with multi-label data have been proposed. A k-nearest neighbours classifier has a high dependency with respect to the definition of a distance function, which is used to retrieve the k-nearest neighbours in feature space. The distance function is sensitive to irrelevant, redundant, and interacting or noise features that have a negative impact on the precision of the lazy algorithms. The performance of lazy algorithms can be significantly improved with the use of an appropriate weight vector, where a feature weight represents the ability of the feature to distinguish pattern classes. In this paper a filter-based feature weighting method to improve the performance of multi-label lazy algorithms is proposed. To learn the weights, an optimisation process of a metric is carried out as heuristic to estimate the feature weights. The experimental results on 21 multi-label datasets and 5 multi-label lazy algorithms confirm the effectiveness of the feature weighting method proposed for a better multi-label lazy learning.

Keywords: Feature weighting, lazy learning algorithms, multi-label classification, label ranking, learning metric, evolutionary algorithms

1. Introduction

In the last few decades, studies in the field of supervised learning have dealt with the analysis of data where the examples were associated with a single label [47,49,66]. However, there are several real problems where the examples belong to a set of labels at the same time, known as multi-label problems [63]. In the last few years an increasing number of modern applications that contain multi-label data have appeared, such as text categorisation [38], emotions evoked by music [35], semantic annotation of images [73] and videos [8], classification of protein function and gene [76].

Several multi-label lazy algorithms derivate of the k-nearest neighbours (k-NN) classifier scheme have been proposed on the multi-label learning context [12,56,72, 74,77]. In general, these algorithms do not construct a model from the training set, postponing almost all the process until classification. They classify a query by retrieving its k-nearest neighbours in feature space and after that, an aggregation strategy is performed to predict the set of labels of a query instance [63]. In the same way of single-label k-NN classifier, the multi-label lazy algorithms have a high dependency with respect to the definition of a distance function that is used to determine the k-nearest neighbours of a query instance. The main disadvantage of the multi-label lazy algorithms is that they consider any feature equally important for classifying a query; yet irrelevant, interacting, redundant and noisy features have a highly negative impact in the precision of these algorithms [67].

*Corresponding author: Sebastián Ventura, Department of Computer Science and Numerical Analysis, University of Córdoba, Albert Einstein Building, Rabanales Campus, Córdoba, Spain. Tel.: +34 957 212 218; Fax: +34 957 218 630; E-mail: sventura@uco.es.

ISSN 1069-2509/14/$27.50

Due to the fact that lazy learning algorithms use a similarity or distance function based in feature space, the feature weighting methods can be easily used to improve the performance of the lazy algorithms [67]. The aim is to find a weight vector W that allows the distance function to recover the k-nearest neighbours in the feature space, given that not all features have the same relevance in a dataset, where each feature weight represents the usefulness of the feature.

Several feature weighting methods to improve the performance of single-label lazy algorithms have been proposed [67]. There is still not a comprehensive treatment in the literature, however, of how to make feature weighting on multi-label data to improve the performance of multi-label lazy algorithms.

Several feature estimation algorithms to multi-label data have been proposed [33,55,59], however, they have been focused on the multi-label feature selection process. Generally, the estimation of the usefulness of features in multi-label data have been done using the problem transformation methods [63], where the multi-label problems are converted into one or more single-label problems, and following some aggregation strategy the score of a feature can be computed using any single-label feature estimation algorithm. The efficiency of these multi-label feature weighting algorithms can be affected on datasets that contain a big number of labels.

In [46] an evolutionary strategy to estimate the feature weights on multi-label data to improve the performance of the multi-label lazy algorithms was proposed. The search process of the best weight vector was performed using a genetic algorithm (GA) [1,4,25, 28,53]. For every sampling instance i two orders are computed: the feature-based instances order, and the label-based instances order. After that, the distance between these two orders is quantified by the Fréchet's permutation metric [16]. The goal of this method is to find a weight vector that minimises the disorder caused by the metric used to compute the distance among the orders generated over all selected instances from the training set.

The previously described method has several disadvantages. Firstly, the iterative computation of the orders for every sampling instance is very expensive for large multi-label datasets, because the method generates orders with cardinality equal to the number of instances in the training set. Second, the metric used to compute the distance between the two orders of an instance does not take into account the possible ties that can appear among some components of the orders. Additionally, for large datasets the number of permutations of instances among orders is high, resulting in the method's poor performance. The third (and not the last) disadvantage is that the searching process, for the best weight vector by GA, is very expensive in more complex multi-label datasets, owing to the number of individuals and generations needed for the GA to converge into an optimal solution.

In this work a new filter-based multi-label feature weighting method that extends and improves the preliminary studies presented in [46] is proposed. In this new approach the manner of constructing the orders with respect to a sampling instance i is different, where only the k-nearest neighbours are considered. Furthermore, an extension of Fréchet's permutation metric that takes into account the possible ties that can appear among some components of the groups of k-nearest neighbours is proposed. Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [2,24] performs the learning of the weight vector W. CMA-ES optimises the defined metric as a heuristic to estimate the feature weights. In the end, the main goal of this process is to introduce the weight vector W into the distance function used by the lazy algorithms, allowing the recovery of those nearest examples in the feature space that are associated with the major confidence set of labels for classifying a query.

The main motivation of the present work is to examine the benefits of feature weighting methods to improve the performance of multi-label lazy algorithms. The new method proposed was proved on 5 multi-label k-NN algorithms and 21 multi-label datasets. In the experimental study, the performance of the original methods and their extensions that use the weight vector learned were compared. Also, a comparative study among the new method proposed and the most relevant multi-label feature weighting methods was carried out. The experimental stage shows the effectiveness of the proposal to improve the performance of multi-label lazy algorithms in multi-label classification and label ranking tasks. In the experiments, 7 multi-label evaluation measures were used to analyse different points of view.

This paper is arranged as follows. In Section 2 a formal definition of the multi-label learning task, a description of the principal multi-label lazy algorithms and the multi-label feature weighting methods that have recently appeared are presented. In Section 3 the basis of the proposed method and the description of learning process for feature weighting are explained. In Section 4 the experimental set up is described. An

analysis of the experiment results appears in Section 5. Finally, in Section 6, some concluding remarks are presented.

2. Preliminaries

In this section the general background of multi-label learning task, the multi-label lazy algorithms and the feature weighting methods that have appeared in this field are briefly presented.

2.1. Multi-label learning task

The main goal of multi-label learning is to construct a predictive model from examples that are associated with multiple labels at the same time [63]. A multi-label dataset can be defined as follows:

- A set E of N examples (instances), where each instance i is a tuple $< x_i, y_i >$, where x_i is the feature vector and y_i is the set of labels of the instance i (known as relevant labels of the instance i).
- The feature vector of an instance i is a tuple of values of features, $x_i \in F, x_i =< x_{i1}, x_{i2}, \ldots, x_{iD} >$, where F is the feature space, x_{if} represent the value of f-th feature for the instance i and D is the cardinality of the feature space F.
- The set of labels of an instance i is a subset of the label space L, $y_i \subseteq L$. Q is defined as the cardinality of the label space L.

In multi-label learning there exist two types of tasks [37,63]: Multi-Label Classification (MLC) and Label Ranking (LR). MLC deals with learning a model that for a given query instance the set of labels is divided into relevant and irrelevant sets. The binary and multi-class classifications can be considered specific cases of MLC. On the other hand, the LR task is concerned with learning a model for which an ordering of the labels is provided for a given query. The generalisation of these two task has been called multi-label ranking (MLR) in [9]. The goal is to perform the MLC and LR tasks at the same time.

2.2. Supervised methods for multi-label learning

The supervised learning methods for multi-label learning can be grouped into three categories recently proposed in [37]: problem transformation, algorithm adaptation and ensemble methods.

In the problem transformation category the algorithms transform the multi-label problem into one or more single-label problems, and later any classic classification algorithm can be used [63]. The Binary Relevance method (BR) [63] trains a binary classifier for each label. A similar method named Classifier Chain (CC) appears in [45]. Ranking by pair-wise comparison (RPC) trains a binary classifier by each pair of labels [19,22,39,61]. The Label Power Set method (LPS) [61] constructs a new single-label dataset from a multi-label dataset, where each unique set of labels is considered as one of the classes of the new dataset. Several algorithms use the LPS approach, such as the Pruned Problem Transformation method (PPT) [44] and HOMER [62].

The algorithm adaptation category groups those algorithms that can handle the multi-label data directly [63]. In [7] the Predictive Clustering Trees (PCTs) method is proposed. An adaptation of the popular algorithm C4.5 to multi-label learning appears in [13]. Adaptations of the artificial neural networks were proposed in [14,76]. In [51] extensions of AdaBoost approach are proposed. In [3] an algorithm that uses Gene Expression Programming called GEP-MLC is presented. In this category several multi-label lazy algorithms have also been proposed. In the Section 2.2.1 a further description of these methods is given.

In the ensemble category several methods have been proposed. In [31,32,43,45,60] the RFML-C4.5 (Random Forest of ML-C4.5), RF-PCTs (Random Forest of Predictive Clustering Trees), EPS (Ensembles of Pruned Sets), ECC (Ensembles of Classifier Chains) and RAkEL (Random k-Label Sets) methods are presented.

2.2.1. Multi-label lazy learning algorithms

The multi-label lazy algorithms have a high dependency with respect to the definition of a distance function that is used to determine the k-nearest neighbours of a query. To classify a query, the k-nearest neighbours in feature space are retrieved by the distance function, and after that an aggregation strategy is performed to predict the set of labels [63]. As far as similarity-based methods are concerned, it is desirable to find the most similar instances to the query, so that the inference process minimises the amount of incorrect predicted set of labels.

In [77] the ML-kNN algorithm is proposed. In the training phase, the prior probabilities of each label are computed, also the conditional probabilities of each la-

bel are calculated taking into account the information of the label sets of the k-nearest neighbours for each instance from the training set. The authors present a Bayesian multi-label K-nearest neighbour approach, where given a query instance, the ML-KNN algorithm determines the label set using the Maximum A Posteriori (MAP) principle based in the prior and conditional probabilities calculated previously.

In [56] the BR-kNN method was proposed. It determines the k-nearest neighbours of a query and calculates the confidences of each label based on the label sets of the neighbour queries. Furthermore, the authors analyse two extensions of this algorithm, BR-kNN-α and BR-kNN-β, to improve its predictive performance. BR-KNN-α predicts the top ranked label in the case that an empty label set is predicted, whereas the BR-KNN-β extension predicts the top n ranked labels based on size of the label sets of the neighbour instances.

DML-kNN appears in [74]. This algorithm can be considered a generalisation of the ML-kNN algorithm. Like ML-kNN, the DML-kNN algorithm determines the label set of a query instance using the MAP principle, but unlike ML-kNN, the MAP rule defined takes into account the number of all labels in the neighbourhood, meaning the dependencies between labels are considered in a better way.

IBLR-ML [12] combines instance-based learning and logistic regression. It creates a new training data with label information as features, and finally for every label created it trains a logistic regression classifier. This method uses the BR method internally.

In [75] an evidence theory k-NN rule for multi-label learning named EML-kNN is presented. It generalizes Dempster-Shafer's single-label evidence-theory to handle multi-label data. Given a query, the label set is determined on the label sets of the k-neighbours, where according to the extension of the evidence theory presented, each neighbour is considered a piece of evidence. For multi-label datasets that have a high cardinality on the label space, this method is very expensive, because the number of members of the frame of discernment is $2^{|L|}$.

In [36] appears Mr.kNN. This method computes a soft relevance for each instance, in order to determine the grade of membership that an instance belongs to a particular label. To determine the set of labels of a query, a new voting function that takes into account the soft relevance of the k-nearest neighbours for each label was defined. Mr kNN was designed only for numerical attributes; the calculation of the relevance scores, as well as the process of search of the best value for the distance function parameters, is also highly expensive, and therefore the application of this method is difficult, practically speaking, for complex multi-label datasets.

In [72] the MLC-WkNN-EFW algorithm was proposed, which is an instance weighted k-NN version for multi-label learning based on the Bayes Theorem. Given a query, the algorithm gives more impact (a greater weight) to the near instances than to the far ones. To determine the weights for each nearest neighbour, an adaptive estimation process based in a quadratic programming method is performed.

The quality of the lazy algorithms depends on the definition of a distance function that determines the k-nearest neighbours. The distance function is defined in feature space, therefore the irrelevant, interacting, redundant or noise features are used in the same category as other features on distance computation [67]. The main disadvantage of the methods previously described is that they consider any feature equally important for classifying a query, reducing the performance of the multi-label lazy learning algorithms.

2.3. Related works about multi-label feature weighting methods

A good review of feature weighting algorithms on single-label data for a class of lazy learning algorithms is carried out in [67]. The aim is to find a weight vector W to parameterise the distance function with feature weights in order to improve the performance of the lazy algorithms.

Several feature weighting algorithms to multi-label data have been proposed, however, they have been focused on the multi-label feature selection task. Generally, the estimation of the feature weights in multi-label data have been done using the problem transformation methods, such as BR or RPC. For each label; or pair of labels in the case of RPC method, a single-label feature estimation algorithm is executed, and the score of a feature is computed by some aggregation strategy, such as MAX (maximum score of a feature across all labels) or AVG (averaging the scores of a feature across all labels). In this case, any single-label feature estimation algorithm can be used, such as χ^2, gain ratio or ReliefF [70].

In [59] an empirical feature evaluation in order to determine the most relevant features of music into emotions was done. The authors propose to use the LPS approach to transform the multi-label problem into a

multi-class problem, and afterward apply any common single-label feature estimation algorithm.

The most relevant feature weighting methods that have appeared in the context of multi-label data are based on the Relief family of algorithms [29,34,48]. In [55] the BR-ReliefF and LPS-ReliefF methods were proposed, and a comparison with other approach based in the Information Gain measure was done, demonstrating the superiority of the multi-label ReliefF extensions.

The BR-ReliefF uses the BR approach in order to measure the contribution of each feature according to each label. The classic ReliefF is performed for each label and the average of the scores of a feature across all labels is computed. The time complexity of BR-ReliefF method is $O(Q \cdot m \cdot N \cdot D)$, being m the number of instances randomly selected from the training set to estimate the feature weights, N the number of instances of the training set, and Q and D the cardinality of the label and feature space respectively. The efficiency of this method can be affected in datasets with a high cardinality on the label space.

The LP-ReliefF uses the LPS approach in order to measure the contribution of each feature directly from the multi-class problem generated. LPS takes into account implicitly the correlation among labels and often performs better than the BR approach which assumes independence among labels. The LPS method has a high complexity on multi-label datasets which present a big number of distinct label sets, limiting its scalability and commonly tends to over-fitting the data. The time complexity of LP-ReliefF extension is $O(m \cdot N \cdot D)$.

The MReliefF method was presented in [33]. The multi-label problem is decomposed into a set of pairwise multi-label 2-class problems, being equivalent to decompose the multi-label problem into several binary classification problems using the RPC approach as problem transformation method. The time complexity of MReliefF is $O(Q^2 \cdot m \cdot N \cdot D)$, therefore this algorithm is very expensive in datasets that have a big number of labels.

The previously described methods have proved their usefulness over the multi-label feature selection process. There is still not a comprehensive treatment in the literature, however, of how to make feature weighting on multi-label data to improve the performance of multi-label lazy algorithms. The lack in the literature is bigger when is referred to make feature weighting in multi-label data using evolutionary algorithms.

To the best of our knowledge, the first attempt to introduce an evolutionary strategy to estimate the feature weights in multi-label data to improve the performance of the multi-label lazy algorithms appeared in [46]. The search process of the best weight vector was performed using a genetic algorithm (GA), with a real codification, following a generational elitism strategy. For each selected instance two orders are calculated: a feature-based instances order and a label-based instances order. The fitness function of the GA algorithm is a function that calculates the distance among the orders generated over all sampling instances. This approach has been proved over the ML-kNN, BR-kNN and IBLR-ML lazy methods, showing its effectiveness to improve the performance of the multi-label lazy algorithms on 6 standard multi-label datasets. However, this method has several disadvantages, some of which were previously mentioned in the introduction to the present work.

A subset of m instances from the training set is extracted to compute the weight vector. Steps are needed for the calculus of the feature-based instances order from a sampling instance $i, O(N \cdot D)$, owing to the distance between i and all the instances of the training set being computed. Steps are needed for the calculus of the order of label-based instances from a sampling instance $i, O(N \cdot P)$, because the cardinality of the order generated is equal to N, P being the sum of the cardinalities of the feature and label space. Therefore, the time complexity of the fitness function used by the GA algorithm is $O(m \cdot N \cdot P)$. On the other hand, the GA algorithm requires $O(g \cdot c)$ steps, being g the number of generations and c the number of individuals that compose the population. In general the time complexity of the multi-label feature weighting method introduced in [46] is $O(g \cdot c \cdot m \cdot N \cdot P)$. In the following, we referred to this method as GFW.

The datasets used in the multi-label learning field are characterised by a high dimension and a large number of instances. In complex datasets the GFW algorithm needs more generations to converge into an optimal solution. Therefore, the GFW method is very expensive in complex multi-label datasets. In the next section a new filter-based evolutionary feature weighting method to improve the performance of multi-label lazy algorithms is proposed.

3. Evolutionary multi-label feature weighting

In this section the basis of the new multi-label feature weighting method proposed is explained.

The distance d_F between the descriptions of two instances is calculated by a weighted version of the

HEOM distance (Heterogeneous Euclidean Overlap Metric) [69] (see Eq. (1)). Given an instance i with a description x_i, and an instance j with a description x_j, the expression Eq. (1) quantifies the distance between the instances i and j, given the features and their weights. This is an "a priori" measure: a measurement of the usefulness of a description j to classify the instance i, without introducing the set of labels of j. It assumes the vector W and tuples of $F \in [0 \dots 1]$ for all features, where W_f represents the weight for the f-th feature.

$$d_F(i,j) = \sqrt{\sum_{\forall f \in F} W_f \cdot \delta(x_{if}, x_{jf})^2} \tag{1}$$

$$\delta(x_{if}, x_{jf}) = \begin{cases} 1 & \textit{discrete}, x_{if} \neq x_{jf} \\ 0 & \textit{discrete}, x_{if} = x_{jf} \\ |x_{if} - x_{jf}| & \textit{continuous} \end{cases} \tag{2}$$

Given an instance i with a set of labels y_i and an instance j with a set of labels y_j, the distance between the sets of labels of i and j can be computed by the function d_L (see Eq. (3)), where Δ is the symmetric difference between the two sets of labels.

The distance d_L is the Hamming Distance, and represents a measure of how much the sets of labels of two instances differ, so that a smaller value of d_L represents a major similarity in the classification of these instances. The distance d_L between two instances is an "a posteriori" criterion, since the set of labels associated with a query instance is not usually known.

$$d_L(i,j) = y_i \Delta y_j \tag{3}$$

Given the formulated distances, two groups of k-nearest neighbours with respect to a sampling instance i can be defined: feature based k-nearest neighbours $K_{d_F}(i)$ and label based k-nearest neighbours $K_{d_L}(i)$.

$$\begin{aligned} K_{d_F}(i) &= (i_{F_1}, \dots, i_{F_k}) | \forall j : d_F(i, i_{F_j}) \\ &\leqslant d_F(i, i_{F_{j+1}}) \end{aligned} \tag{4}$$

$$\begin{aligned} K_{d_L}(i) &= (i_{L_1}, \dots, i_{L_k}) | \forall j : d_L(i, i_{L_j}) \\ &\leqslant d_L(i, i_{L_{j+1}}), \end{aligned} \tag{5}$$

where i_{F_j} and i_{L_j} are the j-th nearest neighbours in the feature space and label space respectively. The feature-based k-nearest neighbours represent the nearest neighbours of a sampling instance i given the features and their weights, while the label based k-nearest neighbours represent those instances that have the closest sets of labels. The $K_{d_L}(i)$ group corresponds to the ideal k-nearest neighbours for an instance i searched by similarity based methods.

Through the modification of the weight vector W the distance function d_F can obtain a finite number of variants of feature based k-nearest neighbours with respect to an instance i. A measurement of the imperfection of the $K_{d_F}(i)$ group generated by d_F for a sampling instance i can be measured by the expression Eq. (6). This metric quantifies the distance that exists between the groups of neighbours $K_{d_F}(i)$ and $K_{d_L}(i)$ with respect to an instance i.

$$\begin{aligned} &F(K_{d_F}(i), K_{d_L}(i)) \\ &= \sum_{j=1}^{k} \frac{1}{2^j} \frac{|d_L(i, i_{L_j}) - d_L(i, i_{F_j})|}{1 + |d_L(i, i_{L_j}) - d_L(i, i_{F_j})|} \end{aligned} \tag{6}$$

The operator $|Z|$ represents the absolute value of the expression Z. The expression Eq. (6) satisfies the non-negativity, reflexivity, symmetry and triangle inequality conditions. This metric takes into account the possible ties that can appear among some components of the groups of k-nearest neighbours, also only the k-nearest neighbours are considered, in contrast to the metric used in [46]. It is an extension of the Fréchet's permutation metric [16] that considers the Hamming distance between pairs of neighbours located in the same position but in different groups.

The weight vector used by the function d_F generates for an instance i the group of k-nearest neighbours $K_{d_F}(i)$, with an ordering of the neighbours by its weighted distances with respect to i. These neighbours may, however, be disordered depending on the Hamming distance between the set of labels of each neighbour and the instance i. The greater difference among the sets of labels of pairs of neighbours located in the same position but in different groups, the greater the penalisation that performs the metric to the weight vector used by d_F.

Therefore, the problem is to find a weight vector W that minimises the following expression:

$$E_W = \sum_{i=1}^{m} F(K_{d_F}(i), K_{d_L}(i)), \tag{7}$$

where m is the number of selected instances from the training set.

The solution to the previously stated problem is to minimise the disorder caused by $F(K_{d_F}(i), K_{d_L}(i))$ over the entire set of sampling instances. The ratio-

nale of this proposal is straightforward: the minimisation of E_W implies the minimisation of the differences between the two k-nearest neighbours groups generated by the functions Eqs (1) and (3). Therefore, one can presume that, in minimum conditions, the generated groups will be similar enough in respect of the sets of labels. The greater similarity among the groups of neighbours for all instances of the set of sampling instances, the greater the likelihood of recovering those nearest instances in the feature space with the major confidence set of labels for classification.

In [67] a framework to classify the feature weighting methods used for single-label data is proposed. However, this framework can be used to classify feature weighting methods on multi-label data too. According to the framework proposed in [67], the feature weighting method proposed in this paper is a method that does not use the feedback from the classifier to assign the weights to the features. Instead, it learns a metric to find the best weight vector; it is a filter-based method. It uses the given representation of the original multi-label data, it learns a single set of weights that are employed globally over the entire instance space and does not employ domain specific knowledge to set feature weights.

3.1. Learning process for feature weighting

In the last years, the metaheuristic methods, such as genetic algorithms, genetic programming, particle swarm and ant colony optimization, have been widely used in the solution of complex problems [4,10,11,21, 26,40–42,50,54,58].

In this work, the search process of the best weight vector was performed using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [2,24]. CMA-ES is a powerful heuristic method to real optimisation problems and it is characterised by its good exploration of search space.

Each solution point is represented as a vector whose dimension (D) is the number of features of the multi-label dataset. A solution point represents a possible weight vector to be used posteriorly by the lazy learning algorithm. The allowed range of values for point dimensions (genes) was [0...1]. Each gene represents the weight of a feature. The fitness function of each solution point was calculated by the expression Eq. (7). CMA-ES has an efficient parameter tuning process as part of the algorithm design, therefore it does not require an expensive search of good parameter values. Also, CMA-ES has a learning rates that prevent degeneration even for small population sizes.

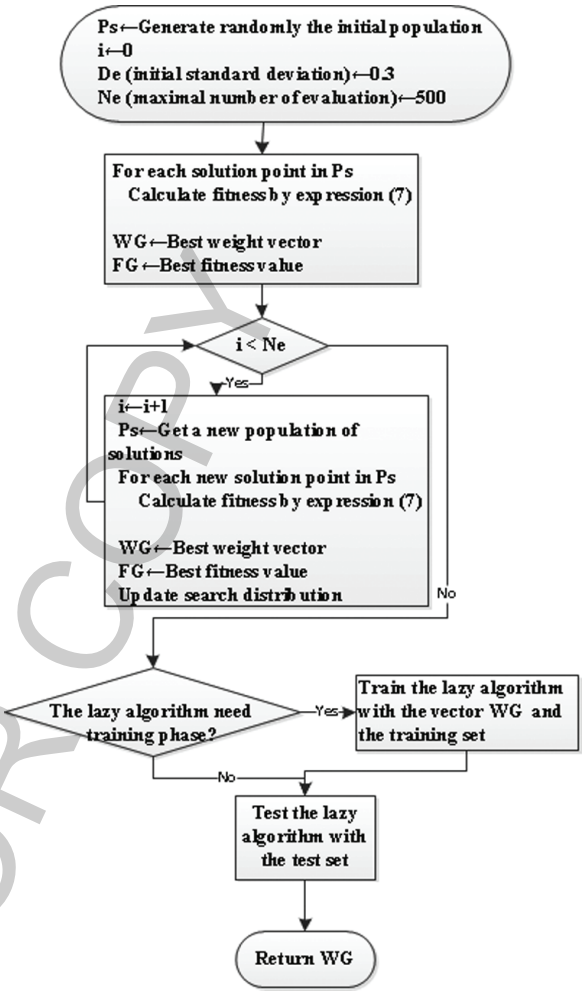


Fig. 1. Flowchart of the learning process for feature weighting.

In the training phase the CMA-ES finds the weight vector that achieves a better fitness value. Once the CMA-ES ends, if is necessary, the lazy learning algorithm is trained using the best weight vector founded. In the test phase, the set of labels for each test instance using the best weight vector founded by CMA-ES is predicted. Figure 1 shows the flowchart of the learning process for feature weighting.

Finding the feature-based k-nearest neighbours and label-based k-nearest neighbours of a sampling instance i can be computed efficiently in $O(\log N)$ steps. To speed up the task of determine the k-nearest neighbours of a query point, several different search algorithms can be used, such as k D-Tree, Ball-Tree or Cover-Tree [70].

Therefore, the time complexity of the expression used as fitness function in the CMA-ES algorithm is $O(m \cdot \log N)$. On the other hand, the CMA-ES algo-

rithm used requires $O(N_e \cdot N_s)$ steps, being N_e the number of iterations and N_s the number of solution points that compose the population. In general, the time complexity of the multi-label feature weighting method proposed in this work is $O(N_e \cdot N_s \cdot m \cdot \log N)$. In the following, we referred to this method as EFW.

From a computational view, the time complexity of the EFW method is lower than the method presented in [46]. CMA-ES algorithm requires a lower number of iterations and solution points than the GA algorithm to converge into an optimal solution. On the other hand, the complexity of the fitness function defined in this work is much lower than the complexity of the fitness function defined in [46]. Also, the time complexity of the EFW method does not depend of the number of labels Q of a multi-label dataset, unlike to BR-ReliefF and MReliefF methods described in the sub-Section 2.3.

4. Experimental section

In this work, the methods ML-kNN, the standard version of BR-kNN, DML-kNN, IBLR-ML and MLC-WkNN were selected to examine the benefits of feature weighting methods to improve the performance of multi-label lazy algorithms. The exposed limitations of the Mr.kNN and EML-kNN methods were not included in the experimentation.

The experiments were divided into 3 parts. In the first part, to prove the effectiveness of our proposal (EFW), the original multi-label lazy methods with the weighted multi-label lazy methods using the EFW algorithm were compared. In addition, to determine the multi-label lazy method that was most benefit by the EFW algorithm, a comparison among the weighted lazy methods was done. In the second part, to determine the evolutionary feature weighting method that reaches the better performance over the multi-label lazy algorithms, the GFW and EFW methods were compared. In the third part, to compare the effectiveness of the EFW algorithm with respect to the most relevant multi-label feature estimation algorithms, a multiple comparison among EWF, BR-ReliefF, LPS-ReliefF and MReliefF algorithms was done.

The algorithms were implemented on MULAN [64]. MULAN is a Java library built on the data mining tool WEKA [70] that contains several methods and evaluation measures for multi-label learning.

For the execution of EFW method, the optimal values for CMA-ES recommended in [2,24] are employed. CMA-ES has as stopping criterions the maximum number evaluations of the objective function (N_e) equal to 500, and a drop of the difference of the fitness function values below $1E{-}13$. The initial standard deviation (D_e) was set to 0.3. The number of restarts is two, the population size is not increased, and the initial population is randomly generated. The population size is adapted according to the number of features of the multi-label dataset, and it is set to 4log D^2, where D is the cardinality of the feature space.

IF ($N \leq 5000$) **THEN** $m = 0.1 \cdot N$

IF ($N > 5000$ AND $N \leq 10000$) **THEN** $m = 0.05 \cdot N$

IF ($N > 10000$) **THEN** $m = 0.01 \cdot N$

Fig. 2. Rules to fix the number of instances (m) to be selected from the training set.

With similar settings, the feature weighting method GFW was tested. The other parameters, such as replacement strategy, crossover and mutation probabilities, etc., are the same as proposed in [46].

The BR-ReliefF, LPS-ReliefF and MReliefF feature weighting methods return the weights in the range $[-1 \ldots 1]$, therefore the weights are scaled in $[0 \ldots 1]$ range to be used as feature weights in the distance function of the multi-label lazy algorithms.

The best number of neighbours (k) for each algorithm in each dataset was determined. The performance of the EFW, GFW, BR-ReliefF, LPS-ReliefF and MReliefF feature weighting methods depend of the number of instances (m) to be selected from the training set for the learning process of the weight vector W. For computational reasons, the value of m should be much less than the number of the instances (N) of the training set. According to the multi-label datasets used in this research, the rules that appear in the Fig. 2 have been set by means of a study testing different values and selecting the most appropriate.

A subset of m instances from the training set is extracted to compute the weight vector. Because of this, in datasets where the label space has a high cardinality, there will be a large number of distinct label subsets. Picking a predefined number of instances randomly from the training set does not guarantee that a representative sample of all label subsets is selected, i.e., in the resulting set of picking instances randomly there will be patterns of label subsets that are poorly represented.

In order to attenuate this situation, 2 sampling methods for multi-label data proposed in [52] were used. These 2 methods were called Labelset-Stratification

Table 1
Statistics of the benchmark datasets, number of instances (N), number of features (D), number of labels (Q), different label subsets (S_L), label cardinality (L_{Car}) and label density (L_{De})

Dataset	N	D	Q	S_L	L_{Car}	L_{De}
Emotions	593	72	6	27	1.869	0.311
Yeast	2417	103	14	198	4.237	0.303
Scene	2407	294	6	15	1.074	0.179
Cal500	502	68	174	502	26.044	0.150
Genbase	662	1186	27	32	1.252	0.046
Medical	978	1449	45	94	1.245	0.028
Enron	1702	1001	53	753	3.378	0.064
TMC2007-500	28596	500	22	1341	2.160	0.098
Mediamill	43907	120	101	6555	4.376	0.043
Corel5k	5000	499	374	3175	3.522	0.009
Corel16k	13811	500	161	4937	2.867	0.018
Bibtex	7395	1836	159	2856	2.402	0.015

and Iterative-Stratification, and the goal is to stratify the multi-label datasets. For datasets where the ratio of distinct label sets over the number of examples is small ($\leqslant 0.1$) the Labelset-Stratification method was used to select the m instances from the training set, whereas for datasets where the ratio of the distinct label sets over the number of examples is large, the Iterative-Stratification method was used as proposed in [52].

For each possible combination of algorithms and datasets a stratified 10-fold cross validation strategy was used. To stratify the multi-label data, the methods proposed in [52] were used. For each fold a training and test set is constructed, m instances from the training set are selected and a weight vector W is learned. After that, the lazy algorithm is tested with the test set. The average of the test results for each fold was calculated. Owing to the random nature of the feature weighting algorithms considered, for each experiment 4 different runs were executed and the average value was calculated.

The results from the experiments were analysed using the non-parametrical statistical tests as proposed in [15] in order to determine whether or not the results of the study are statistically significant. For the comparison between two methods, the Wilcoxon signed-ranks test [68] was used. For the multiple comparison between all feature weighting methods, the Friedman [20] test was used, and if the null hypothesis is rejected, then the Bergmann-Hommel post-hoc test [6] is used as proposed in [23] for all pairwise comparisons.

4.1. Datasets

Multi-label datasets with different scales and from different application domains were included, in order to understand the behaviour of the feature weighting method proposed in datasets with diverse characteristics.

The datasets come from three domains: biology, multimedia and text categorisation. The Yeast [19] and Genbase [17] datasets come from the biological domain, and include information about the function of genes and proteins, respectively. The datasets Cal500, Emotions, Scene, Mediamill, Corel5k and Corel16k (10 samples) belong to the multimedia domain. Cal500 [65] contains pieces of music for semantic annotation and retrieval of music and sound effects. Emotions dataset [59] stores the classification of songs according to the emotions they evoke. Scene dataset [8] contains a series of patterns about kinds of landscapes. The Mediamill dataset [71] contains information about annotated videos. The Corel5k [18] and Corel16k [5] (10 samples) datasets contain Corel images. In the domain of text categorisation four datasets were selected: Medical, Enron, TMC2007 and Bibtex. The data in Medical [45] was used in the Medical Natural Language Processing Challenge in 2007. The Enron [30] dataset contains emails from 151 users distributed in 3500 folders. TMC2007 [57] stores examples of aviation safety reports. A reduced version of this dataset with the top 500 features selected was used, same as [60]. Bibtex [27] contains information for bibtex items and is used for automatic tag suggestion.

Table 1 shows some statistics from the datasets used. The label cardinality is the average number of labels per example and label density is the same number divided by the total number of labels. The label cardinality, label density and different subsets of labels are measures that represent the complexity of a multi-label dataset [63]. The values of the properties of the Corel16k dataset that appear in Table 1 are the average over all 10 samples used.

The datasets selected have different characteristics. The variation ranges are from 502 up to 43, 907 training examples, from 68 up to 1836 features, from 6 to 374 labels, from 15 to 6555 different subsets of labels, from 1.074 to 26.044 label cardinality, and from 0.009 to 0.311 label density.

4.2. Evaluation measures to multi-label learning

In order to verify the effectiveness of the proposal, the following evaluation measures that have been suggested for MLC and LR tasks in [63] were used:

4.2.1. Bipartition-based

In the Example-based subgroup, the Hamming loss (H_L), F1-Measure (F_{1Ex}) and Accuracy (A_{cc}) were

chosen. The H_L measure averages the symmetrical differences among the predicted and actual label sets; F_{lEx} is the harmonic mean of precision and recall; and A_{cc} averages the proportion of the predicted correct labels to the total number (predicted and actual) of labels. In the Label-based subgroup, the Micro-F1 measure (F_{lMi}), which calculates the micro-average of harmonic mean of precision and recall, was used [37,63].

4.2.2. Ranking-based

The One Error measure (O_E) averages how many times the top ranked predicted label is not in the set of true labels. The Ranking Loss measure (R_L) averages the proportion of label pairs that are incorrectly ordered. The Average Precision (A_P) measure averages how many times a particular label is ranked above other label which is actually in the truth label set [37,63].

The H_L, F_{lEx}, A_{cc} and F_{lMi} evaluation measures are associated with the MLC task, whereas O_E, R_L and A_P are associated with the LR task. The higher the value of F_{lEx}, A_{cc}, F_{lMi} and A_P, and the lower the value of H_L, O_E and R_L, the better the performance of a multi-label learning algorithm.

5. Results and discussion

The algorithms as standalone runnable files and all the results of the experimental study are available to facilitate the replicability of the experiments.[1] In this manuscript, due to lack of space, only a summary of the results appear.

The results show that the weighted versions of the lazy methods using the EFW algorithm perform better than the original lazy methods (not weighted) in the Emotions, Yeast, Scene, Cal500, Genbase, Medical, Enron, TMC2007-500, Mediamill and Bibtex datasets with the 7 measures used in the experiment. For other complex datasets, such as Corel5k and the 10 samples of Corel16k, the results vary according to the algorithm and the measure employed.

All the results, for each feature weighting algorithm, lazy method, multi-label dataset and measure considered can be consulted in the available web site.

The Tables 3 and 4 show the results of Wilcoxon signed-ranks test used to determine whether or not the results of the study are statistically significant. In the case of refusing the null hypothesis with a significance level $\alpha = 0.05$, the p-value is highlighted in bold typeface. The evidence suggests that the weighted versions ML-kNN-EWF, DML-kNN-EWF and MLC-WkNN-EFW are statistically superior to the original lazy methods in all the bipartition-based and ranking-based measures. The BR-kNN-EWF is statistically better than the original algorithm in all the bipartitions-based and ranking-based measures, except in R_L that Wilcoxon's test does not detect significant differences. The IBLR-ML-EWF is statistically better than the original lazy algorithm in all the bipartitions-based and ranking-based measures, except in F_{lMi} and O_E measures that Wilcoxon's test does not detect significant differences.

Table 2
Comparison among all multi-label lazy classifiers using the EFW method. Each column represents the ranking and the last row contains the p-value returned by Friedman's test

Methods	H_L	F_{1Ex}	A_{cc}	F_{1Mi}	O_E	R_L	A_P
ML-kNN-EFW	*2.357*	3.548	3.476	3.452	2.571	2.190	2.476
BR-kNN-EFW	3.571	3.809	3.809	3.833	4.714	3.976	4.762
DML-kNN-EFW	2.833	4.357	4.500	4.333	*2.095*	*1.905*	2.405
IBLR-ML-EFW	3.548	2.071	2.000	2.167	2.309	2.500	*1.976*
MLC-WkNN-EFW	2.691	*1.214*	*1.210*	*1.214*	3.309	4.429	3.381
Friedman test				p-value			
	0.04	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table 2 shows the results of Friedman's test comparing the weighted lazy method versions using the EFW algorithm to determine what the best weighted version for each measure is. According to the results, the weighted version ML-kNN-EWF obtains the better performance for the H_L measure, the weighted version MLC-WkNN-EFW obtains better results for the F_{lEx}, A_{cc} and F_{1Mi} measures, the weighted version DML-kNN-EWF obtains the better performance for the O_E and R_L ranking-based measures, whereas the IBLR-ML-EWF is better for the A_P measure. In the case of BR-kNN lazy method the advantages of the weighting features process EFW are lower compared to the other selected lazy algorithms.

In this paper, the statistical information obtained from the Bergmann-Hommel post-hoc test is graphically illustrated as a graph, where each method is represented as one vertex of the graph. An edge $\Phi_1 \rightarrow \Phi_2$ represents that the method Φ_1 outperforms the method Φ_2. Each edge is labeled with the measures that Φ_1 outperforms the Φ_2 method, and the adjusted p-value of the Bergamnn-Hommel post-hoc test is indicated between parentheses.

The Fig. 3 shows the results of Bergmann-Hommel post-hoc test comparing the weighted lazy methods

[1] http://www.uco. es/grupos/kdis/kdiswiki/ML-FW.

Table 3
Wilcoxon signed-ranks test. Comparison of the original multi-label lazy classifiers and the weighted versions using the EFW method. The table summarizes the positive ranks (R^+), negative ranks (R^-) and p-values

Measure	ML-kNN-EFW vs ML-kNN			DML-kNN-EFW vs DML-kNN			BR-kNN-EFW vs BR-kNN		
	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value
H_L	192.0	39.0	**0.01**	171.0	39.00	**0.01**	178.5	52.5	**0.03**
O_E	231.0	0.0	**0.00**	231.0	0.0	**0.00**	185.5	24.5	**0.00**
R_L	229.5	1.5	**0.00**	231.0	0.0	**0.00**	112.0	98.0	0.77
F_{lEx}	216.5	14.5	**0.00**	219.0	12.0	**0.00**	215.0	16.0	**0.00**
A_{cc}	199.0	11.0	**0.00**	216.5	14.5	**0.00**	194.5	15.5	**0.00**
F_{lMi}	200.0	10.0	**0.00**	223.5	7.5	**0.00**	198.5	11.5	**0.00**
A_P	226.5	4.5	**0.00**	21.0	0.0	**0.00**	180.0	30.0	**0.00**

Table 4
Wilcoxon signed-ranks test. Comparison of the original multi-label lazy classifiers and the weighted versions using the EFW method

Measure	IBLR-ML-EFW vs IBLR-ML			MLC-WkNN-EFW vs MLC-WkNN		
	R^+	R^-	p-value	R^+	R^-	p-value
H_L	157.5	52.5	**0.047**	192.0	39.0	**0.01**
O_E	134.0	76.0	0.24	229.5	1.5	**0.00**
R_L	166.5	43.5	**0.013**	229.5	1.5	**0.00**
F_{1Ex}	181.0	29.0	**0.00**	185.0	46.0	**0.00**
A_{cc}	190.5	40.5	**0.01**	176.5	33.5	**0.00**
F_{lMi}	160.5	70.5	0.10	190.0	41.0	**0.01**
A_P	160.5	49.5	**0.03**	228.5	2.5	**0.00**

using the EFW algorithm. The MLC-WkNN-EWF method outperforms the BR-kNN-EWF method in the A_{cc}, F_{lEx}, F_{lMi}, A_P and O_E measures; also, it is statistically better than ML-kNN-EWF and DML-kNN-EWF methods in the A_{cc}, F_{lEx} and F_{lMi} measures. IBLR-ML-EFW outperforms the BR-kNN-EWF method, excepting H_L, in all the measures, besides it is statistically better than ML-kNN-EWF and DML-kNN-EWF methods in the A_{cc}, F_{lEx} and F_{lMi} measures. The ML-kNN-EWF and DML-kNN-EWF methods are statistically better than BR-kNN-EWF in all the ranking-based measures. The ML-kNN-EWF, DML-kNN-EWF and IBLR-ML-EFW methods outperform the MLC-WkNN-EWF method in the R_L measure; furthermore, the IBLR-ML-EFW obtains better results than MLC-WkNN-EWF method in the A_P measure. The evidences suggest that for bipartition-based measures, the MLC-WkNN-EWF and IBLR-ML-EWF methods obtain the better results, yet for the ranking-based measure are the ML-kNN-EWF, DML-kNN-EWF and IBLR-ML-EWF methods.

A comparison among the EFW and GFW methods was carried out. The comparison was done on 7 simple multi-label datasets because the GFW method is very expensive when used with complex datasets. The results show that the approach proposed in this work (EFW) outperforms the GFW method on the 7 evaluation measures considered. Tables 5 and 6 show the results of Wilcoxon's test comparing the two methods on the 5 lazy algorithms. In the case of refusing the null hypothesis with a significance level $\alpha = 0.05$ the p-value is highlighted in bold typeface. According to the results, the method EFW obtains better results than GFW in all the evaluation measures, except in the F_{lMi} measure over the DML-kNN lazy method where Wilcoxon's test does not detect significant differences for the significance level considered.

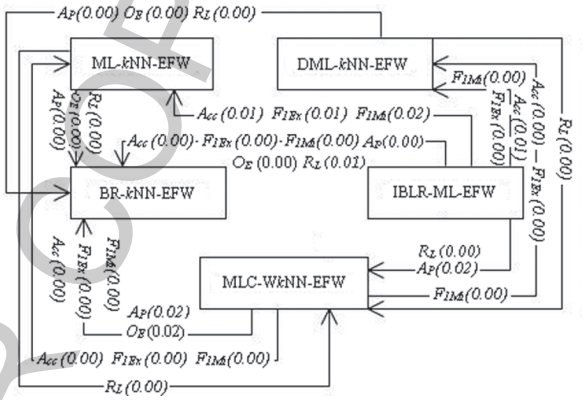


Fig. 3. Significant differences of the performance among all multi-label lazy classifiers using the EFW method.

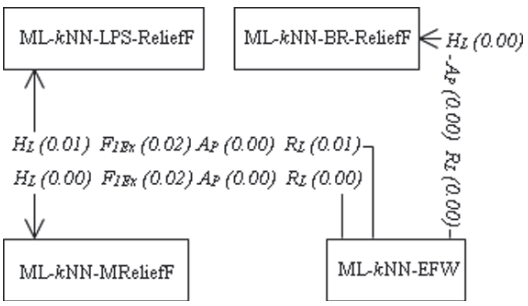


Fig. 4. Significant differences of the performance among multi-label feature weighting methods on the ML-kNN classifier.

Table 5
Wilcoxon signed-ranks test. Comparison of the multi-label lazy classifiers using the EFW and GFW methods

Measure	ML-kNN-EFW vs ML-kNN-GFW			DML-kNN-EFW vs DML-kNN-GFW			BR-kNN-EFW vs BR-kNN-GFW		
	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value
H_L	21.0	0.0	**0.02**	21.0	0.0	**0.02**	21.0	0.0	**0.02**
O_E	28.0	0.0	**0.01**	28.0	0.0	**0.01**	28.0	0.0	**0.01**
R_L	21.0	0.0	**0.02**	19.5	1.5	**0.046**	27.0	1.0	**0.02**
F_{1Ex}	28.0	0.0	**0.01**	28.0	0.0	**0.01**	21.0	0.0	**0.02**
A_{cc}	28.0	0.0	**0.01**	21.0	0.0	**0.02**	27.0	1.0	**0.02**
F_{lMi}	28.0	0.0	**0.01**	25.0	3.0	0.051	28.0	0.0	**0.01**
A_P	21.0	0.0	**0.02**	28.0	0.0	**0.01**	28.0	0.0	**0.01**

Table 6
Wilcoxon signed-ranks test. Comparison of the multi-label lazy classifiers using the EFW and GFW methods

MEASURE	IBLR-ML-EFW vs IBLR-ML-GFW			MLC-WkNN-EFW vs MLC-WkNN-GFW		
	R^+	R^-	p-value	R^+	R^-	p-value
H_L	19.0	2.0	**0.03**	21.0	0.0	**0.01**
O_E	21.0	0.0	**0.02**	28.0	0.0	**0.01**
R_L	26.5	1.5	**0.03**	21.0	0.0	**0.02**
F_{1Ex}	28.0	0.0	**0.01**	20.0	1.0	**0.03**
A_{cc}	28.0	0.0	**0.01**	21.0	0.0	**0.02**
F_{lMi}	28.0	0.0	**0.01**	21.0	0.0	**0.02**
A_P	28.0	0.0	**0.01**	28.0	0.0	**0.01**

Table 7
Comparison among multi-label feature weighting methods on the ML-kNN and DML-kNN classifiers

Methods	H_L	F_{lEx}	R_L	A_P	Methods	H_L	F_{lEx}	R_L	A_P
ML-kNN-EFW	1.167	1.786	*1.381*	*1.452*	DML-kNN-EFW	*1.381*	*1.238*	*1.357*	*1.429*
ML-kNN-BR-ReliefF	3.143	2.405	2.905	3.119	DML-kNN-BR-ReliefF	2.857	3.071	3.143	2.762
ML-kNN-LPS-ReliefF	2.548	2.881	2.571	2.571	DML-kNN-LPS-ReliefF	2.762	2.786	2.500	2.857
ML-kNN-MReliefF	3.143	2.929	3.143	2.857	DML-kNN-MReliefF	3.000	2.905	3.000	2.952
Friedman test	p-value				Friedman test	p-value			
	0.00	**0.01**	**0.00**	**0.00**		**0.00**	**0.00**	**0.00**	**0.00**

Table 8
Comparison among multi-label feature weighting methods on the BR-kNN and MLC-WkNN classifiers

Methods	H_L	F_{lEx}	R_L	A_P	Methods	H_L	F_{lEx}	R_L	A_P
BR-kNN-EFW	*1.286*	*1.405*	*1.405*	*1.524*	MLC-WkNN-EFW	*1.333*	*1.357*	*1.429*	*1.333*
BR-kNN-BR-ReliefF	2.976	3.238	2.500	2.643	MLC-WkNN-BR-ReliefF	3.167	2.809	3.071	2.762
BR-kNN-LPS-ReliefF	2.881	2.524	2.976	2.762	MLC-WkNN-LPS-ReliefF	2.786	2.905	2.309	2.929
BR-kNN-MReliefF	2.857	2.833	3.119	3.071	MLC-WkNN-MReliefF	2.714	2.929	3.190	2.976
Friedman test	p-value				Friedman test	p-value			
	0.00	**0.00**	**0.00**	**0.00**		**0.00**	**0.00**	**0.00**	**0.00**

A comparison among EFW, BR-ReliefF, LPS-ReliefF and MReliefF feature weighting methods over the multi-label lazy algorithms ML-kNN, DML-kNN, BR-kNN and MLC-WkNN was carried out. Tables 7 and 8 show the results of Friedman's test comparing the 4 lazy methods using the 4 feature weighting algorithms, considering only the H_L, F_{lEx}, R_L and A_P measures. All the results of this part of the experimental study can be consulted in the available web site. According to the Friedman' test results, the weighted version ML-kNN-EFW obtains the better performance for all the measures considered. The same situation occurs for the DML-kNN-EFW, BR-kNN-EFW and MLC-W kNN-EFW weighted versions.

Figures 4–7 show the results of the Bergmann-Hommel post-hoc test comparing the 4 feature weighting methods on the ML-kNN, DML-kNN, BR-kNN and MLC-WkNN-EFW classifiers respectively. The ML-kNN-EFW method outperforms the ML-kNN-LPS-ReliefF and ML-kNN-MReliefF methods in all the measures considered, furthermore it is statistically better than ML-kNN-BR-ReliefF metod in the H_L,

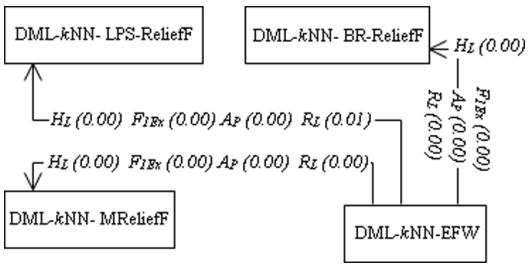


Fig. 5. Significant differences of the performance among multi-label feature weighting methods on the DML-kNN classifier.

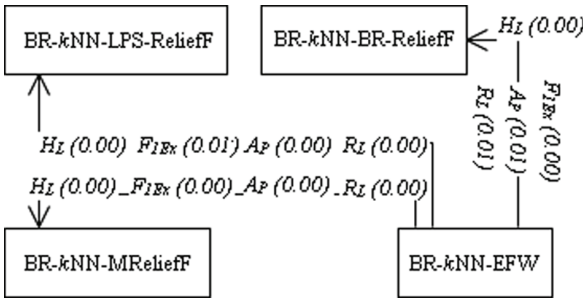


Fig. 6. Significant differences of the performance among multi-label feature weighting methods on the BR-kNN classifier.

R_L and A_P measures. The DML-kNN-EFW method outperforms the DML-kNN-LPS-ReliefF, DML-kNN-MReliefF and DML-kNN-BR-ReliefF methods in all the measures considered. The BR-kNN-EFW method is statistically better than BR-kNN-LPS-ReliefF, BR-kNN-MReliefF and BR-kNN-BR-ReliefF methods in all the measures considered. The MLC-WkNN-EFW method outperforms the MLC-WkNN-BR-ReliefF and MLC-WkNN-MReliefF methods in all the measures considered, furthermore it is statistically better than MLC-WkNN-LPS-ReliefF in the H_L, F_{1Ex} and A_P measures. The results show that the EFW method is competitive with respect to the most relevant multi-label feature weighting algorithms proposed in the literature.

In general, the feature weighting method (EFW) proposed in this work to improve the performance of the multi-label lazy algorithms performs well for the two task analysed, MLC and LR tasks. The weight vector learned in the process allows that the distance function recover those nearest examples in the feature space that are associated with the major confidence set of labels for classifying a query instance. The evidences suggest that for simple multi-label datasets, e.g. Emotions, Genbase, Cal500, Medical, Scene and Enron datasets, the approach proposed obtains similar results in both task MLC and LR. However, in complex

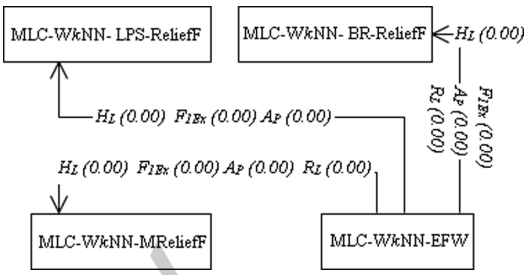


Fig. 7. Significant differences of the performance among multi-label feature weighting methods on the MLC-WkNN classifier.

multi-label datasets, such as the Corel5k and Corel16k datasets, the results for ranking-based measures are a little better than for bipartition-based measures. In this sense, we hypothesise that this situation is caused since the ranking-based measures used in this work are less sensitive to the cardinality of the label space.

Moreover, the feature weighting method proposed performs well in simple and complex multi-label datasets. The results, however, show that there is a smaller performance increase in those multi-label datasets with a small label density and a big number of distinct label sets at the same time, e.g. the Corel5k and Corel16k datasets. The determination of the label based k-nearest neighbours of a sampling instance is sensitive with respect to the label density and the number of distinct label sets of the multi-label dataset.

6. Conclusions

In this paper a new filter method for multi-label feature weighting to improve the performance of the multi-label lazy algorithms was presented. For each sampling instance, two groups of k-nearest neighbours were determined and a defined metric calculated the distance between the groups. The searching process of the best weight vector was carried out by CMA-ES in order to determine a weight vector W that minimises the distance among the k-nearest neighbours groups over all sampling instances, as a heuristic to estimate the feature weights. The feature weighting method does not use the feedback from the classifier to assign the weights to the features. This method uses the given representation of the original multi-label data and learns a single set of weights that are employed globally over the entire instance space and does not employ domain specific knowledge to set feature weights.

The experimental results on 21 multi-label datasets and 5 multi-label lazy algorithms confirm the effec-

tiveness of the feature weighting method proposed for a better multi-label lazy learning. The evidence suggests that for simple multi-label datasets, the approach obtains similar results in both task MLC and LR, yet for more complex multi-label datasets the results for ranking-based measures are a little better than those for bipartition-based measures. On the other hand, the results show that there is a smaller performance increase in those multi-label datasets with a small label density and a large number of distinct label sets at the same time.

The results obtained show that our approach is robust; it does well in multi-label datasets with different properties. Furthermore, the proposed method improves the performance of multi-label lazy learning algorithms to retrieve the neighbours that have the most relevant set of labels for classifying a query instance, which is consistent with the reasoning exposed in Section 3. Moreover, the steps described in Section 3 can be used to extend any multi-label lazy algorithm.

The feature weighting approach described in this paper outperforms the evolutionary feature weighting algorithm GFW, both statistically and from the point of view of time complexity. Furthermore, the results show that the proposed method is competitive with respect to the most relevant multi-label feature estimation algorithms that have been proposed in the literature.

The experimental study confirms that the method proposed has significant advantages to improve the performance of the multi-label lazy algorithms in MLC and LR tasks, the main motivation for the present work.

Future research will study the use of other metrics learning to improve the performance of multi-label lazy methods. On the other hand, it would be interesting to analyse the effectiveness of the approach in the multi-label feature selection area, an area in which, in contrast to single-label learning, far less research has been done. Furthermore, it would be important to do a comparative study of how the multi-label lazy algorithms using the feature weighting methods scale to other state-of-the-art algorithms for multi-label learning.

Acknowledgment

The authors are grateful to the anonymous reviewers for the valuable suggestions and comments in order to improve the quality of this manuscript.

References

[1] H. Adeli and S.L. Hung, Machine learning-neural networks, genetic algorithms, and fuzzy sets, John Wiley and Sons, New York, 1995.

[2] A. Auge and N. Hansen, A restart CMA evolution strategy with increasing population size, in: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC-2005* (2005), 1769–1776.

[3] J.L. Ávila, E.L. Gibaja and S. Ventura, Multi-label classification with gene expression programming, *LNCS (LNAI)*, Springer, Heidelberg, **5572** (2009), 629–637.

[4] P. Baraldi, R. Canesi, E. Zio, R. Seraoui and R. Chevalier, Genetic algorithm-based wrapper approach for grouping condition monitoring signals of nuclear power plant components, *Integrated Computer-Aided Engineering* **18**(3) (2011), 221–234.

[5] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei and M.I. Jordan, *Matching words and pictures, Journal of Machine Learning Research* **3** (2003), 1107–1135.

[6] M.G. Bergmann and G. Hommel, Improvements of general multiple test procedures for redundant systems of hypotheses, *Multiple Hypotheses Testing*, Springer, Berlin, (1988), 100–115.

[7] H. Blockeel, L. Raedt and J. Ramon, Top-down induction of clustering trees, in: *Proceedings of the 15th International Conference on Machine Learning* (1998), 55–63.

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

[9] K. Brinker, J. Fürnkranz and E. Hüllermeier, An unified model for multi-label classification and ranking, in: *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 06)* (2006), 489–493.

[10] L. Carro-Calvo, S. Salcedo-Sanz, G. Ortiz-Garc and A. Portilla-Figueras, An incremental-encoding evolutionary algorithm for color reduction in images, *Integrated Computer-Aided Engineering* **17**(3) (2010), 261–269.

[11] T. Chabuk, J.A. Reggia, J. Lohn and D. Linden, Causally-guided evolutionary optimization and its application to antenna array design, *Integrated Computer-Aided Engineering* **19**(2) (2012), 111–124.

[12] W. Cheng and E. Hüllermeier, Combining instance-based learning and logistic regression for multi-label classification, *Machine Learning* **76**(2–3) (2009), 211–225.

[13] A. Clare and R. King, Knowledge discovery in multi-label phenotype data, in: *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001)* (2001), 42–53.

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

[15] J. Demšar, Statistical comparisons of classifiers over multiple datasets, *Journal of Machine Learning Research* **7** (2006), 1–30.

[16] E. Deza and M.M. Deza, Dictionary of distances, Elsevier, (2006).

[17] S. Diplarisa, G. Tsoumakas, P. Mitkas and I. Vlahavas, Protein classification with multiple algorithms, in: *Proceedings 10$^{\text{th}}$ Panhellenic Conference on Informatics (PCI 2005)* (2005), 448–456.

[18] P. Duygulu, K. Barnard, N. de Freitas and D. Forsyth, Object recognition as machine translation: Learning a lexicon for

a fixed image vocabulary, in: *Proceedings of 7th European Conference on Computer Vision* (2002), 97–112.
[19] A. Elisseeff and J. Weston, A kernel method for multi-labeled classification, in: *Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval* (2005), 274–281.
[20] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, *Annals of Mathematical Statistics* **11** (1940), 86–92.
[21] C. Fuggini, E. Chatzi, D. Zangani and T.B. Messervey, Combining genetic algorithm with a meso-scale approach for system identification of a smart polymeric textile, *Computer-Aided Civil and Infrastructure Engineering* **28**(3) (2013), 227–245.
[22] J. Fürnkranz, E. Hüllermeier, E. Mencia and K. Brinker, Multi-label classification via calibrated label ranking, Machine Learning, (2008).
[23] S. García and F. Herrera, An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, *Journal of Machine Learning Research* **9** (2008), 2677–2694.
[24] N. Hansen and A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evolutionary Computation* **9**(2) (2001), 159–195.
[25] F.Y. Hsiao, S.S. Wang, W.C. Wang, C.P. Wen and W.D. Yu, Neuro-fuzzy cost estimation model enhanced by fast messy genetic algorithms for semiconductor hookup construction, *Computer-Aided Civil and Infrastructure Engineering* **27**(10) (2012), 764–781.
[26] S.L. Hung and H. Adeli, A parallel genetic/neural network learning algorithm for mimd shared memory machines, *IEEE Transactions on Neural Networks* **5**(6) (1994), 900–909.
[27] I. Katakis, G. Tsoumakas and I. Vlahavas, Multilabel text classification for automated tag suggestion, in: *Proceedings of the ECML/PKDD Discovery Challenge* (2008).
[28] H. Kim and H. Adeli, Discrete cost optimization of composite floors using a floating point genetic algorithm, *Engineering Optimization* **33**(4) (2001), 485–501.
[29] K. Kira and L. Rendell, A practical approach to feature selection, in: *Proceedings of Int Conf on Machine Learning*, Morgan Kaufmann, (1992), 249–256.
[30] B. Klimt and Y. Yang, The enron corpus: A new dataset for email classification research, in: *Proceedings of the 15th European Conference on Machine Learning* (2004), 217–226.
[31] D. Kocev, C. Vens, J. Struyf and S. Džeroski, Ensembles of multi-objective decision trees, in: *Proceedings of the 18th European conference on Machine Learning* (2007), 624–631.
[32] D. Kocev, Ensembles for predicting structured outputs, Ph. D. thesis, IPS Jozef Stefan, Ljubljana, Slovenia (2011).
[33] D. Kong, C. Ding, H. Huang and H. Zhao, Multi-label ReliefF and F-statisticfeature selections for image annotation, in: *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (2012), 2352–2359.
[34] I. Kononenko, Estimating attributes: Analysis and extension of ReliefF, in: *Proceedings of the 7th European Conference in Machine Learning, ECML94*, Springer-Verlag, (1994), 171–182.
[35] T. Li and M. Ogihara, Detecting emotion in music, in: *Proceedings of the International Symposium on Music Information Retrieval* (2003).
[36] X. Lin and X.W. Chen, Mr.kNN: Soft relevance for multi-label classification, in: *Proceedings of the CIKM-10*, Toronto, Canada, ACM, (2010).
[37] G. Madjarov, D. Kocev, D. Gjorgjevikj and S. Džeroski, An extensive experimental comparison of methods for multi-label learning, *Pattern Recognition* **45** (2012), 3084–3104.
[38] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: *Working Notes of the AAAI-99 Workshop on Text Learning* (1999).
[39] E.L. Mencía and J. Fürnkranz, Pairwise learning of multi-label classifications with perceptrons, in: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN-08)* (2008), 2900–2907.
[40] E.C. Pedrino, V.O. Roda, E.R.R. Kato, J.H. Saito, M.L. Tronco, R.H. Tsunaki, O. Morandin and M.C. Nicoletti, A genetic programming based system for the automatic construction of image filters, *Integrated Computer-Aided Engineering* **20**(3) (2013).
[41] R. Putha, L. Quadrifoglio and E. Zechman, Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions, *Computer-Aided Civil and Infrastructure Engineering* **27**(1) (2012), 14–28.
[42] J.L. Olmo, J.M. Luna, J.R. Romero and S. Ventura, Mining association rules with single and multi-objective grammar guided ant programming, *Integrated Computer-Aided Engineering* **20**(3) (2013).
[43] J. Read, B. Pfahringer and G. Holmes, Multi-label classification using ensembles of pruned sets, in: *Proceedings of the 8th IEEE International Conference on Data Mining* (2008), 995–1000.
[44] J. Read, A pruned problem transformation method for multi-label classification, in: *Proceedings 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)* (2008), 143–150.
[45] J. Read, B. Pfahringer, G. Holmes and E. Frank, Classifier chains for multi-label classification, in: *Proceedings of the 20th European Conference on Machine Learning* (2009), 254–269.
[46] O. Reyes, C. Morell and S. Ventura, Learning similarity metric to improve the performance of lazy multi-label ranking algorithms, in: *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, IEEE, (2012).
[47] M. Rizzi, M. D'Aloia and B. Castagnolo, A supervised method for microcalcification cluster diagnosis, *Integrated Computer-Aided Engineering* **20**(2) (2013), 157–167.
[48] M. Robnik-Sikonja and I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, *Machine Learning* **53**(1) (2003), 23–69.
[49] T.K. Ryan, J.L.T. Chiu, H.J. Dai, R.T.H. Tsai, M.Y. Day and W.L. Hsu, A supervised learning approach to biological question answering, *Integrated Computer-Aided Engineering* **16**(3) (2009), 271–281.
[50] K.C. Sarma and H. Adeli, Bi-level parallel genetic algorithms for optimization of large steel structures, *Computer-Aided Civil and Infrastructure Engineering* **16**(5) (2001), 295–304.
[51] R. Schapire and Y. Singer, Boostexter: A boosting-based system for text categorization, *Machine Learning* **39** (2000), 135–168.
[52] K. Sechidis, G. Tsoumakas and I. Vlahavas, On the stratification of multi-label data, in: *Proceedings of the 2011 European conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD-11* Part III, Springer-Verlag, (2011), 145–158.
[53] L. Sgambi, K. Gkoumas and F. Bontempi, Genetic algorithms for the dependability assurance in the design of a long span suspension bridge, *Computer-Aided Civil and Infrastructure*

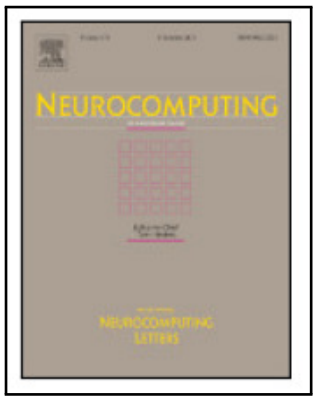
Engineering **27**(9) (2012), 655–675.
[54] Y. Shafahi and M. Bagherian, A customized particle swarm method to solve highway alignment optimization problem, *Computer-Aided Civil and Infrastructure Engineering* **28**(1) (2013), 52–67.
[55] N. Spolaor, E. Alvares, M. Carolina and H. Diana, A comparison of multi-label feature selection methods using the problem transformation approach, *Electronic Notes in Theoretical Computer Science* (292) (2013), 135–151.
[56] E. Spyromitros, G. Tsoumakas and I. Vlahavas, An empirical study of lazy multilabel classification algorithms, *LNAI*, Springer-Verlag, Heidelberg, (2008), 401–406.
[57] A. Srivastava and B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: *Proceedings of the IEEE Aerospace Conference* (2005), 55–63.
[58] H. Tao, J.M. Zain, M.M. Ahmed, A.N. Abdalla and W. Jing, A wavelet-based particle swarm optimization algorithm for digital image watermarking, *Integrated Computer-Aided Engineering* **19**(1) (2012), 81–91.
[59] K. Trohidis, G. Tsoumakas, G. Kalliris and I. Vlahavas, Multi-label classification of music into emotions, in: *Proceedings 2008 International Conference on Music Information Retrieval (ISMIR 2008)* (2008), 325–330.
[60] G. Tsoumakas and I. Vlahavas, Random k-label sets: An ensemble method for multi-label classification, in: *Proceedings of the 18th European Conference on Machine Learning* (2007), 406–417.
[61] G. Tsoumakas and I. Katakis, Multi-label classification: An overview, *International Journal of Data Warehousing & Mining* **3** (2007), 1–13.
[62] G. Tsoumakas, I. Katakis and I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: *Proceedings of the ECML/PKDD Workshop on Mining Multidimensional Data* (2008), 30–44.
[63] G. Tsoumakas, I. Katakis and I. Vlahavas, Mining multi-label data, data mining and knowledge discovery handbook, 2nd Edition, Springer, 2010.
[64] G. Tsoumakas, E. Spyromitros-Xioufi, J. Vilcek and I. Vlahavas, MULAN: A java library for multi-label learning, *Journal of Machine Learning Research* **12** (2011), 2411–2414.
[65] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Transactions on Audio, *Speech and Language Processing* **16**(2) (2008), 467–476.
[66] E.D. Wandekokem, E. Mendel, F. Fabris, M. Valentim, R.J. Batista, F.M. Varejao and T.W. Rauber, Diagnosing multiple faults in oil rig motor pumps using support vector machine classifier ensembles, *Integrated Computer-Aided Engineering* **18**(1) (2011), 61–74.
[67] D. Wettschereck, D.W. Aha and T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, *Artificial Intelligence Review* **11** (1997), 273–314.
[68] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics* **1** (1945), 80–83.
[69] D. Wilson and T.R. Martinez, Improved heterogeneous distance functions, *Journal of Artificial Intelligence Research (JAIR)* **6** (1997), 1–34.
[70] I. Witten and E. Frank, Data mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann (2005).
[71] M. Worring, C.G.M. Snoek, J.C. van Gemert, J.-M. Geusebroek and A.W.M. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: *Proceedings of the 14th Annual ACM International Conference on Multimedia* (2006), 421–430.
[72] J. Xu, Multi-label weighted k-nearest neighbour classifier with adaptive weight estimation, in: *Proceedings of ICONIP 2011, Part II, LNCS*, Springer, Heidelberg, **7063** (2011), 79–88.
[73] S. Yang, S. Kim and Y. Ro, Semantic home photo categorization, circuits and systems for video technology, *IEEE Transactions* **17** (2007), 324–335.
[74] Z. Younes, F. Abdallah and T. Denceux, Multi-label classification algorithm derived from k-nearest neighbour rule with label dependencies, in: *Proceedings of 16$^{\text{th}}$ European Signal Processing Conference (EUSIPCO 2008)* (2008).
[75] Z. Younes, F. Abdallah and T. Denceux, An evidence-theoretic k-nearest neighbour rule for multi-label classification, *SUM 2009, LNAI 5785*, Springer-Verlag, Berlin Heidelberg, (2009), 297–308.
[76] M.L. Zhang and 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.
[77] M.L. Zhang and Z.H. Zhou, ML-kNN: A lazy learning approach to multi-label learning, *Pattern Recognition* **40**(7) (2007), 2038–2048.

TITLE:

Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context

AUTHORS:

O. Reyes, C. Morell, and S. Ventura



Neurocomputing, *Volume 161, pp. 168-182, 2015*

RANKING:

Impact factor (JCR 2015): 2.392

Knowledge area:

Computer Science, Artificial Intelligence: 31/130

DOI: 10.1016/J.NEUCOM.2015.02.045

Neurocomputing 161 (2015) 168–182

Contents lists available at ScienceDirect

Neurocomputing

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

Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context

Oscar Reyes [a], Carlos Morell [b], Sebastián Ventura [c,d,*]

[a] Department of Computer Science, University of Holguín, Cuba
[b] Department of Computer Science, Universidad Central de Las Villas, Cuba
[c] Department of Computer Science and Numerical Analysis, University of Córdoba, Spain
[d] Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 4 September 2014
Received in revised form
12 February 2015
Accepted 13 February 2015
Communicated by Jiayu Zhou
Available online 21 February 2015

Keywords:
Multi-label learning
ReliefF algorithm
Feature weighting
Feature selection
Multi-label classification
Label ranking

ABSTRACT

Multi-label learning has become an important area of research due to the increasing number of modern applications that contain multi-label data. The multi-label data are structured in a more complex way than single-label data. Consequently the development of techniques that allow the improvement in the performance of machine learning algorithms over multi-label data is desired. The feature weighting and feature selection algorithms are important feature engineering techniques which have a beneficial impact on the machine learning. The ReliefF algorithm is one of the most popular algorithms to feature estimation and it has proved its usefulness in several domains. This paper presents three extensions of the ReliefF algorithm for working in the multi-label learning context, namely ReliefF-ML, PPT-ReliefF and RReliefF-ML. PPT-ReliefF uses a problem transformation method to convert the multi-label problem into a single-label problem. ReliefF-ML and RReliefF-ML adapt the classic ReliefF algorithm in order to handle directly the multi-label data. The proposed ReliefF extensions are evaluated and compared with previous ReliefF extensions on 34 multi-label datasets. The results show that the proposed ReliefF extensions improve preceding extensions and overcome some of their drawbacks. The experimental results are validated using several nonparametric statistical tests and confirm the effectiveness of the proposal for a better multi-label learning.

1. Introduction

Traditional machine learning applications have been derived from the analysis of data where the examples are associated with a single label [1]. However, recently studies over data that are structured in a more complex way than single-label data have received especial attention. Multi-label problems are concerned to those problems where the examples belong to a set of labels at the same time [2,3]. The goal of the Multi-Label Learning (MLL) paradigm is to learn a model that correctly generalises unseen multi-label data [2,3]. On the MLL context two problems are studied, multi-label classification (MLC) and label ranking (LR). MLC divides the set of labels into relevant and irrelevant sets, whereas the LR provides an ordering of the labels for a given query instance [3,4].

In the last few years, an increasing number of modern applications that contain multi-label data have appeared, such as text categorisation [5], emotions evoked by music [6], semantic annotation of images [7] and videos [8], classification of protein function [9] and gene [10,11].

Generally speaking, the multi-label datasets contain a large number of features that describe the instances, e.g. description of texts, images, proteins and genes [5,7–16]. The irrelevant, interacting, redundant and noisy features have a highly negative impact in the performance of the learning algorithms [17]. Moreover, the number of features is much bigger than the number of instances in several multi-label applications [13]. On the other hand, in some domains the number of possible labels can be in the region of hundreds (even thousands) and the distribution of instances per label can be showed in a non-uniform way [8,12,14–16]. Consequently, some multi-label learning algorithms present a poor performance with regard to time complexity and efficiency [4]. As a result of the above situations, nowadays the designing process of MLL algorithms faces several challenges and it is an open field of research.

The preprocessing techniques have demonstrated to be an important step of the knowledge discovery in databases [1,18]. Feature engineering techniques such as feature weighting (FW) and feature selection (FS) improve the performance of machine

* Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, Spain, Tel.: +34 957212218; fax: +34 957218630.
E-mail addresses: oreyesp@facinf.uho.edu.cu (O. Reyes), cmorellp@uclv.edu.cu (C. Morell), sventura@uco.es (S. Ventura).

http://dx.doi.org/10.1016/j.neucom.2015.02.045
0925-2312/

learning algorithms [19,20]. The FW assigns a weight to each feature representing the usefulness of the feature to distinguish pattern classes. The weight vector can be used to improve the performance of the lazy algorithms, parameterising the distance function used to retrieve the *k*-nearest neighbours of a given query instance [19]. Furthermore, the weight vector can be used as a ranking of features for guiding the search of the best subset of features [21–26]. The FS can be seen as a specific case of the FW process [19], where the feature weights are binary values representing whether a feature is removed or conserved. FS tries to reduce the dimensionality, which has a positive effect on the efficiency, effectiveness and comprehensibility of machine learning [20,27–29].

The FS algorithms can be divided into three main categories, filter, wrapper and embedded methods. The filter methods are independent of any classification algorithm, so that the biases of the learning algorithms do not influence the feature selection method. The filter methods evaluate the usefulness of a feature, or set of features, through measures of distance, dependency, information or correlation on data. The major disadvantage of the filter approach is that it ignores the effects of the selected features on the performance of a learning algorithm. Consequently the selected features may be suboptimal features for a certain classification algorithm [30].

The wrapper approach uses a specific classifier to evaluate the quality of selected features. The wrapper methods are dependent on the classification algorithm selected, therefore the bias of a learning algorithm influences the feature selection method. It obtains better performance for a predefined classifier. The major disadvantage of the wrapper approach is that it is computationally expensive [30].

The embedded methods include the feature selection in the classifier construction process, taking into account the interaction with the learning model, but it is less computational expensive than wrapper methods. Nowadays, structured sparsity learning is a promising paradigm for learning in high-dimensional settings. In the embedded approach by regularisation methods, the classifier induction and feature selection are achieved simultaneously by estimating a weight vector. Theoretical and empirical studies have suggested the usefulness of structure sparsity for feature learning, e.g. LASSO and its extensions [31–35]. Several adaptations of these type of methods have been proposed for the Multi-task Learning paradigm [36–38], that it can be considered a generalisation of MLL paradigm.

In this work we focus in the feature selection based in filter approach for MLL, specifically in the Relief family of algorithms. ReliefF [22,23] is one of the most popular feature estimation algorithms for working on single-label data. The ReliefF algorithm follows the filter approach; since it does not use the feedback from the classifier to assign the weights to the features. ReliefF learns a single set of weights that are employed globally over the entire instance space. ReliefF does not employ domain specific knowledge to set feature weights [19,23]. Traditionally, the ReliefF applications have been focused in the FS [21,24] process. However, ReliefF has proved its usefulness in other domains as feature ranking [39], building tree-based models [23], association rules-based classifier [40], constructive induction [41], and improving the efficiency of the genetic algorithms [42] and lazy classifiers [19].

The problem transformation methods (PTM) decompose a multi-label problem into one or more single-label problems [3]. Generally, the estimation of the utility of features on multi-label data has been done using a PTM [3,43–47]. A PTM transforms the multi-label problem and by an aggregation strategy the score of a feature is computed using a single-label feature estimation algorithm.

Owing to the simplicity and effectiveness of the ReliefF algorithm, several extensions of the ReliefF on the MLL context have been proposed in the literature [26,46–48]. Traditionally, the multi-label ReliefF extensions work together with a PTM, and finally, the weights for each feature are computed by some aggregation strategy, e.g. the average, minimum and maximum.

These previous works are an important contribution to MLL for dealing with high dimensional data. However, these approaches have several limitations. First, the performance of a PTM generally depends on the number of labels of the dataset. Consequently, they are very expensive for domains that contain a moderate number of labels. Second, a drawback of some PTM is that they do not consider label correlations. Third, the previous ReliefF extensions have been restricted on FS area and few multi-label datasets are used in the experimental study. Fourth, they are not usually compared with other ReliefF extensions in order to determine which ReliefF extension achieves a superior performance.

The usefulness of ReliefF as a feature estimation algorithm and the drawbacks presented by the previous studies have led to the design of new scalable ReliefF extensions for working on the MLL context. For this, three new ReliefF extensions, namely ReliefF-ML, PPT-ReliefF and RReliefF-ML, that outperform previous ReliefF extensions have been proposed. ReliefF-ML can be considered as a generalisation of the classic ReliefF, where the updating weights equation is modified. PPT-ReliefF uses the Pruned Problem Transformation method (PPT) [49] to convert the original multi-label dataset into a new multi-class dataset. On the other hand, RReliefF-ML is based on the principles of the well-known adaptation of ReliefF to regression problems [50]. The three proposed ReliefF extensions include interaction among features and take into account label dependencies. PPT-ReliefF depends on a PTM, whereas ReliefF-ML and RReliefF-ML handle the multi-label data directly.

In this paper, an analysis of the ReliefF extensions on the FW process to improve the performance of the multi-label lazy algorithms is carried out. The lazy algorithms depend on the definition of a distance function that determines the *k*-nearest neighbours of a query instance. The distance function is defined on feature space, therefore using a weight vector can reduce the negative impact of the irrelevant, redundant or noise features on distance computation [19]. The proposal is proved on four multi-label *k*-NN algorithms. On the other hand, the ReliefF extensions are tested on the FS process. The weight vector is transformed in a ranking of features, which is used for guiding the search of the best subset of features. The proposal is proved on one multi-label algorithm, which is completely sensitive to the presence of irrelevant, redundant and noise features.

The experiments are carried out on 34 multi-label datasets, considering different problem domains, number of instances, features and labels. Several multi-label evaluation measures are used to analyse different viewpoints. The experimental stage shows the effectiveness of the proposal, obtaining significantly better results than previous ReliefF extensions on MLC and LR tasks. The experimental study includes a statistical analysis based on several nonparametric test as proposed in [51–54].

The main contribution of the present work is the enhancement of the understanding of the ReliefF's applicability on the MLL context. This work aims to study the benefits of the ReliefF algorithm for a better MLL. To the best of our knowledge, this paper presents the first attempt to compare the most significant multi-label ReliefF extensions over a large number of multi-label datasets, two main areas of study (FS and FW process) and two MLL tasks (MLC and LR tasks).

This paper is arranged as follows: Section 2 describes the MLL paradigm and the ReliefF extensions to multi-label learning that have appeared in the literature. Section 3 presents the ReliefF-ML,

PPT-ReliefF and RReliefF-ML extensions. Section 4 describes the experimental set-up and analyses the experimental results. Finally, Section 5 provides some concluding remarks.

2. Preliminaries

In this section the general definition of the MLL paradigm and the multi-label evaluation measures used in this work are presented. A general background of the Relief family of algorithms is shown, highlighting the simplicity, effectiveness and applicability of the ReliefF algorithm. Finally, the previous extensions of ReliefF to MLL that have appeared in the literature are briefly described.

2.1. Multi-label learning paradigm

A multi-label dataset can be defined as follows:

- A feature space F with a cardinality equal to d (number of features) and a label space L with a cardinality equal to q (number of labels).
- A set E of n instances, where each instance i is a tuple $\langle x_i, y_i\rangle$. x_i is the feature vector of the instance i. A feature vector is a tuple of values of features, where x_{if} represents the value of fth feature for the instance i. y_i is the set of labels of the instance i, it is a subset of the label space L.

The goal of the Multi-Label Learning (MLL) is to perform a machine learning process from instances that are associated with multiple labels at the same time [3]. The Multi-label Classification (MLC) task is concerned with learning a predictive model that divides the label space into relevant and irrelevant label sets. The binary and multi-class classification are specific cases of the MLC task [3]. On the other hand, the Label Ranking (LR) task is concerned with learning a predictive model that for a given query instance, provides an ordering of the labels. The generalisation of the MLC and LR tasks has been called Multi-label Ranking (MLR) [55].

2.1.1. Multi-label evaluation measures

Given a test set $T=\{\langle x_i, y_i\rangle, i=1\ldots m\}$, a MLC method predicts a set of labels z_i for a given query instance i. On the other hand, a LR method provides a ranking of labels r_i for a given query instance i, being $r_i(\lambda)$ the rank predicted for the label λ[3].

The Hamming Loss measure (H_L) [56] averages the symmetrical differences among the predicted and actual label sets.

$$H_L=\frac{1}{m}\sum_{i=1}^{m}\frac{|y_i\Delta z_i|}{q} \tag{1}$$

where Δ denotes the symmetric difference among two sets.

The Example-Based F1 measure (F_{1Ex}) [57] is the harmonic mean of precision and recall.

$$F_{1Ex}=\frac{1}{m}\sum_{i=1}^{m}\frac{2|y_i\cap z_i|}{|y_i|+|z_i|} \tag{2}$$

The Ranking Loss measure (R_L) [3] averages the proportion of label pairs that are incorrectly ordered.

$$R_L=\frac{1}{m}\sum_{i=1}^{m}\frac{|(\lambda_a,\lambda_b): r_i(\lambda_a)>r_i(\lambda_b),(\lambda_a,\lambda_b)\in y_i\times\overline{y_i}|}{|y_i||\overline{y_i}|} \tag{3}$$

where $\overline{y_i}$ denotes the complementary set of y_i in L.

The Average Precision measure (A_P) [3] averages how many times a particular label is ranked above other label which actually is in the truth label set.

$$A_P=\frac{1}{m}\sum_{i=1}^{m}\frac{1}{|y_i|}\sum_{\lambda\in y_i}\frac{|\{\lambda'\in y_i:\ r_i(\lambda')\le r_i(\lambda)\}|}{r_i(\lambda)} \tag{4}$$

The H_L and F_{1Ex} evaluation measures are associated with the MLC task, whereas R_L and A_P are associated with the LR task. The higher the value of F_{1Ex} and A_P, and the lower the value of H_L and R_L, the better the performance of a MLL algorithm.

2.2. ReliefF algorithm

In [21] was initially presented the Relief family of algorithms. Relief is a classical method for feature estimation in single-label data. The Relief method was designed for binary class problems without missing values. The method randomly selects m instances from the training set. For each selected instance i, Relief computes the nearest neighbour from the same class of i and the nearest neighbour from the opposite class. The quality of each feature is estimated with respect to whether the feature differentiates two instances from the same class and from different classes. A feature has a undesired property if it differentiates two near instances that belong to the same class. A feature has a desired property if it differentiates two near instances that belong to different classes [24]. The Relief family of algorithms can be implemented efficiently and avoids an expensive search on the feature space.

In [22] an extension of Relief method called ReliefF was proposed. ReliefF deals with incomplete and noisy data, and with multi-class problems. For each sampling instance i, the k-nearest neighbours from the same class (*Hits* neighbours) of i and for each different class (*Misses* neighbours) are determined. ReliefF takes into account the information of the k-nearest neighbours to increase the precision of the feature estimation. ReliefF takes into account the effect of interacting features, although it is sensitive to redundant and irrelevant features [24]. ReliefF returns a weight vector W, where W_f represents the weight for the fth feature. The weights of the vector W are in $[-1,1]$ range. The higher the weight assigned to the feature, the more useful the feature.

The time complexity of the ReliefF algorithm is $O(m\cdot n\cdot d)$. However, several search algorithms can be used to reduce the time complexity of ReliefF to $O(d\cdot n\cdot \log n)$, such as kD-Tree, Ball-Tree or Cover-Tree. Also, ReliefF can be split into independent tasks, which is a requisite for the parallelisation of an algorithm [1,24,58].

In [59] is empirically observed that the Relief family of algorithms approximate a gradient ascent based in a margin-based criterion. In [60,61] a mathematical interpretation of the heuristic used by Relief as an online method that solves a convex optimisation problem with a margin-based objective function is presented.

Similar to structured sparsity learning methods, ReliefF implicitly resolves an optimisation problem to find the best weight vector. However, the objective function that ReliefF optimises does not depend of the feedback of a classifier. The structured sparsity learning methods (e.g. regularisation methods) optimise an objective function that it is a composite of a loss function (e.g. quadratic loss, hinge loss or logistic loss) and a regularisation term (e.g. LASSO or Bridge regularisation). The computational cost of evaluating the loss function depends on the performance of the classification algorithm used. Consequently, the efficiency of the algorithm used to optimise the objective function is a primordial point in this type of methods.

The interpretability of the feature estimation with ReliefF makes it more attractive than other feature estimation algorithms [58]. The mathematical interpretation of the Relief scheme explains the success of this family of algorithms in several real applications [19,21,23,24,39–42]. The ease of application of ReliefF

has been proven on other problems, such as in the evaluation of literals in inductive logic programming [62], cost-sensitive classification [63], evaluation of features with ordered values [64] and learning similarity metrics [65].

Owing to the simplicity, efficiency, effectiveness and applicability of the ReliefF algorithm, it has been also adapted to other machine learning paradigms, such as Multi-Instance Learning [66], Multi-Target Learning in the context of structured targets [67] and MLL [26,46–48,68].

2.2.1. Previous extensions of ReliefF to multi-label learning

Several ReliefF extensions to MLL have appeared in the literature. The previous ReliefF extensions use a PTM to transform the multi-label dataset into one or more single-label datasets. Afterwards, the classic ReliefF algorithm is performed on every single-label dataset generated and finally the feature weights are computed by an aggregation strategy.

The BR-ReliefF extension is proposed in [48]. BR-ReliefF is evaluated on ten multi-label datasets in [26]. Later on [47], BR-ReliefF and a multi-label feature estimation algorithm based on the Information Gain measure are compared. BR-ReliefF uses the Binary Relevance (BR) [3] approach to measure the contribution of each feature according to each label. The BR is a PTM, that decomposes the multi-label problem into q binary classification problems. The classic ReliefF is performed for each label and the average of the score of each feature across all labels is computed. The BR-ReliefF extension makes $O(q \cdot m \cdot n \cdot d)$ operations, owing to BR-ReliefF executes q times the classic ReliefF. The efficiency of BR-ReliefF is affected by datasets with a high number of labels. Moreover, the BR approach does not consider label correlations.

The MReliefF extension is presented in [46]. The multi-label problem is decomposed into a set of pairwise single-label problems. This decomposition is equivalent to breaking down the multi-label problem into several binary classification problems using the Ranking by Pair Wise Comparison (RPC) [3] approach. The RPC is a PTM that decomposes the multi-label problem into $q(q-1)/2$ binary classification problems. MReliefF excludes those examples that fall into *Hits* and *Misses* neighbours at the same time. In [46] the authors write that the occurrence of these cases is very rare and therefore the exclusion of these instances does not significantly affect the results. However, this is mentioned because the two specific datasets used possess this property. MReliefF carries out $O(q^2 \cdot m \cdot n \cdot d)$ operations, owing to MReliefF executes $q(q-1)/2$ times the classic ReliefF. This ReliefF extension has a high computational cost on datasets with a moderate number of labels.

The LP-ReliefF extension is proposed in [47]. LP-ReliefF uses the Label Power Set approach (LP) [2] to measure the contribution of each feature directly from the multi-class problem generated. The LP approach is a PTM that constructs one multi-class dataset from a multi-label dataset. LP considers each unique set of labels as one of the classes of the new multi-class dataset. The time complexity of the LP-ReliefF extension is $O(m \cdot n \cdot d)$. The LP method needs $O(n \cdot q)$ steps to construct a multi-class dataset from a multi-label dataset and the classic ReliefF needs $O(m \cdot n \cdot d)$ steps, but $m \cdot n > n$ and generally $d > q$ in real multi-label problems.

LP takes into account implicitly label correlations. However, the LP method has a high complexity on multi-label datasets which present a large number of distinct label sets, limiting its scalability and leading to a tendency to over-fitting the data [49,69]. Several set of labels can sporadically appear on multi-label problems, resulting in a scarcity of instances on the generated multi-class dataset. Consequently, the LP method can generate a highly imbalanced multi-class datasets.

Previous extensions of ReliefF have only dealt with the FS process to improve the efficiency and effectiveness of MLL. In addition, the present work also focuses on the FW process to improve the performance of several multi-label lazy algorithms.

3. Scalable extensions of ReliefF to multi-label learning

In this section the basis of three new ReliefF extensions to MLL is explained. PPT-ReliefF uses a PTM approach, whereas the two other extensions handle the multi-label data directly.

3.1. ReliefF-ML

We initially proposed the ReliefF-ML extension in [68]. ReliefF-ML was evaluated on FW process to improve the performance of three multi-label lazy algorithms on eleven multi-label datasets, showing the effectiveness of the proposal.

An instance in MLL context is assigned to multiple labels at the same time. Consequently, the nearest *Hits* and *Misses* concepts of ReliefF algorithm cannot be used in a strict sense. Given a multi-label dataset, the prior probability that an instance belongs to a label l can be computed as follows [70]:

$$P_l = \frac{C_l + b}{n + 2b} \tag{5}$$

where C_l is the number of instances in the dataset that belong to label l, n is the number of the instances of the training set, and b is the smoothing parameter controlling the strength of uniform prior ($b=1$ yields the Laplace smoothing).

Given two instances i and j, the distance among the sets of labels of i and j is calculated by the Hamming Distance (see Eq. (6). The distance $d_{\mathcal{L}}$ represents a measure of how much the sets of two instances differ. A smaller value of $d_{\mathcal{L}}$ represents a major similarity in the classification of these instances.

$$d_{\mathcal{L}}(i,j) = \frac{|y_i \Delta y_j|}{q} \tag{6}$$

For each relevant and irrelevant label of a sampling instance i a group of k-nearest neighbours is defined. The following groups of *Hits* (H_i^l) and *Misses* (M_i^l) with respect to an instance i are computed:

- H_i^l: k-nearest neighbours that have the relevant label l of i as relevant label.
- M_i^l: k-nearest neighbours that have the irrelevant label l of i as relevant label.

Based in the defined groups H_i^l and M_i^l, the following probabilities are defined:

$$P_{H_i^l} = \frac{\sum_{j \in H_i^l} d_{\mathcal{L}}(i,j)}{k} \tag{7}$$

$$P_{M_i^l} = \frac{\sum_{j \in M_i^l} d_{\mathcal{L}}(i,j)}{k} \tag{8}$$

$P_{H_i^l}$ represents the probability that two nearest instances that share the label l as relevant belong to different set of labels. $P_{M_i^l}$ represents the probability that two nearest instances belong to different set of labels, where i has the label l as irrelevant and the k-nearest neighbours have the label l as relevant.

ReliefF-ML takes into account the dependence among labels by the calculus of the probabilities $P_{H_i^l}$ and $P_{M_i^l}$. A feature weight reflects the ability of the feature to distinguish class labels. A high feature weight indicates that the feature has different values on instances with dissimilar sets of labels. Also, a high feature weight indicates that the feature has similar values on instances with similar sets of labels. ReliefF-ML iteratively updates the feature

weights as follows:

$$W_f = W_f - \sum_{l \in y_i}\left(\frac{P_l}{\sum_{q \in y_i} P_q}\frac{1-P_{H_i^l}}{1+P_{H_i^l}}\sum_{j \in H_i^l}\frac{\delta(x_{if},x_{jf})}{mk}\right) + \sum_{l \in \overline{y_i}}\left(\frac{P_l}{\sum_{q \in \overline{y_i}} P_q}P_{M_i^l}\sum_{j \in M_i^l}\frac{\delta(x_{if},x_{jf})}{mk}\right) \quad (9)$$

where the function $\delta(x_{if}, x_{jf})$ calculates the difference between the values of the fth feature on the instances i and j. The parameter m represents the number of sampling instances selected from the training set to estimate the feature weights.

The contributions of each relevant and irrelevant label are weighted by the factors $P_l/\sum_{q \in y_i} P_q$, $(1-P_{H_i^l})/(1+P_{H_i^l})$ and $P_l/\sum_{q \in \overline{y_i}} P_q$, $P_{M_i^l}$ respectively. ReliefF-ML uses the given representation of the original datasets, i.e. it does not use a PTM.

ReliefF-ML requires the retrieval of the k-nearest neighbours for each relevant and irrelevant label of an instance i. However, through a linear search over the training set the groups of the k-nearest neighbours for an instance i can be find efficiently. Consequently, the time complexity of ReliefF-ML is equal to the classic ReliefF algorithm ($O(m \cdot n \cdot d)$). Algorithm 1 describes the ReliefF-ML extension.

Algorithm 1. ReliefF-ML algorithm.

Input: $E \rightarrow$ training set of multi-label instances
$m \rightarrow$ number of sampling instances
$k \rightarrow$ number of nearest neighbours
Output: $W \rightarrow$ weight vector
begin
  foreach $f \in F$ **do**
    $W_f \leftarrow 0$;
  end
  foreach $l \in L$ **do**
    $P_l \leftarrow$ `LabelProbability`(l); (equation 5)
  end
  for $t \leftarrow 1$ **to** m **do**
    $i \leftarrow$ Randomly pick an instance from E;
    foreach *label* $l \in y_i$ **do**
      $H_i^l \leftarrow$ `kNearestHits`(i, l);
      $P_{H_i^l} \leftarrow$ `Probability`(i, H_i^l); (equation7)
    end
    foreach *label* $l \in \overline{y_i}$ **do**
      $M_i^l \leftarrow$ `KNearestMisses`(i, l);
      $P_{M_i^l} \leftarrow$ `Probability`(i, M_i^l); (equation 8)
    end
    foreach $f \in F$ **do**
      $W_f \leftarrow$ `UpdateWeight`(f); (equation 9)
    end
  end
  Return W;
end

3.2. PPT-ReliefF

The principal drawback of the LP method is the generation of multi-class datasets with a large number of classes. Generally, the LP method returns an extreme class imbalance problem for learning. The Pruned Problem Transformation (PPT) method is proposed in [49] to address with the drawbacks of the LP method.

The simplest PPT method selects those label sets that occur more times than a defined threshold and discards the rest of the label sets. PPT has the power of the LP approach, where the correlation among labels is implicitly taken into account. Unlike to LP method, PPT only considers the most important label relationships. The PPT approach reduces the scarcity of labels and the over-fitting of data [49].

In [49] a more elaborated PPT approach that introduces disjoint subsets is presented. The disjointed subsets are extracted from those label sets that exist fewer times than a threshold. In general, the PPT approach contains two steps, a *pruning step* and a *subsampling step* of label sets. The *pruning step* removes the label sets that occur fewer times than a defined threshold. This step reduces the complexity of the generated multi-class dataset. However, if a high number of label sets is discarded then a considerable information can be lost. The *subsampling step* reintroduces into the training data those instances that have label subsets which occur more frequently, ensuring a minimal information loss [71].

The PPT method is influenced by the parameters p and s. The parameter p represents how much pruning needs to be carried out. A value of $p=0$ signifies that all the label sets are accepted (being the PPT method equivalent to the LP method). A value of $p=1$ indicates that all the instances that have a label set that appears at most one time are pruned, and so on.

The parameter s is related with the *subsampling step*. The parameter s represents how many subsets are recovered from those label sets previously discarded. When given a label set discarded, the PPT method finds all label subsets which occur more than p times. The label subsets are sorted according to their frequency of apparition. Afterwards, the top s label subsets are used to create new instances that are copies of the original multi-label instance. The new instances created have the top s label subsets previously selected as label sets.

The PPT method not only outperforms the LP method in predictive performance. Also, it improves the efficiency and scalability on larger datasets. PPT has a worse-case time complexity equal to LP method, $O(n \cdot q)$ [71].

In this work we propose a new ReliefF extension to MLL called PPT-ReliefF. The PPT-ReliefF uses the PPT approach to measure the contribution of each feature directly from the multi-class problem generated. The time complexity of PPT-ReliefF extension is $O(m \cdot n \cdot d)$. Algorithm 2 describes the proposed PPT-ReliefF extension.

Algorithm 2. PPT-ReliefF algorithm.

input: $E \rightarrow$ training set of multi-label instances
$m \rightarrow$ number of sampling instances
$k \rightarrow$ number of nearest neighbours
$p \rightarrow$ pruning parameter
$s \rightarrow$ number of label subsets to select
output: $W \rightarrow$ weight vector
begin
  `multiClassDataset` $\leftarrow$ `PPTTransformation`(E, p, s);
  $W \leftarrow$ `ReliefF`$($`multiClassDataset`$, m, k)$;
  return W;
end

In Algorithm 2 the function *PPTTransformation*(E, p, s) represents the PPT method which needs the multi-label dataset to convert, the pruning parameter (p) and the sampling parameter (s) as parameters. This function given a multi-label dataset returns a multiclass dataset. The function *ReliefF*$($*multiClassDataset*$, m, k)$ refers to the classic ReliefF algorithm. The classic ReliefF algorithm

only needs the multi-class dataset, the number of sampling instances (m) and the number of neighbours (k) as parameters.

3.3. RReliefF-ML

In this work we propose a new ReliefF extension to MLL called RReliefF-ML. RReliefF-ML is based on the well known ReliefF adaptation to regression problems (RReliefF) [50]. From the point of view of probability, the classic ReliefF algorithm estimates the feature weight as follows [22]:

$$W_f = P^f_{eqval|\,eqcl} - P^f_{eqval|\,difcl} \tag{10}$$

where $P^f_{eqval|\,eqcl}$ is the probability that nearest instances have the same value in the fth feature and the same prediction. $P^f_{eqval|\,difcl}$ is the probability that nearest instances have the same value in the fth feature and different classes. Using the Bayes Rule Eq. (10) is transformed into [24]:

$$W_f = \frac{P^f_{eqcl|\,eqval} P^f_{eqval}}{P_{eqcl}} - \frac{(1 - P^f_{eqcl|\,eqval}) P^f_{eqval}}{1 - P_{eqcl}} \tag{11}$$

and Eq. (11) is equivalent to [24]:

$$W_f = \frac{P^f_{difcl|\,difval} P^f_{difval}}{P_{difcl}} - \frac{(1 - P^f_{difcl|\,difval}) P^f_{difval}}{1 - P_{difcl}} \tag{12}$$

where $P^f_{difcl|\,difval}$ is the prior probability that nearest instances have different feature values in the fth feature and different predictions. P^f_{difval} is the prior probability that nearest instances have different feature values in the fth feature. P_{difcl} is the prior probability that nearest instances belong to different classes. In Relief family of algorithms, Eq. (12) represents a unified view of the feature quality estimation in classification and regression problems [24].

Analysing the unified view (Eq. (12)) on the MLL context, the estimation of a feature weight is computed as follows:

$$W_f = \frac{P^f_{difY|\,difval} \cdot P^f_{difval}}{P_{difY}} - \frac{(1 - P^f_{difY|\,difval}) \cdot P^f_{difval}}{1 - P_{difY}} \tag{13}$$

where $P^f_{difY|\,difval}$ is the prior probability that nearest instances have different feature values in the fth feature and belong to different set of labels. P_{difY} is the prior probability that nearest instances belong to different set of labels.

The prior probability P_{difY} is computed by the distance among the set of labels of two learning instances. Given two instances i and j, the distance between the sets of labels of i and j is calculated by the Hamming Distance (see Eq. (6)). The prior probability that nearest instances belong to different set of labels is computed as follows:

$$P_{difY} = \frac{\sum_{i \in S} \sum_{j \in N_i} d_{\mathcal{L}}(i,j)}{k} \tag{14}$$

where S is a set of m instances randomly selected from a training set E of multi-label instances. N_i represents the set of nearest neighbours of the instance i.

The prior probability that nearest instances have different feature values in the fth feature (P^f_{difval}) is computed as follows:

$$P^f_{difval} = \frac{\sum_{i \in S} \sum_{j \in N_i} \delta(x_{if}, x_{jf})}{k} \tag{15}$$

The prior probability that nearest instances have different feature values in the fth feature and belong to different set of labels ($P^f_{difY|\,difval}$) is computed as follows:

$$P^f_{difY|\,difval} = \frac{\sum_{i \in S} \sum_{j \in N_i} d_{\mathcal{L}}(i,j) \cdot \delta(x_{if}, x_{jf})}{k} \tag{16}$$

Algorithm 3 describes the main steps that follows the RReliefF-ML extension.

Algorithm 3. RReliefF-ML algorithm.

input: $E \rightarrow$ training set of multi-label instances
$m \rightarrow$ number of sampling instances
$k \rightarrow$ number of nearest neighbours
output: $W \rightarrow$ weight vector
begin
  $P_{difY} \leftarrow 0$;
  foreach $f \in F$ **do**
    $S_f \leftarrow 0$; $P^f_{difY|\,difval} \leftarrow 0$; $W_f \leftarrow 0$;
  end
  for $t \leftarrow 1$ **to** m **do**
    $i \leftarrow$ Randomly pick an instance from E;
    $N_i \leftarrow k$`NearestNeighbors` (i);
    foreach $j \in N_i$ **do**
      $P_{difY} = P_{difY} + d_{\mathcal{L}}(i,j)/k$;
      foreach $f \in F$ **do**
        $P^f_{difval} = P^f_{difval} + \delta(x_{if}, x_{jf})/k$;
        $P^f_{difY|\,difval} = P^f_{difY|\,difval} + d_{\mathcal{L}}(i,j) \cdot \delta(x_{if}, x_{jf})/k$;
      end
    end
  end
  foreach $f \in F$ **do**
    $W_f = \frac{P^f_{difY|\,difval} \cdot P^f_{difval}}{P_{difY}} - \frac{(1 - P^f_{difY|\,difval}) \cdot P^f_{difval}}{1 - P_{difY}}$
  end
  return W;
end

Independently, in [72] was proposed a similar method to RReliefF-ML. The authors prove the proposal on synthetic datasets and the FS process. The synthetic multi-label datasets are generated following two different strategies (*HyperCubes* and *HyperSpheres*) proposed in [73]. The equations to update the label and feature dissimilarities differ from our proposal. The dissimilarity values are multiplied by a weighted distance. On the other hand, the partial weights are combined in a different manner.

RReliefF-ML takes into account the label correlations by the calculus of the probabilities $P^f_{difY|\,difval}$ and P_{difY}. A high feature weight indicates that the feature has different values on instances with dissimilar sets of labels. It also indicates that the feature has similar values on instances with similar sets of labels.

The RReliefF-ML extension does not use a PTM for the estimation of the feature weights. RReliefF-ML retrieves only k-nearest neighbours for each sampling instance. The computational complexity of RReliefF-ML is $O(m \cdot n \cdot d)$.

4. Empirical study

In this section a description of the multi-label datasets, algorithms, statistical tests and other settings used in the experimental study is explained. The experimental results on different classifiers, datasets and the statistical analysis are showed. The empirical study was divided into two parts: a comparative study between the ReliefF extensions on the FW process and a comparison of the ReliefF extensions on the FS process.

Table 1
Statistics of the benchmark datasets, number of instances (n), number of features (d), number of labels (q), different subsets of labels (d_s), label cardinality (l_c) and label density (l_d). The datasets are ordered by their complexity calculated as $n \cdot d$.

Dataset	*Domain*	n	d	q	d_s	l_c	l_d
Flags	Image	194	19	7	54	3.392	0.485
Cal500	Music	502	68	174	502	26.044	0.150
Emotions	Music	593	72	6	27	1.869	0.311
Birds	Audio	645	260	19	133	1.014	0.053
Yeast	Biology	2417	103	14	198	4.237	0.303
Scene	Image	2407	294	6	15	1.074	0.179
Genbase	Biology	662	1186	27	32	1.252	0.046
Medical	Text	978	1449	45	94	1.245	0.028
Enron	Text	1702	1001	53	753	3.378	0.064
Corel5k	Image	5000	499	374	3175	3.522	0.009
Mediamill	Video	43,907	120	101	6555	4.376	0.043
Corel16k (10 samples)	Image	13,811	500	161	4937	2.867	0.018
Bibtex	Text	7395	1836	159	2856	2.402	0.015
TMC2007-500	Text	28,596	500	22	1341	2.16	0.098
Arts	Text	7484	23,146	26	599	1.654	0.064
Science	Text	6428	37,187	40	457	1.450	0.036
Business	Text	11,214	21,924	30	233	1.599	0.053
Health	Text	9250	30,605	32	335	1.644	0.051
Reference	Text	8027	39,679	33	275	1.174	0.035
Education	Text	12,030	27,534	33	511	1.463	0.044
Recreation	Text	12,828	30,324	22	530	1.429	0.065
Entertainment	Text	12,730	32,001	21	337	1.414	0.067
Computers	Text	12,444	34,096	33	428	1.507	0.046
Society	Text	14,512	31,802	27	1054	1.670	0.062
Social	Text	12,111	52,350	39	361	1.279	0.033

4.1. Multi-label datasets

In the experiments 34 real multi-label datasets were used[1], where the number of relevant, irrelevant and redundant features is unknown. Multi-label datasets with different scale and from different application domains were included to analyse the behaviour of the ReliefF extensions in datasets with diverse properties.

The datasets come from six domains. The dataset Birds [74] contains examples of multiple bird species for acoustic classification. Cal500 [16] contains pieces of music for semantic annotation. Emotions dataset [43] stores examples of songs according to the emotions that they evoke. Flags [75] stores examples about nations and their national flags. Scene [8] contains a series of patterns about kinds of landscapes. The Corel5k [12] and Corel16k [14] datasets contain Corel images. Mediamill [76] contains examples for the automatic detection of semantic concepts in videos. The Yeast [77] and Genbase [9] datasets come from the biological domain, including information about the function of genes and proteins. Medical [78] was used in the Medical Natural Language Processing Challenge in 2007. Enron [79] contains emails from 151 users. TMC2007-500 [80,81] stores examples of reports of aviation safety. Bibtex [82] contains bibtex examples for automatic tag suggestion. The other eleven datasets come from the Yahoo text collection [13].

Table 1 shows some statistics of the multi-label datasets. The values of the properties of the Corel16k dataset are averaged over all ten samples used. The label cardinality is the average number of labels per example. The label density is the label cardinality divided by the total number of labels. The label cardinality, label density and different subsets of labels are measures that represent the complexity of a multi-label dataset [3]. The datasets vary in size: from 194 up to 43,907 instances, from 19 up to 52,350 features, from 6 to 374 labels, from 15 to 6555 different subset of labels, from 1.014 to 26.044 label cardinality, and from 0.009 to 0.485 label density.

[1] All these datasets are available at http://mulan.sourceforge.net/datasets.html

4.2. Experimental setting

The multi-label ReliefF extensions and algorithms were implemented on MULAN [83]. MULAN is a Java library which contains several methods for MLL. For each possible combination of algorithm and dataset a stratified 10-fold cross validation strategy was used. The methods proposed in [84] were used to stratify the multi-label data. For each fold in the training phase, a ReliefF extension learned the weight vector on the training set. The whole training set was used to retrieve the k-nearest neighbours of a sampling instance. The best number of neighbours (k) for each classifier and ReliefF extension on each dataset was determined.

Several studies have proved that a larger number of sampling instances (m) results in a more reliable approximation in the feature estimation process [22,24]. For the sake of fairness, in the experiments the whole training set was used on the feature estimation process. Also, the values of p and s parameters for PPT-ReliefF were tuned by a cross validation on the training set, as recommended in [69].

Feature weighting setting. The performance of lazy algorithms can be significantly improved with the use of an appropriate weight vector. The aim is to find a weight vector W that allows the distance function to recover the k-nearest neighbours in the feature space, given that not all features have the same relevance in a dataset [19].

In the experiments four multi-label lazy algorithms were used. In [70] is proposed the Multi-Label k-Nearest Neighbour algorithm (MLkNN). MLkNN determines the label set of a query instance by the maximum a posteriori (MAP) principle. The BRkNN algorithm appears in [85]. BRkNN calculates the confidences of each label based on the label sets of the neighbour queries. BRkNN is conceptually equivalent to use the BR method in conjunction with the kNN algorithm. In [86] the Dependent Multi-Label k-Nearest Neighbour algorithm (DMLkNN) is proposed. DMLkNN defines a MAP rule that takes into account the number of all labels in the neighbourhood. The Multi-Label Weighted k-Nearest Neighbour Classifier (MLCWkNN) appears in [87]. MLCWkNN is based on the Bayes Theorem and it proposes an instance weighted kNN version.

Each lazy classifier was compared with six weighted lazy classifier versions. A weighted lazy classifier parameterize its distance function with a learned weight vector by a ReliefF extension. All the feature weights were scaled to $[0, 1]$ using the min–max normalisation [1].

Feature selection setting. The ReliefF extensions were compared into the FS process to improve the effectiveness of the multi-label learning algorithms. The BRkNN classifier was used as base-line algorithm, owing to its simplicity and high sensitivity to the presence of irrelevant and redundant features.

A learned weight vector by the ReliefF algorithm can be viewed as a feature ranking. A feature ranking is important to guide the search on FS process, especially where the search space to find the best feature subset is large [1]. Several methods have been proposed to evaluate a feature ranking on the FS process [1,25,88]. The procedure proposed in [25] was used, owing to its simplicity and computational complexity. It provides a heuristic for comparing several feature rankings.

In the experiments the 100-top features of a ranking R were selected for constructing a R' ranking. The process starts from the first feature on R' and continues with the next ranked attribute. Algorithm 4 shows the procedure used to evaluate a feature ranking.

Algorithm 4. Iterative procedure used to evaluate a feature ranking. The expression $X \succ Y$ means that the X value is better than the Y value.

input: $W \rightarrow$ weight vector
$\Phi \rightarrow$ multi−label learning algorithm
output: Best evaluation measure value
begin
  $R \leftarrow \texttt{rank}(W)$;
  $R' \leftarrow f\texttt{TopFeatures}(R)$;
  $\texttt{BestEvaluation} \leftarrow Empty$;
  $F_s \leftarrow \emptyset$;
  foreach $f \in R'$ **do**
    $\texttt{TempEvaluation} \leftarrow \texttt{Evaluate}(\Phi, F_s \cup f)$;
    if $\texttt{TempEvaluation} \succ \texttt{BestEvaluation}$ **then**
      $\texttt{BestEvaluation} \leftarrow \texttt{TempEvaluation}$;
      $F_s = F_s \cup f$;
    end
  end
  return BestEvaluation;
end

A R_x ranking is considered better than a R_y ranking if the baseline algorithm reaches a better performance with the feature subset determined from the R_x ranking. The ideal ranking would have all the relevant features placed at the top of the ranking [1,25,88].

4.3. Statistical tests

To analyse and validate the results, several nonparametric statistical tests were used, as proposed in [51–54,89]. The Friedman's test [90] was performed to evaluate whether there are significant difference in the results of the algorithms. If the Friedman's test indicated that the results were significantly different, the Bergmann–Hommel post-hoc test [91] was used to perform multiple comparisons among all methods. The Bergmann–Hommel's test is a more powerful test than other post-hoc tests, such as Nemenyi's test [92]. Nemenyi's test is conservative and many of the obvious differences could not be detected [52,54].

In the statistical analysis the Adjusted p-values (APVs) [93] were considered. APVs provide more information in a statistical analysis. APVs take into account the fact that multiple tests are conducted and can be compared directly with any significance level α [52,54]. In this work a significance level $\alpha = 0.05$ was considered.

The statistical information obtained from the Bergmann–Hommel's test was graphically illustrated as a graph. An edge $\Phi_1 \rightarrow \Phi_2$ represents that the method Φ_1 outperforms the method Φ_2. Each edge was labelled with the evaluation measures which Φ_1 outperformed the Φ_2 method. The APVs of the Bergamnn-Hommel's test were indicated between parentheses.

4.4. Results and discussion

The algorithms as standalone runnable files and all the results of the empirical study are available in order to facilitate the replicability of the experiments.[2] In this manuscript only a summary of the results appear. In all cases, the best results are highlighted in bold typeface in the tables.

[2] http://www.uco.es/grupos/kdis/kdiswiki/MLL/ReliefF

4.4.1. Feature weighting

Due to lack of space, the table of results on FW process for the BRkNN, DMLkNN and MLCWkNN classifiers are not included in this paper. Tables 2 and 3 show the results of the MLkNN classifier using the ReliefF extensions for the H_L and F_{1Ex} measures. The results for the A_P and R_L measures can be consulted in the available web page. In the tables the last two rows show the average rank (Rank) and the position in the ranking (Pos.) of each method.

Generally speaking, the results showed that the weighted lazy classifiers using PPT-ReliefF, ReliefF-ML and RReliefF-ML performed better than the original lazy methods (not weighted). On the other hand the lazy classifiers using PPT-ReliefF, ReliefF-ML and RReliefF-ML obtained the lower average ranks. There were significant results over the four evaluation measures considered on complex datasets, such as Corel5k, Corel16k collection, Mediamill, TMC2007-500, Bibtex and datasets that belong to Yahoo collection. In some cases the original lazy classifiers performed better than the weighted lazy classifiers using BR-ReliefF and MReliefF extensions.

A statistical analysis to detect significant differences on the performance among the non-weighted and weighted lazy classifiers was carried out. The Friedman's test rejected the null hypothesis in all cases analysed, considering a significance level $\alpha = 0.05$. The p-values returned by the Friedman's test can be consulted on the tables.

Afterwards, a Bergmann–Hommel's post-hoc test for all pairwise comparisons was carried out. The results of the Bergmann–Hommel's test for the MLkNN classifier are showed in Fig. 1.

From a statistical point of view, the MLkNN classifier using PPT-ReliefF, RReliefF-ML and ReliefF-ML extensions outperformed the MLkNN classifier using BR-ReliefF, MReliefF and LP-ReliefF on the four multi-label evaluation measures considered. A similar result was obtained over the BRkNN, DMLkNN and MLCWkNN classifiers.

The Bergmann–Hommel's test did not detect significant differences among PPT-ReliefF, ReliefF-ML and RReliefF-ML extensions (the three extension proposed in this work). However, PPT-ReliefF obtained the first position in 14 of 16 average rankings (16 rankings = four lazy methods $\times$ four evaluation measures) returned by the Friedman's test. RReliefF-ML obtained the first position in two average rankings, the second position in eleven and the third position in three rankings. ReliefF-ML obtained the second position in three and the third position in 13 rankings. The weighted lazy classifiers that used the MReliefF and BR-ReliefF extensions performed worse than the non-weighted original lazy algorithms.

For the H_L, F_{1Ex}, R_L and A_P measures, the weighted lazy classifiers that used the weight vector of PPT-ReliefF obtained the best results. Followed by those classifiers that used the weight vector from RReliefF-ML and ReliefF-ML.

The results showed that the PPT-ReliefF outperformed the LP-ReliefF extension. This result showed that the PPT technique reduces the scarcity of labels and over-fitting of data. PPT produced more balanced multi-class datasets than LP approach and reduced the complexity of the multi-label problems.

From the statistical analysis, we concluded that the three proposed ReliefF extensions learn the usefulness of each feature correctly. It also confirms the effectiveness of the ReliefF algorithm as a feature weighting method for a better multi-label lazy learning.

4.4.2. Feature selection

Tables 4 and 5 show the results of the BRkNN classifier using the whole feature space and the feature subsets determined from the ReliefF extensions for the H_L and F_{1Ex} evaluation measures. The results for the A_P and R_L measures can be consulted in the available

Table 2
$H_L(\downarrow)$ results for ML*k*NN on FW process. The Friedman's test rejects the null hypothesis with a p-value equal to 6.488E−7.

Dataset	ML*k*NN						
	–	BR-ReliefF	LP-ReliefF	MReliefF	PPT-ReliefF	ReliefF-ML	RReliefF-ML
Flags	0.268	0.274	0.273	0.292	0.276	0.268	**0.265**
Cal500	0.139	0.140	0.140	**0.138**	0.140	**0.138**	**0.138**
Emotions	0.209	0.205	0.193	0.198	0.190	**0.188**	0.191
Birds	0.051	0.055	0.051	0.052	0.049	0.049	**0.045**
Yeast	0.198	0.197	0.198	0.199	0.197	**0.196**	0.197
Scene	0.095	0.097	0.098	0.096	0.097	0.095	**0.093**
Genbase	0.005	0.006	0.004	0.006	**0.003**	0.004	**0.003**
Medical	0.019	0.023	0.017	0.019	0.017	0.017	**0.015**
Enron	0.051	0.062	0.051	0.062	**0.050**	0.053	0.051
Corel5k	**0.009**	0.010	**0.009**	0.010	**0.009**	**0.009**	**0.009**
Mediamill	**0.031**	**0.031**	**0.031**	**0.031**	**0.031**	**0.031**	**0.031**
Corel16k01	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**
Corel16k02	0.020	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**
Corel16k03	0.021	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**
Corel16k04	0.019	0.019	0.019	**0.018**	0.019	0.019	0.019
Corel16k05	0.020	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**
Corel16k06	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**	**0.019**
Corel16k07	0.018	**0.017**	0.018	0.018	**0.017**	0.018	**0.017**
Corel16k08	0.019	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**
Corel16k09	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**
Corel16k10	0.020	0.020	0.020	0.20	0.019	**0.017**	0.019
Bibtex	0.014	0.014	0.014	0.014	**0.013**	0.014	**0.013**
TMC2007-500	0.058	0.065	0.064	0.064	0.063	**0.056**	**0.056**
Arts	0.063	0.063	0.063	0.063	**0.062**	0.063	0.063
Science	0.036	0.036	0.036	0.036	**0.035**	**0.035**	0.036
Business	0.029	0.029	0.029	0.029	0.029	**0.028**	0.029
Health	0.049	0.051	**0.048**	0.052	**0.048**	0.049	**0.048**
Reference	0.035	0.036	0.035	**0.034**	**0.034**	**0.034**	**0.034**
Education	0.044	**0.043**	**0.043**	**0.043**	**0.043**	**0.043**	**0.043**
Recreation	0.063	0.063	**0.062**	**0.062**	**0.062**	**0.062**	0.063
Entertainment	0.063	0.062	0.062	0.062	**0.061**	0.062	0.062
Computers	0.040	0.043	0.040	0.045	**0.039**	**0.039**	0.040
Society	0.060	0.062	0.058	0.058	0.055	0.055	**0.054**
Social	0.030	0.030	0.030	0.030	**0.029**	**0.029**	0.030
Rank	4.956	5.029	4.235	4.706	3.000	3.029	3.044
Pos.	6	7	4	5	1	2	3

web page. In the tables the last two rows show the average rank (Rank) and the position in the ranking (Pos.) of each method.

Generally speaking, the results showed that the BR*k*NN classifier using the feature subsets determined from the feature rankings of PPT-ReliefF, ReliefF-ML and RReliefF-ML performed better than the BR*k*NN that used the whole feature space. On the other hand, BR*k*NN classifier obtained a lower average rank when it used the feature subsets determined from PPT-ReliefF, ReliefF-ML and RReliefF-ML.

The results showed that on simple multi-label datasets; such as Flags, Birds, Emotions, Cal500, Yeast, Genbase and Medical, the FS process yielded a considerable improvement over the effectiveness of the BR*k*NN classifier.

It is important to highlight that BR*k*NN using the feature subsets of PPT-ReliefF, ReliefF-ML and RReliefF-ML obtained the best results on the eleven complex multi-label datasets that come from Yahoo collection. Several of the multi-label datasets have a large number of features, e.g. Bibtex (1836 features), Medical (1449 features), Genbase (1186 features), Enron (1001 features) and Yahoo collection (from 21,924 to 52,350 features). The experimental results can be considered as formidable results, considering that only the 100-top features of the feature rankings were considered in the FS process. The BR*k*NN classifier using a small number of features (fewer or equal than 100 features) performed better than the BR*k*NN that used the whole feature space.

A statistical analysis to detect significant differences on the performance among the ReliefF extensions was carried out. The Friedman's test rejected the null hypothesis in all cases analysed considering a significance level $\alpha = 0.05$. The p-values returned by the Friedman's test can be consulted on the tables.

Afterwards, a Bergmann–Hommel's post-hoc test for all pairwise comparison was carried out. The results of the Bergmann–Hommel's test are displayed in Fig. 2.

From a statistical point of view, the BR*k*NN classifier using the feature subsets determined from PPT-ReliefF, RReliefF-ML and ReliefF-ML extensions outperformed the BR*k*NN classifier that used the whole feature space for the four multi-label evaluation measures considered. The BR*k*NN classifier using the three proposed ReliefF extensions performed better than the BR*k*NN classifier using the BR-ReliefF, LP-ReliefF and MReliefF extensions for the four multi-label evaluation measures.

The Bergmann–Hommel's test did not detect significant differences among PPT-ReliefF, ReliefF-ML and RReliefF-ML extensions. However, PPT-ReliefF obtained the first position in two of four average rankings (four rankings = one classifier $\times$ four evaluation measures) returned by the Friedman's test. RReliefF-ML obtained the first position in one average ranking and the second position in two rankings. ReliefF-ML obtained the third position in three rankings.

The RReliefF-ML extension obtained the best results for H_L measure, followed by PPT-ReliefF. ReliefF-ML performed better for F_{1Ex} measure, followed by PPT-ReliefF. PPT-ReliefF obtained the best results for the R_L and A_P measures, followed by RReliefF-ML.

From the statistical analysis, we concluded that the three proposed ReliefF extensions correctly determine the usefulness of each feature. Also, it confirms the effectiveness of the ReliefF algorithm as a feature selection method for a better multi-label learning.

Table 3
$F_{1Ex}(\uparrow)$ results for ML*k*NN on FW process. The Friedman's test rejects the null hypothesis with a *p*-value equal to 5.295E − 11.

Dataset	ML*k*NN						
	–	BR-ReliefF	LP-ReliefF	MReliefF	PPT-ReliefF	ReliefF-ML	RReliefF-ML
Flags	0.696	0.713	0.718	0.684	0.694	0.708	**0.720**
Cal500	0.326	0.328	0.331	0.308	0.331	0.332	**0.339**
Emotions	0.587	0.628	0.626	0.633	0.664	**0.676**	0.643
Birds	0.527	0.490	0.524	0.518	0.523	0.528	**0.569**
Yeast	0.611	0.609	**0.617**	0.611	0.609	0.611	**0.617**
Scene	0.682	0.669	0.658	0.656	**0.685**	0.656	0.663
Genbase	0.950	0.0953	0.977	0.0952	0.980	0.967	**0.981**
Medical	0.439	0.412	0.533	0.421	0.515	0.515	**0.543**
Enron	0.417	0.417	0.461	0.357	0.475	0.419	**0.482**
Corel5k	0.019	0.017	0.005	0.010	0.023	0.027	**0.035**
Mediamill	0.533	0.533	0.533	0.528	0.534	0.533	**0.537**
Corel16k01	0.013	0.010	0.007	0.010	0.013	0.011	**0.015**
Corel16k02	0.016	0.013	0.009	0.010	**0.033**	0.020	0.027
Corel16k03	0.012	0.012	0.009	0.015	**0.020**	0.015	0.012
Corel16k04	0.015	0.012	0.026	0.027	0.020	**0.028**	0.022
Corel16k05	0.015	0.018	0.006	0.013	0.019	**0.024**	0.018
Corel16k06	0.018	0.018	0.016	0.017	**0.034**	0.020	0.025
Corel16k07	0.017	0.020	0.011	0.018	**0.027**	0.020	0.025
Corel16k08	0.018	0.017	0.009	0.005	**0.024**	0.023	0.020
Corel16k09	0.005	0.006	0.009	0.010	**0.018**	0.013	0.016
Corel16k10	0.006	0.008	0.009	0.006	0.016	**0.018**	0.012
Bibtex	0.161	0.174	0.189	0.201	**0.226**	0.201	0.205
TMC2007-500	0.660	0.604	0.614	0.609	0.617	**0.680**	0.671
Arts	0.034	0.029	0.033	0.024	0.050	**0.055**	0.039
Science	0.015	0.010	0.017	0.016	0.018	0.022	**0.023**
Business	**0.737**	0.728	0.736	0.729	0.735	**0.737**	0.735
Health	0.363	**0.423**	0.347	0.421	0.387	0.365	0.376
Reference	0.236	0.221	0.131	0.219	0.241	**0.264**	0.244
Education	0.026	0.026	0.026	**0.032**	**0.032**	0.028	0.027
Recreation	0.058	0.060	0.059	**0.065**	0.064	0.060	0.056
Entertainment	0.108	0.104	0.113	0.107	**0.128**	0.116	0.124
Computers	0.369	0.409	0.432	0.378	**0.433**	0.427	0.429
Society	0.155	0.150	0.173	0.135	0.165	**0.178**	0.169
Social	0.263	0.223	0.310	0.286	0.285	**0.336**	0.321
Rank	4.941	5.323	4.691	5.265	2.618	2.706	2.456
Pos.	5	7	4	6	2	3	1

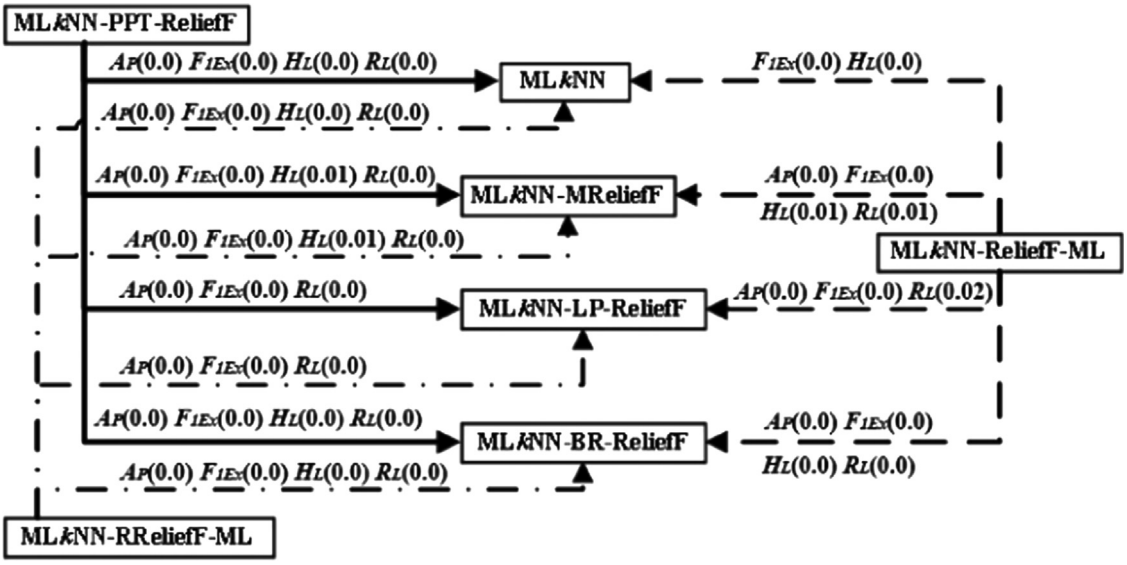


Fig. 1. Significant differences of the performance among ReliefF extensions on the ML*k*NN classifier according to the Bergmann–Hommel's test.

4.4.3. Discussion

In Table 6 a summary of the main characteristics of the state-of-the-art ReliefF extensions and of the three extensions proposed in this paper is shown. In the first column appears the time complexity of each ReliefF extension. The column named "Label correlation" specify whether the corresponding ReliefF extension handles the dependencies among labels or not. The column "Transformation" states the type of PTM that use each ReliefF

Table 4
$H_L(\downarrow)$ results for BR*k*NN on FS process. The Friedman's test rejects the null hypothesis with a p-value equal to 5.122E−11.

Dataset	BR*k*NN						
	–	BR-ReliefF	LP-ReliefF	MReliefF	PPT-ReliefF	ReliefF-ML	RReliefF-ML
Flags	0.271	0.250	**0.226**	0.237	0.228	0.228	0.232
Cal500	0.145	0.140	0.140	0.140	0.140	0.140	**0.139**
Emotions	0.197	0.182	0.178	**0.175**	0.176	0.179	0.178
Birds	0.049	**0.042**	0.044	0.045	0.043	0.046	**0.042**
Yeast	0.203	0.208	0.205	**0.200**	0.201	0.203	**0.200**
Scene	**0.108**	0.115	0.115	0.115	0.114	0.146	0.118
Genbase	0.003	0.004	0.003	0.004	0.003	0.002	**0.001**
Medical	0.021	0.027	**0.015**	0.024	**0.015**	0.026	**0.015**
Enron	0.058	0.053	0.050	0.052	**0.049**	0.058	**0.049**
Corel5k	0.010	**0.009**	0.010	0.010	**0.009**	**0.009**	**0.009**
Mediamill	0.032	0.033	0.032	0.032	0.032	**0.031**	**0.031**
Corel16k01	0.020	0.020	0.020	0.020	0.020	**0.019**	**0.019**
Corel16k02	0.020	0.019	0.019	0.019	0.019	**0.018**	0.019
Corel16k03	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**	**0.020**
Corel16k04	**0.019**	**0.019**	**0.019**	0.020	**0.019**	**0.019**	**0.019**
Corel16k05	0.019	0.019	0.019	0.019	0.018	**0.017**	0.018
Corel16k06	0.020	0.019	0.019	0.019	0.018	**0.017**	0.018
Corel16k07	0.018	0.018	0.018	0.018	**0.017**	**0.017**	**0.017**
Corel16k08	0.019	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**	**0.018**
Corel16k09	0.019	0.018	0.018	0.018	**0.017**	**0.017**	**0.017**
Corel16k10	0.021	0.020	0.020	0.020	0.019	**0.017**	0.019
Bibtex	0.015	0.014	0.014	0.014	**0.013**	**0.013**	0.014
TMC2007-500	**0.064**	0.066	0.065	**0.064**	**0.064**	0.066	**0.064**
Arts	0.063	0.062	0.063	0.062	0.062	0.062	**0.061**
Science	0.035	0.033	0.034	0.036	0.033	**0.032**	**0.032**
Business	0.029	0.029	0.030	0.032	**0.027**	**0.027**	0.029
Health	0.050	0.051	0.051	0.049	0.049	**0.047**	**0.047**
Reference	0.037	0.040	0.037	0.036	0.032	0.032	**0.031**
Education	0.044	0.045	0.046	0.042	0.037	**0.035**	0.036
Recreation	0.063	0.062	0.064	**0.058**	**0.058**	**0.058**	**0.058**
Entertainment	0.063	0.061	0.060	0.064	0.057	0.057	**0.056**
Computers	0.040	0.042	0.040	0.039	0.039	**0.036**	0.040
Society	0.058	**0.050**	0.054	0.051	0.049	**0.050**	0.051
Social	0.030	0.032	0.032	0.031	0.026	**0.025**	0.026
Rank	5.485	5.000	4.779	4.515	2.794	2.853	2.573
Pos.	7	6	5	4	2	3	1

extension. The last column describes some advantages and disadvantages of the ReliefF extensions.

Those ReliefF extensions that use BR and RPC as PTM are very expensive in multi-label datasets that have a large number of labels. The PPT-ReliefF, RReliefF-ML and ReliefF-ML have the same computational complexity and perform faster than MReliefF and BR-ReliefF extensions. However, the RReliefF-ML method is faster than ReliefF-ML and PPT-ReliefF, since RReliefF-ML does not use any PTM and retrieves only k-nearest neighbours for a sampling instance. On the other hand, it is important to highlight that RReliefF-ML is the most simplest proposed extension and it obtains very good results on the FW and FS processes.

The evidence suggests that the ReliefF extensions which use the Label Powerset family of methods (i.e. LP and PPT) as PTM perform better than those ReliefF extensions that use a PTM that converts the multi-label problem into several single-label problems (e.g. BR and RPC), not only we referred on computing time but on the efficacy to determine the feature weights. On the other hand, the ReliefF extensions which consider the label dependencies perform better than those extensions which do not consider the label dependencies.

In the case of the FW process, the PPT-ReliefF, ReliefF-ML and RReliefF-ML extensions improved the performance of the four lazy classifier on the four evaluation measures considered. However, the results of the weighted lazy classifiers using BR-ReliefF, LP-Relief and MReliefF vary according to the measure and dataset employed.

The evidence suggests that the learned weight vector in the training phase allows the distance function recover those nearest examples in the feature space that are associated with the major confidence set of labels for classifying a query instance. The proposed ReliefF extensions performed well for simple and complex multi-label datasets on the MLC and LR tasks. However, the results showed that there was a smaller increase of the performance in those multi-label datasets with a small label density and a big number of distinct label sets at the same time, e.g. the Corel5k and Corel16k collection.

In the case of the FS process, the results showed that the PPT-ReliefF outperformed the LP-ReliefF extension. This result confirmed that the PPT technique is significantly superior to the LP method. PPT permitted to reduce the complexity of the multi-label datasets without loss of effectiveness in machine learning.

The evidence suggested that the distributions of the relevant features on the f-top features of the rankings determined by PPT-ReliefF, ReliefF-ML and RReliefF-ML were better than the distributions of the relevant features on the three other ReliefF extensions. Moreover, the three proposed ReliefF extensions performed well on simple and complex multi-label datasets for the MLC and LR tasks on the FS process. According to the results, the proposed ReliefF extensions performed better on datasets that have a small label density.

Table 5
$F_{1Ex}(\uparrow)$ results for BRkNN on FS process. The Friedman's test rejects the null hypothesis with a p-value equal to 6.091E−11.

Dataset	BRkNN						
	–	BR-ReliefF	LP-ReliefF	MReliefF	PPT-ReliefF	ReliefF-ML	RReliefF-ML
Flags	0.675	0.728	**0.746**	0.724	0.730	0.730	0.729
Cal500	0.302	0.321	0.318	0.318	0.318	**0.325**	0.320
Emotions	0.584	0.624	0.631	0.600	0.621	**0.639**	0.633
Birds	0.523	0.584	0.582	0.572	0.589	0.542	**0.598**
Yeast	0.583	0.590	0.585	0.583	0.580	**0.593**	0.590
Scene	**0.551**	0.523	0.550	0.545	0.541	0.500	0.521
Genbase	0.976	0.800	0.982	0.705	**0.983**	**0.983**	0.900
Medical	0.335	0.142	0.506	0.237	**0.595**	0.400	0.560
Enron	0.267	0.388	0.400	0.333	**0.486**	0.347	0.410
Corel5k	0.004	0.018	0.003	**0.040**	0.025	0.016	0.021
Mediamill	0.527	0.528	0.518	0.523	0.527	0.529	**0.534**
Corel16k01	0.009	0.019	0.019	0.019	**0.020**	**0.020**	**0.020**
Corel16k02	0.034	0.038	0.040	0.035	0.042	**0.044**	0.041
Corel16k03	0.013	0.015	0.014	0.016	**0.024**	0.023	0.019
Corel16k04	0.030	0.038	0.036	0.035	0.039	**0.040**	0.038
Corel16k05	0.014	0.008	0.009	0.012	0.020	**0.027**	0.018
Corel16k06	0.048	0.050	0.055	0.061	0.090	**0.099**	0.079
Corel16k07	0.008	0.009	0.010	0.006	0.009	**0.024**	0.009
Corel16k08	0.028	0.020	0.020	0.017	**0.031**	0.030	0.022
Corel16k09	0.060	0.050	0.066	0.060	0.085	**0.099**	0.090
Corel16k10	0.038	0.040	0.040	0.055	**0.066**	0.060	0.065
Bibtex	0.070	0.104	0.127	0.160	**0.224**	0.129	0.149
TMC2007-500	0.596	0.578	0.567	0.585	**0.600**	0.593	0.590
Arts	0.028	0.044	0.038	0.055	0.066	**0.067**	0.060
Science	0.014	0.016	0.015	0.017	**0.020**	**0.020**	0.018
Business	0.726	0.716	0.718	0.687	**0.798**	0.766	0.792
Health	0.230	0.221	0.225	0.229	0.245	0.247	**0.256**
Reference	0.403	0.398	0.365	0.415	**0.428**	0.421	0.419
Education	0.039	0.032	0.039	0.041	0.085	**0.089**	0.078
Recreation	0.037	0.032	0.030	0.021	0.065	**0.098**	0.077
Entertainment	0.096	0.092	0.095	0.085	0.122	0.123	**0.125**
Computers	0.366	0.354	0.325	0.365	**0.421**	0.410	0.410
Society	0.152	0.150	0.150	0.148	0.169	**0.174**	0.145
Social	0.217	0.223	0.215	0.245	0.248	0.248	**0.261**
Rank	5.412	5.103	4.912	5.118	2.338	2.279	2.838
Pos.	7	5	4	6	2	1	3

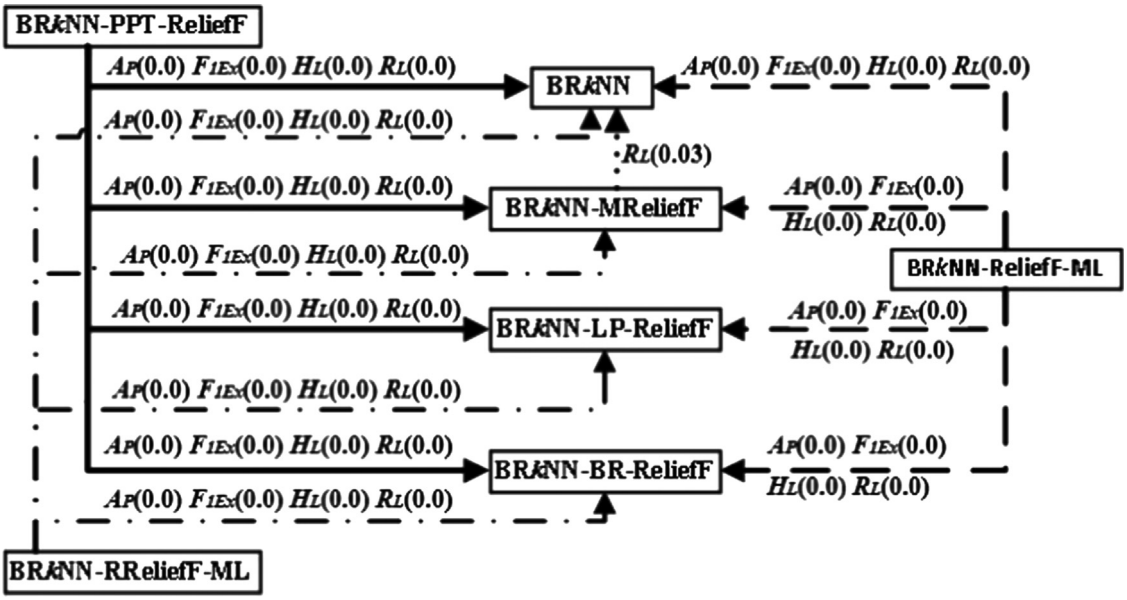


Fig. 2. Significant differences of the performance among ReliefF extensions on the BRkNN classifier according to the Bergmann–Hommel's test.

5. Conclusions

In this work, three scalable ReliefF extensions to multi-label learning called PPT-ReliefF, ReliefF-ML and RReliefF-ML have been presented. The PPT-ReliefF extension uses the PPT method to convert the original multi-label dataset into a new multi-class dataset. The ReliefF-ML extension can be considered as a generalisation of the classic ReliefF. On the other hand, the RReliefF-ML

Table 6
Summary of the characteristics of the ReliefF extensions.

Name	Time complexity	Label correlations	Transformation	Observations
BR-ReliefF	$O(q \cdot m \cdot n \cdot d)$	No	BR method	Poor performance on datasets with high number of labels.
MReliefF	$O(q^2 \cdot m \cdot n \cdot d)$	Partially	RPC method	Poor performance on datasets with high number of labels.
LP-ReliefF	$O(m \cdot n \cdot d)$	Yes	LPS method	Poor performance on datasets with high number of label sets.
ReliefF-ML	$O(m \cdot n \cdot d)$	Yes	None	Acceptable computing time.
PPT-ReliefF	$O(m \cdot n \cdot d)$	Yes	PPT method	Reduces the scarcity of labels and the over-fitting of data.
RReliefF-ML	$O(m \cdot n \cdot d)$	Yes	None	The most simplest ReliefF extension.

extension is based on the principles of the well-known ReliefF to regression problems. The three proposed extensions take into account the label dependencies and the issue of interacting features.

The proposed ReliefF extensions were extensively compared with previous ReliefF extensions. The experimental study was divided into two parts. In the first part, the ReliefF extensions were analysed on the FW process to improve the performance of the multi-label lazy algorithms. The statistical analysis showed that the three proposed ReliefF extensions outperformed previous ReliefF extensions, improving the performance of the multi-label lazy algorithms. The statistical tests showed that the weighted lazy algorithms that use the learned weight vector by PPT-ReliefF perform well, followed by those that use RReliefF-ML and ReliefF-ML. In the second part of the experiment, the ReliefF extensions were evaluated on the FS process. The baseline classifier using the feature subsets determined from the three proposed ReliefF extensions outperformed the classifier that uses the whole feature space and the feature subsets determined from previous ReliefF extension. The study shows that with a small number of features the baseline classifier obtains good results on complex multi-label datasets.

The PPT-ReliefF, RReliefF-ML and ReliefF-ML extensions performed well for the MLC and LR tasks on the FW and FS processes. These extensions are scalable on simple and complex multi-label datasets with different properties. The experimental study confirms the benefits of the ReliefF algorithm as feature engineering technique for a better MLL, the main motivation for the present work. We recommend that when new ReliefF extensions to MLL are proposed, they should be compared with PPT-ReliefF, RReliefF-Ml and ReliefF-ML extensions using several evaluation measures.

Future work will carry out a comparative study of how the proposed ReliefF extensions scale to other state-of-the-art multi-label feature estimation and feature selection algorithms. Furthermore, it would be important to examine the effectiveness of the proposal on synthetic multi-label datasets, where the number of relevant, irrelevant and redundant features is known.

References

[1] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, Morgan Kaufmann, San Francisco, United States of America, 2005.
[2] G. Tsoumakas, I. Katakis, Multi-label classification: an overview, Int. J. Data Warehous. Min. 3 (2007) 1–13.
[3] G. Tsoumakas, I. Katakis, I. Vlahavas, Data Mining and Knowledge Discovery Handbook, 2nd edition, Springer-Verlag, New York, United States of America, 2010, Mining Multi-label Data, pp. 667–686.
[4] G. Madjarov, D. Kocev, D. Gjorgjevikj, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (2012) 3084–3104.
[5] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: Working Notes of the AAAI-99 Workshop on Text Learning, 1999.
[6] T. Li, M. Ogihara, Detecting emotion in music, in: Proceedings of the International Symposium on Music Information Retrieval, Washington DC, USA, 2003, pp. 239–240.
[7] S. Yang, S. Kim, Y. Ro, Semantic home photo categorization, IEEE Trans. Circuits Syst. Video Technol. 17 (2007) 324–335.
[8] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (2004) 1757–1771.
[9] S. Diplarisa, G. Tsoumakas, P. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: Proceedings 10th Panhellenic Conference on Informatics (PCI 2005), 2005, pp. 448–456.
[10] M.L. Zhang, Z.H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (2006) 1338–1351.
[11] M.G. Larese, P.M. Granitto, J.C. Gómez, Spot defects detection in cDNA microarray images, Pattern Anal. Appli. 16 (2013) 307–319 Springer-Verlag.
[12] P. Duygulu, K. Barnard, N. de Freitas, D. Forsyth, Object recognition as machine translation: learning a lexicon for a fixed image vocabulary, in: Proceedings of the 7th European Conference on Computer Vision, 2002, pp. IV:97–112.
[13] N. Ueda, K. Saito, Parametric mixture models for multi-labeled text, in: Proceedings of the Neural Information Processing Systems 15 (NIPS 15)Kira, MIT Press, 2002, pp. 737–744.
[14] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, M.I. Jordan, Matching words and pictures, J. Mach. Learn. Res. 3 (2003) 1107–1135.
[15] M. Worring, C. Snoek, J. van Gemert, J.M. Geusebroek, A. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of the 14th Annual ACM International Conference on Multimedia, 2006, pp. 421–430.
[16] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Trans. Audio Speech Lang. Process. 16 (2) (2008) 467–476.
[17] R. Bellman, Adaptive Control Processes: A Guided Tour, Rand Corporation Research Studies, Princeton University Press, Princeton, New Jersey, United States of America, 1961.
[18] D.T. Larose, Discovering Knowledge in Data: An Introduction to Data Mining, Jhon Wiley & Sons, New Jersey, United States of America, 2005.
[19] D. Wettschereck, D.W. Aha, T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1997) 273–314.
[20] A. Abraham, E. Corchado, J. Corchado, Hybrid learning machines, Neurocomputing 72 (2009) 2729–2730.
[21] K. Kira, L. Rendell, A practical approach to feature selection, in: Proceedings of the International Conference on Machine Learning, Morgan Kaufmann, 1992, pp. 249–256.
[22] I. Kononenko, Estimating attributes: analysis and extension of ReliefF, in: Proceedings of the 7th European Conference in Machine Learning, ECML-94, Springer-Verlag, 1994, pp. 171–182.
[23] I. kononenko, E. Simec, M.R. Sikonja, Overcoming the myopia of inductive learning algorithms with ReliefF, Appl. Int. 7 (1997) 39–55.
[24] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (1–2) (2003) 23–69.
[25] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Heuristic search over a ranking for feature selection, in: Proceedings of IWANN 2005, Lectures Notes in Computer Science, vol. 3512, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 742–749.
[26] N. Spolar, E. Cherman, M. Monard, H. Lee, Filter approach feature selection methods to support multi-label learning based on ReliefF and Information Gain, in: Proceedings of the Advances in Artificial Intelligence—SBIA 2012, Lectures Notes in Computer Science, Springer, Berlin, Heidelberg, 2012, pp. 72–81.
[27] M. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 359–366.
[28] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the 20th International Conference on Machine Learning, ICML-00, 2003, pp. 856–863.
[29] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[30] J. Tang, S. Alelyani, H. Liu, Data Classification: Algorithms and Applications, CRC Press, Boca Raton, FL, United States of America, 2015, Feature Selection for Classification: A Review, pp. 37–64.
[31] R. Tibshirani, Regression shrinkage and selection via the LASSO, J. R. Stat. Soc. (1996) 267–288.
[32] H. Zou, The adaptive lasso and its oracle properties, J. Am. Stat. Assoc. 101 (476) (2006) 1418–1429.

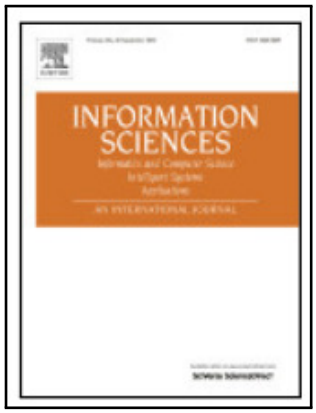
[33] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. 68 (1) (2006) 49–67.
[34] P. Zhao, B. Yu, On model selection consistency of lasso, J. Mach. Learn. Res. 7 (2006) 2541–2563.
[35] D. Kong, R. Fujimaki, J. Liu, F. Nie, C. Ding, Exclusive feature learning on arbitrary structures via l1,2-norm, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 1655–1663.
[36] Y. Zhou, R. Jin, S.C.H. Hoi, Exclusive lasso for multi-task feature selection, J. Mach. Learn. Res. 9 (2010) 988–995.
[37] P. Gong, J. Ye, C. Zhang, Robust multi-task feature learning, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, ACM, New York, USA, 2012, pp. 895–903.
[38] J. Zhou, J. Liu, V. Narayan, J. Ye, Modeling disease progression via fused sparse group lasso, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, ACM, New York, NY, USA, 2012, pp. 1095–1103.
[39] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Fast feature ranking algorithm, in: Proceedings of Knowledge-Based Intelligent Information and Engineering Systems, KES-2003, Springer Berlin, 2003, pp. 325–331.
[40] V. Jovanoski, N. Lavrac, Feature subset selection in association rules learning systems, in: Proceedings of Analysis, Warehousing and Mining the Data, 1999, pp. 74–77.
[41] B. Zupan, M. Bohanec, J. Demsar, I. Bratko, Learning by discovering concept hierarchies, Artif. Intell. 109 (1–2) (1999) 211–242.
[42] J.J. Liu, J.T.-Y. Kwok, An extended genetic rule induction algorithm, in: Proceedings of Congress of Evolutionary Computation, 2000, pp. 458–463.
[43] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas, Multilabel classification of music into emotions, in: Proceedings 2008 International Conference on Music Information Retrieval, ISMIR 2008, 2008, pp. 325–330.
[44] S. Dendamrongvit, P. Vateekul, M. Kubat, Irrelevant attributes and imbalanced classes in multi-label text-categorization domains, Intell. Data Anal. 15 (6) (2011) 843–859.
[45] G. Lastra, O. Luaces, J.R. Quevedo, A. Bahamonde, Graphical feature selection for multilabel classification tasks, in: Proceedings of the International Conference on Advances in Intelligent Data Analysis, 2011, pp. 246–257.
[46] D. Kong, C. Ding, H. Huang, H. Zhao, Multi-label ReliefF and F-statistic feature selections for image annotation, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2352–2359.
[47] N. Spolaor, E. Alvares, M. Carolina, H. Diana, A comparison of multi-label feature selection methods using the problem transformation approach, Electron. Notes Theor. Comput. Sci. 292 (2013) 135–151.
[48] N. Spolaor, E.A. Cherman, M.C. Monard, Using ReliefF for multi-label feature selection, in: Proceedings of the Conferencia Latinoamericana de Informática, Brazil, 2011, pp. 960–975.
[49] J. Read, A pruned problem transformation method for multi-label classification, in: Proceedings 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008), 2008, pp. 143–150.
[50] M. Robnik-Sikonja, I. Kononenko, An adaptation of Relief for attribute estimation in regression, in: Proceedings of the ICML-97, 1997, pp. 296–304.
[51] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[52] S. García, F. Herrera, An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[53] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[54] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.
[55] K. Brinker, J. Furnkranz, E. Hullermeier, A unified model for multilabel classification and ranking, in: Proceedings of the 17th European Conference on Artificial Intelligence, ECAI-06, 2006, pp. 489–493.
[56] R. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, Mach. Learn. 39 (2000) 135–168.
[57] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2004, 2004, pp. 22–30.
[58] I. Kononenko, M. Robnik-Sikonja, Non-Myopic feature quality evaluation with (R)ReliefF, Computational Methods of Feature Selection, Chapman & Hall/CRC, 2008, pp. 169–191.
[59] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection- theory and algorithms, in: Proceedings of the 21st International Conference on Machine Learning, 2004, pp. 43–50.
[60] Y. Sun, Iterative relief for feature weighting: algorithms, theories, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 1035–1051.
[61] Y. Sun, D. Wu, A RELIEF based feature extraction algorithm, in: Proceedings of the SIAM International Conference on DataMining, Atlanta, USA, 2008, pp. 188–195.
[62] U. Pompe, I. Kononenko, Linear space induction in first order logic with ReliefF, in: Mathematical and Statistical Methods in Artificial Intelligence, Springer Verlag, New York.
[63] M. Robnik-Sikonja, Experiments with cost-sensitive feature evaluation, in: Proceedings of the European Conference in Machine Learning, ECML-2003, 2003, pp. 325–336.
[64] M. Robnik-Sikonja, K. Vanhoof, Evaluation of ordinal attributes at value level, Data Min. Knowl. Discov. 14 (2007) 225–243.
[65] A.M. Qamar, E. Gaussier, RELIEF algorithm and similarity learning for *k*-NN, Int. J. Comput. Inf. Syst. Ind. Manag. Appl. 4 (2012) 445–458.
[66] A. Zafra, M. Pechenizkiy, S. Ventura, ReliefF-MI: an extension of ReliefF to multiple instance learning, Neurocomputing 75 (2012) 210–218.
[67] I. Slavkov, J. Karcheska, D. Kocev, S. Kalajdziski, S. Dzeroski, Extending ReliefF for hierarchical multi-label classification, in: Proceedings of the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases, ECML/PKDD-14, 2014.
[68] O. Reyes, C. Morell, S. Ventura, ReliefF-ML: an extension of ReliefF algorithm to multi-label learning, in: Proceedings of the CIARP 2013, vol. 8259, Part II, Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, Habana, Cuba, 2013, pp. 528–535.
[69] J. Read, B. Pfahringer, G. Holmes, Multi-label classification using ensembles of pruned sets, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 995–1000.
[70] M.L. Zhang, Z.H. Zhou, ML-*k*NN: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.
[71] J. Read, Scalable multi-label classification (Ph.D. thesis), University of Waikato, Hamilton, New Zeland, 2010.
[72] N. Spolaor, E.A. Cherman, M.C. Monard, H.D. Lee, ReliefF for multi-label feature selection, in: Proceedings of the International Brasilian Conference, IEEE, 2013.
[73] M.L. Zhang, J.M. Peña, V. Robles, Feature selection for multi-label naive Bayes classification, Inf. Sci. 179 (2009) 3218–3229.
[74] F. Briggs, et al., The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment, in: Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2013.
[75] E. Correa, A. Plastino, A. Freitas, A genetic algorithm for optimizing the label ordering in multi-label classifier chains, in: Proceedings of the ICTAI-2013, 2013.
[76] C. Snoek, M. Worring, J. van Gemert, J.-M. Geusebroek, A. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of ACM Multimedia, ACM, Santa Barbara, USA, 2006, pp. 421–430.
[77] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, Adv. Neural Inf. Process. Syst. 14.
[78] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Proceedings of the 20th European Conference on Machine Learning, 2009, pp. 254–269.
[79] B. Klimt, Y. Yang, The Enron corpus: a new dataset for email classification research, in: Proceedings of the 15th European Conference on Machine Learning, 2004, pp. 217–226.
[80] A. Srivastava, B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: Proceedings of the IEEE Aerospace Conference, 2005, pp. 55–63.
[81] G. Tsoumakas, I. Vlahavas, Random *k*-labelsets: an ensemble method for multilabel classification, in: Proceedings of the 18th European conference on Machine Learning, 2007, pp. 406–417.
[82] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Proceedings of the ECML/PKDD 2008 Discovery Challenge, Antwerp, Belgium, 2008.
[83] G. Tsoumakas, E. Spyromitros-Xioufi, J. Vilcek, I. Vlahavas, MULAN: a java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.
[84] K. Sechidis, G. Tsoumakas, I. Vlahavas, On the stratification of multi-label data, in: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases, vol. Part III, ECML/PKDD-11, Springer-Verlag, 2011, pp. 145–158.
[85] E. Spyromitros, G. Tsoumakas, I. Vlahavas, An empirical study of lazy multi-label classification algorithms, in: Proceedings of the SETN-2008, vol. 5138, Lectures Notes in Artificial Intelligence, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 401–406.
[86] Z. Younes, F. Abdallah, T. Denceux, Multi-label classification algorithm derived from *k*-nearest neighbor rule with label dependencies, in: Proceedings of the 16th European Signal Processing Conference, Lausanne, Switzerland, 2008, pp. 297–308.
[87] J. Xu, Multi-label weighted *k*-nearest neighbor classifier with adaptive weight estimation, in: Proceedings of the ICONIP 2011, Part II, vol. 7073, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2011, pp. 79–88.
[88] I. Slavkov, An evaluation method for feature rankings (Ph.D. thesis), Josef Stefan International Postgraduade School, 2012.
[89] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC-2005 Special Session on Real Parameter Optimization, J. Heurist., Springer 15 (2009) 617–644.
[90] M. Friedman, A comparison of alternative tests of significance for the problem of *m* rankings, Ann. Math. Stat. 11 (1940) 86–92.
[91] G. Bergmann, G. Hommel, Improvements of General Multiple Test Procedures for Redundant Systems of Hypotheses, Multiple Hypotheses Testing, Springer, Berlin, 1988, pp. 100–115.
[92] P.B. Nemenyi, Distribution-free multiple-comparisons (Ph.D. thesis), Pricenton University, 1963.
[93] S.P. Wright, Adjusted p-values for simultaneous inference, Biometrics 48 (1992) 1005–1013.

TITLE:

Effective lazy learning algorithm based on a data gravitation model for multi-label learning

AUTHORS:

O. Reyes, C. Morell, and S. Ventura



Information Sciences, *Volume 340-341, pp. 159-174, 2016*

RANKING:

Impact factor (JCR 2015): 3.364

Knowledge area:

Computer Science, Information Systems: 8/143

DOI: 10.1016/J.INS.2016.01.006

Information Sciences 340–341 (2016) 159–174

Contents lists available at ScienceDirect

Information Sciences

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

Effective lazy learning algorithm based on a data gravitation model for multi-label learning

Oscar Reyes[a], Carlos Morell[b], Sebastián Ventura[c,d,*]

[a] Department of Computer Science, University of Holguín, Holguín, Cuba
[b] Department of Computer Science, Universidad Central de Las Villas, Villa Clara, Cuba
[c] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
[d] Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 14 April 2015
Revised 22 December 2015
Accepted 1 January 2016
Available online 8 January 2016

Keywords:
Multi-label learning
Data gravitation model
Lazy learning
Multi-label classification
Label ranking

ABSTRACT

In the last decade, an increasing number of real-world problems surrounding multi-label data have appeared, and multi-label learning has become an important area of research. The data gravitation model is an approach that applies the principles of the universal law of gravitation to resolve machine learning problems. One advantage of the data gravitation model, compared with other techniques, is that it is based on simple principles with high performance levels. This paper presents a multi-label lazy algorithm based on a data gravitation model, named MLDGC. MLDGC directly handles multi-label data, and considers each instance as an atomic data particle. The proposed multi-label lazy algorithm was evaluated and compared to several state-of-the-art multi-label lazy methods on 34 datasets. The results showed that our proposal outperformed state-of-the-art lazy methods. The experimental results were validated using non-parametric statistical tests, confirming the effectiveness of this data gravitation model for multi-label lazy learning.

1. Introduction

The study of problems where examples are simultaneously associated with a set of labels has received special attention. Problems where this type of data appears are known as multi-label problems. Particular problems involving multi-label data include text categorization [27,35,40,48], emotions evoked by music [32], semantic annotation of images [1,11,65], classification of music [54] and videos [3,26], classification of protein function [14,37] and gene function [8,31,69], acoustic classification [4], chemical data analysis [56] and many more.

Multi-label learning is with a form of learning method that deals with a model which correctly generalizes unseen multi-label data [21,51]. Two tasks have been studied concerning the question of multi-labels: multi-label classification and label ranking. Multi-label classification divides a set of labels into relevant and irrelevant sets, whereas the label ranking task establishes an order of the labels for a given test instance [34,51].

For more than a decade, a considerable number of multi-label learning algorithms have been proposed. These multi-label algorithms can be divided into problem transformation methods and algorithm adaptation methods [21]. The former transform multi-label problems into one or more single-label problems, in order for classic learning algorithms to be used.

* Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain. Tel.:+34 957212218; fax:+34 957218630.
E-mail addresses: ogreyesp@gmail.com (O. Reyes), cmorellp@uclv.edu.cu (C. Morell), sventura@uco.es (S. Ventura).

http://dx.doi.org/10.1016/j.ins.2016.01.006
0020-0255/

On the other hand, the algorithm adaptation methods directly handle the multi-label data. To date, several lazy algorithms have been proposed for multi-label learning. The most significant works related to multi-label lazy learning appeared in the following literature: [6,10,24,33,47,62,66–68,70].

These previous works represent an important contribution to multi-label lazy learning studies. These approaches have only been tested on a handful of multi-label datasets, commonly 2 or 3 simple datasets, except in cases [6,70] and [10], where the proposed methods were tested on 13, 8 and 7 datasets, respectively. These lazy methods are nevertheless commonly compared against the ML*k*NN algorithm [70], which, to the best of our knowledge, was the first multi-label lazy algorithm proposed in the literature. However, these methods are not compared with other state-of-the-art multi-label lazy algorithms to determine which lazy method provides the best performance levels.

In this work, an effective lazy algorithm for multi-label learning is proposed, known as the Multi-label Data Gravitation Classification (MLDGC). MLDGC is based on the data gravitation model [38]. This latter is an approach that applies the principles of the universal law of gravitation to resolve machine learning problems. MLDGC considers each training instance as an atomic data particle. MLDGC introduces the Neighborhood-based Gravitation Coefficient, which is used instead of the particles masses in the calculation of gravitational force. MLDGC directly handles the multi-label data, and has an acceptable level of computational complexity.

To the best of our knowledge, this paper presents the first attempt to apply the data gravitation model for multi-label learning. It also serves to compare the most significant multi-label lazy methods using a sizeable number of datasets and two multi-label learning tasks (multi-label classification and label ranking tasks) and several evaluation measures. The main motivation of this present work is to examine the benefits of the data gravitation model for multi-label lazy learning.

The experiments were carried out on 34 multi-label datasets, considering different problem domains and properties. Several multi-label evaluation measures were used to analyze the various viewpoints. The experimental stage showed the effectiveness of our proposal, obtaining significantly better results than state-of-the-art multi-label lazy methods. The experimental study included statistical analysis based on non-parametric tests as proposed in [12,19,20].

This paper is arranged as follows: Section 2 describes the multi-label learning paradigm, the most relevant multi-label lazy algorithms and the data gravitation based algorithms that have appeared in the literature; Section 3 presents the basis of the MLDGC algorithm; Section 4 describes the experimental set-up and analyzes the experimental results. Finally, Section 5 provides some concluding remarks.

2. Preliminaries

In this section, the general definition of the multi-label learning paradigm is presented. The state-of-the-art multi-label lazy algorithms are described. A general background describing the data gravitational approach and the most relevant related works that have appeared in the literature, highlighting their advantages and disadvantages, are also shown.

2.1. Multi-label learning paradigm

Unlike single-label problems where the examples are associated with a single class, in multi-label problems the examples are classified altogether into a set of labels. The binary and multi-class classification problems can be considered specific cases of the multi-label learning problem. The main goal of multi-label learning is to allow us to form a predictive model to learn from multi-label data. Given a test instance, if the predictive model is capable of dividing the possible labels into relevant and irrelevant label sets, it is said that the model resolves the multi-label classification task. On the other hand, if the model is capable of ranking labels according to their relevance for a given test instance, then it is said that the model resolves the label ranking task [21]. The generalization of the multi-label classification and label ranking tasks is known as multi-label ranking [5], and its primary objective is to develop a model that resolves both tasks at the same time.

The multi-label data are defined on a feature space $\mathcal{F}$ with a cardinality equal to d (number of features) and a label space $\mathcal{L}$ with a cardinality equal to q (number of labels). A multi-label instance can be viewed as a tuple $\langle \mathbf{X_i}, Y_i \rangle$, where $\mathbf{X_i}$ is the feature vector of the instance i, and Y_i is the set of labels of the instance i.

To date, several multi-label learning algorithms have been proposed. These multi-label algorithms can be divided into two main categories [21]: problem transformation methods and algorithm adaptation methods. The problem transformation methods transform multi-label datasets into one or more single-label datasets. Then, for each transformed dataset, a single-label classifier is executed, and an aggregation strategy is performed. The algorithm adaptation category regroups algorithms that are designed to directly handle multi-label data. To date, several lazy algorithms have been proposed for multi-label learning. Section 2.2 briefly describes the most relevant multi-label lazy algorithms that have appeared in the literature.

2.2. Multi-label lazy algorithms

To the best of our knowledge, the first lazy algorithm for multi-label learning appeared in case [70]. The authors proposed the Multi-label *k*NN (ML*k*NN) algorithm which is based on a Bayesian approach. The prior probabilities and conditional probabilities of each label are computed. This process takes into account the information contained within the label sets of the *k*-nearest neighbors from the training set for each instance. The ML*k*NN algorithm determines the label set for a test instance using the Maximum A Posteriori (MAP) principle, based on the prior and conditional probabilities of each label.

The authors proved the effectiveness of ML*k*NN across 13 datasets using 5 evaluation measures. MLkNN was compared to 3 model-based multi-label algorithms.

A case-based multi-label ranking algorithm was presented in study [6]. The authors proposed a case-based framework, viewing the multi-label ranking generalization as a special case of rank aggregation with ties. The complexity of the method is determined by the aggregation stage, which is dependent on the complexity of the target space. Generally speaking, the aggregation stage can be carried out in a highly efficient manner. The authors proved the effectiveness of the method over 8 datasets using 1 evaluation measure. It was also compared to 4 model-based algorithms. In the following, we refer to this method as *k*NNMLR.

In study [47], the BR*k*NN method was proposed. BR*k*NN is equivalent to using the Binary Relevance [51] approach with *k*NN as a base classifier. It determines the *k*-nearest neighbors of a test instance, and calculates the confidences of each label based on the label sets of the neighbor instances. The authors also analyzed two extensions, BR*k*NN-α and BR*k*NN-β. BRkNN-α predicts the top ranked label where an empty label set is predicted, whereas the BR*k*NN-β extension predicts the top *b* ranked labels based on the size of the label sets of the neighbor instances. In addition, the authors proposed the LP*k*NN algorithm, which transforms the multi-label dataset into a new multi-class dataset and uses the classic *k*NN method to classify the test instances. LP*k*NN converts the multi-label dataset into a multi-class dataset by means of the Label Power Set [50] approach, which considers each subset of labels as a unique class of the new multi-class dataset. LP*k*NN can fail in multi-label datasets which have a high number of distinct label sets; as the Label Power Set approach generates imbalanced multi-class datasets. The authors tested BR*k*NN, BR*k*NN-α, BR*k*NN-β and LP*k*NN algorithms in 3 datasets using 3 evaluation measures, and they were compared to the ML*k*NN algorithm.

In study [66], the Dependent Multi-label *k*NN (DML*k*NN) algorithm was presented. This is a generalization of the ML*k*NN algorithm. DML*k*NN determines the label set of a test instance using the MAP principle, but unlike ML*k*NN, the MAP rule defined takes into account the number of all labels in the neighborhood. DML*k*NN was proved on 2 datasets using 5 evaluation measures; it was also compared to the ML*k*NN algorithm.

An algorithm which combines instance-based learning and logistic regression, known as IBLRML, was presented in case [10]. IBLRML creates new training data with label information as features. For every label created, it trains a logistic regression classifier. The authors proved the effectiveness of this method on 7 datasets using 5 evaluation measures. IBLRML was compared to 5 multi-label algorithms, including ML*k*NN.

The Fuzzy Veristic *k*NN (FV*k*NN) algorithm was presented in study [68]. FV*k*NN uses a fuzzy *k*NN rule based on the theory of veristic variables [63]. In the training phase, the label sets of the training instances are converted to fuzzy sets. The labels are considered as veristic variables. Given a test instance, the veristic statements corresponding to all pieces of knowledge are combined, and FV*k*NN predicts that the test instance belongs to a certain label if the degree of membership is greater than a given threshold. The authors proved the effectiveness of the approach in 3 simple multi-label datasets using 4 evaluation measures. FV*k*NN was compared to the ML*k*NN algorithm.

In study [67], a lazy algorithm known as the Evidential Multi-label *k*NN (EML*k*NN) algorithm, was presented. EML*k*NN generalizes Dempster–Shafer's single-label evidence-theory [13] to handle multi-label data. The label set of a test instance is determined on the label sets of the *k*-nearest neighbors, where each neighbor is considered a piece of evidence. The computational complexity exponentially grows on multi-label datasets that have a high cardinality on the label space, as the number of members in the frame of discernment is 2^q, where q is the number of labels. EML*k*NN was tested on 2 simple multi-label datasets using 4 evaluation measures. It was also compared to the ML*k*NN algorithm.

In case [33], the Mr.*k*NN algorithm was proposed. This method computes a soft relevance for each instance by means of an adaptation of the fuzzy c-means clustering algorithm, in order to determine the grade of membership in which an instance belongs to a particular label. Mr.*k*NN was solely designed for numerical attributes, as the calculation of relevance scores and the process of searching for the best value for the distance function parameters, are also highly expensive. Therefore, the application of this method is difficult, practically speaking, for large-scale multi-label datasets. Mr.*k*NN was proved on 3 simple multi-label datasets using 4 evaluation measures. It was compared to the ML*k*NN algorithm.

In study [62], the MLCW*k*NN algorithm was proposed. MLCW*k*NN is an instance-weighted *k*NN version for multi-label learning based on the Bayes Theorem. Given a test instance, MLCW*k*NN attributes more impact to the near instances than to the far ones. To determine the weights for each neighbor, an adaptive estimation process based on a quadratic programming method is performed. An experimental study on 2 simple datasets using 5 evaluation measures illustrated the effectiveness of the approach. MLCW*k*NN was compared to the ML*k*NN and IBLRML algorithms.

In case [24], the Fuzzy Similarity-based *k*NN (FS*k*NN) algorithm was proposed. FS*k*NN uses fuzzy similarity measures and the *k*NN approach to perform multi-label text classification. FS*k*NN tries to reduce the computational complexity of finding the *k*-nearest neighbors for a certain instance. It groups the training instances into clusters, so only the clusters whose similarities to a query exceed a predefined threshold are used to calculate the nearest neighbors. The label set of a test instance is determined using the MAP estimate. FS*k*NN was tested on 3 multi-label text datasets using 4 evaluation measures. It was also compared to the ML*k*NN algorithm and 2 other model-based algorithms.

2.3. Classification based on data gravitation model

The data gravitation model is a machine learning approach that applies the principles of the universal law of gravitation formulated by Newton in 1687, for the resolution of machine learning problems. Several learning methods inspired by

physical gravitation have been proposed. To the best of our knowledge, in case [61] the first learning algorithm appeared for performing cluster analysis based on data gravitation. In study [22], a clustering algorithm which considers every instance as an object in the feature space was proposed. The objects are moved by gravitational law and Newton's second law. Later, in study [17], a dynamic clustering algorithm based on the universal law of gravitation was proposed, taking into account the global and local information of the data.

As for classification tasks, several data gravitation-based algorithms have been proposed. In case [57], came an improvement of the *k*NN classifier using simulated gravitational collapse, generating prototypes from the original dataset. In study [64], the Data Gravitation-based Classifier (DGC) was proposed, and applied for abnormal network intrusion detection. In case [25], a new classification algorithm was proposed, but instead of considering data gravity as a scalar value, the authors suggested handling data gravity as a vector. This is a non-linear classification technique.

One of the most complete works concerning data gravitation classification was presented in study [38]. It is a more elaborate version of the DGC algorithm that first appeared in study [64]. The authors aimed to construct data particles from the original dataset. A data particle is data that also has a data mass. A data particle is formed from several instances which have some form of relationship. A data particle has a data centroid and a data mass. The data centroid is computed from the feature vectors of the instances that form the particle. The data mass is the number of instances that form the particle. The authors proposed a method based on the Maximum Distance Principle for data particle creation. It is an iterative method, whereby an instance is selected and its feature vector is set as the centroid of a new data particle. The instances whose distance from the centroid falls below a predefined threshold are combined to re-compute the centroid of the particle. This process is repeated until all other non-selected instances whose distance from the centroid falls below the threshold are eliminated. The process ends when there are no more instances to process.

Given a test instance, the gravitational force of each data particle to the instance is computed. The gravitational field of the data for each class is calculated according to the superposition principle. The test instance is classified according to the class with the highest gravitational field. The authors proved the effectiveness of the DGC algorithm in 12 single-label datasets. However, they stated that one of the major drawbacks of the DGC algorithm is that it severely suffers faced with imbalanced data. The DGC algorithm fails in the classification of minority class instances as the gravitation towards a certain class is extremely strong or extremely weak.

The threshold distance used to create a data particle is problem-dependent. Finding an optimal threshold value on large-scale datasets is computationally expensive. A large threshold value implies that the resulting data particles have a large mass; therefore the particles will have a greater impact on the gravitational force that acts over a test instance. The method proposed in case [38] for creating data particles is dependent on the initial selection of the instances used to compute the particles. Consequently, the effectiveness of the DGC algorithm depends on the initial instances selected to compute the particles and the threshold value used.

In addition, the calculation of the data particle centroid provokes a loss of information concerning the shape of the instance cloud, especially in instance clouds which possess an irregular shape. In dense regions that are over-represented by examples of a certain class, the resulting particles will have a large mass which could have a negative impact on the performance of the algorithm. This is due to their gravitational forces, which will have a great impact on the gravitational field of the class to which they belong. On the other hand, in sparse regions, the resulting particles will have a small mass with a correspondingly small influence on the gravitational field of the class to which they belong.

In case [7], the DGC+ algorithm was proposed, as it avoids the aforementioned problems of the data particle creation process by considering each instance as an atomic data particle. An atomic data particle has a mass equal to 1, and its centroid is the feature vector of the instance. Each instance is considered as an atomic data particle, and an accurate local classification is provided. It also provides good generalization where there are no closer training instances. The authors proved the usefulness of the DGC+ method on 35 standard datasets and 44 imbalanced datasets.

In study [45], the Gravitation-based Classification (GBC) algorithm appeared, based on the gravitational potential energy between particles. Its objective is to find the equilibrium condition of the classifier. GBC demonstrated a classification approach that is highly robust to noise. However, GBC is not suitable for classification of large-scale datasets, owing to its computational complexity in the training phase. GBC was tested on 12 real-world datasets.

In case [58], the Cognitive Gravitation Model (CGM) algorithm was presented. CGM is based on gravitational and cognitive laws, where the self-information of each instance is used instead of the mass of the particles. CGM considers each instance as an atomic data particle, as does the DGC+ algorithm. It estimates the self-information of each instance through density functions. A query instance is classified according to the class with the highest cognitive gravitation. The authors proved the effectiveness of this approach in 2 artificial datasets, 8 real datasets and 2 face images databases.

Recently, in study [39], the IDGC algorithm – an improvement on the classic DGC algorithm – was proposed. It attempts to resolve the problems of the DGC algorithm on imbalanced datasets. The class imbalance information is handled through a coefficient called the Amplified Gravitation Coefficient, which strengthens and weakens the gravitational field of the minority and majority classes, respectively. IDGC follows the same principle as the DGC algorithm for creating artificial data particles. The authors proved the effectiveness of the approach in 44 binary class and 15 multi-class imbalanced datasets.

Note that all the aforementioned algorithms based on data gravitation models have been designed to work on single-label learning problems. To the best of our knowledge, this work is the first attempt to adapt the data gravitation model to a multi-label learning context.

3. Multi-label lazy algorithm based on a data gravitation model

The particle creation process on multi-label data is a more complex task, owing to instances belonging to several labels at the same time. In addition to the drawbacks that the particle creation process presents for single-label data, there are other challenges associated with multi-label data.

- If the label set of each instance is considered as one class of a multi-class dataset, as is the case for the Label Power Set [50] transformation method, an imbalanced dataset can be created, where some classes overcome other classes in terms of number of particles.
- A simple idea is to decompose the multi-label problem into several binary classification problems, e.g. by using the Binary Relevance [51] or Ranking by Pairwise Comparison [23] transformation methods. However, when a multi-label dataset is analyzed for each label independently (or pairs of label), we can see that in most cases there is a non-uniform distribution of instances per label [9].
- Another simple idea is to take a threshold distance and not to take into account the label set of each instance. To form a particle, an instance is selected and all instances whose distance falls below a threshold value are used to create the particle. However, several instances that form the particle may have dissimilar label sets. Consequently, a lot of information is lost, unless a sophisticated aggregation method is proposed to handle the multi-label information in the particles.

As a consequence of the above situation, in this work we consider each instance of a multi-label dataset as an atomic data particle, i.e. the mass of the particle is equal to 1, as in the DGC+ [7] and CGM [58] algorithms for single-label data. In taking the multi-label information into account, a new concept is introduced, namely the Neighborhood-based Gravitation Coefficient, and this concept is used instead of the mass of the particle.

Before defining the Neighborhood-based Gravitation Coefficient (NGC) of a particle, we must first define certain notations and distance measures. A multi-label instance i is transformed into an atomic data particle which is represented as a 3-tuple $\langle \mathbf{X_i}, Y_i, g_i \rangle$. $\mathbf{X_i}$ is the feature vector of the particle i, it is the centroid of the particle as the mass of the particle is equal to 1. Y_i is the label set of the particle i. The component g_i is the NGC value of the particle i.

In the following, we refer to the term "atomic data particle" simply as "particle", due to the fact that all the particles in our problem formulation are atomic. Given two particles i and j, the distance between their centroids is calculated as follows:

$$d_{\mathcal{F}}(i,j) = \sqrt{\sum_{\forall f \in \mathcal{F}} \delta(x_{if}, x_{jf})^2} \tag{1}$$

$$\delta(x_{if}, x_{jf}) = \begin{cases} 1 & discrete,\ x_{if} \neq x_{jf} \\ 0 & discrete,\ x_{if} = x_{jf} \\ \frac{|x_{if} - x_{jf}|}{max(f) - min(f)} & continuous \end{cases} \tag{2}$$

where x_{if} and x_{jf} represent the value of the fth feature for particles i and j, respectively, and $\mathcal{F}$ is the feature space. The $d_{\mathcal{F}}$ function represents the HEOM (Heterogeneous Euclidean Overlap Metric) distance [59], although other distance measures can be used to compute the distance between two particles. The $\delta(x_{if}, x_{jf})$ function measures the difference in the fth feature, taking into account the type of data that are stored. The $max(f)$ and $min(f)$ functions return the maximum value and minimum value of the fth feature, respectively.

Based on $d_{\mathcal{F}}$ function, the set of the k-nearest neighbors of a particle i is defined as follows:

$$N_i = (i_1, i_2, \ldots, i_k) \mid d_{\mathcal{F}}(i, i_1) < d_{\mathcal{F}}(i, i_2) < \cdots < d_{\mathcal{F}}(i, i_k) \tag{3}$$

Given two particles i and j, the distance between the sets of labels of i and j is calculated by the Hamming distance notion (see Eq. 4). The distance $d_{\mathcal{L}}$ represents a measure of how much the label sets of two particles differ. A smaller value of $d_{\mathcal{L}}$ represents a major similarity in the classification of these particles

$$d_{\mathcal{L}}(i,j) = \frac{|\, Y_i \triangle Y_j \,|}{q} \tag{4}$$

where $\triangle$ is the symmetrical difference between two sets, and q is the number of labels that exist in the label space $\mathcal{L}$.

NGC is a coefficient that strengthens or weakens the gravitational force that a particle has over a test instance. Given a particle i, its NGC value (g_i) is computed as follows:

$$g_i = d_i^{w_i} \tag{5}$$

where d_i and w_i are the neighborhood-density and neighborhood-weight of the particle i, respectively.

The neighborhood-density (d_i) represents the distribution of the particles in the neighborhood of the particle i. A particle with a high density means that in its neighborhood the particles which have similar label sets to i are closer than those with

dissimilar label sets. The neighborhood-density of the particle i is computed by means of Eq. (6). The larger neighborhood-density value corresponds to the case when all the particles in the neighborhood have the same label set as particle i and they are near in the feature space

$$d_i = 1 + \sum_{j \in N_i} \frac{1 - d_{\mathcal{L}}(i,j)}{d_{\mathcal{F}}(i,j)} \tag{6}$$

The neighborhood-weight (w_i) represents the usefulness of the neighborhood of the particle i. A particle i with a high neighborhood-weight means that the probability of having particles in the neighborhood with dissimilar label sets is lower than the probability of having particles in the neighborhood with similar label sets. To compute the neighborhood-weight, in this work we were inspired by the estimation formula that uses the well-known extension of the ReliefF algorithm for regression problems [29,30,41,42]. From a probabilistic point of view, the estimation of the neighborhood-weight of a particle i is computed as follows:

$$w_i = P^i_{sim\mathcal{F}|simY} - P^i_{sim\mathcal{F}|disY} \tag{7}$$

where $P^i_{sim\mathcal{F}|simY}$ is the probability that the nearest particles are close in the feature space given that they have similar label sets. $P^i_{sim\mathcal{F}|disY}$ is the probability that the nearest particles are close in the feature space given that they have dissimilar label sets. Using the Bayes Rule Eq. (7) is transformed into

$$w_i = \frac{P^i_{simY|sim\mathcal{F}} P^i_{sim\mathcal{F}}}{P^i_{simY}} - \frac{(1 - P^i_{simY|sim\mathcal{F}}) P^i_{sim\mathcal{F}}}{1 - P^i_{simY}} \tag{8}$$

Eq. (8) can be transformed, so that it contains the probability that the nearest particles belong to different label sets provided that they are far in the feature space

$$w_i = \frac{P^i_{disY|dis\mathcal{F}} P^i_{dis\mathcal{F}}}{P^i_{disY}} - \frac{(1 - P^i_{disY|dis\mathcal{F}}) P^i_{dis\mathcal{F}}}{1 - P^i_{disY}} \tag{9}$$

where $P^i_{disY|dis\mathcal{F}}$ is the probability that the nearest particles have dissimilar label sets given that they are far in the feature space. $P^i_{dis\mathcal{F}}$ is the prior probability that the nearest particles are far in the feature space. P^i_{disY} is the prior probability that the nearest particles belong to dissimilar label sets. Each neighborhood-weight value must be normalized in the [0, 1] range before computing the corresponding NGC value.

The prior probability that nearest particles to i belong to different set of labels is computed as follows:

$$P^i_{disY} = \frac{\sum_{j \in N_i} d_{\mathcal{L}}(i,j)}{k} \tag{10}$$

where N_i represents the set of nearest neighbors of the particle i and k represents the number of nearest neighbors considered. The prior probability that the nearest particles to i are far in the feature space is computed as follows:

$$P^i_{dis\mathcal{F}} = \frac{\sum_{j \in N_i} d_{\mathcal{F}}(i,j)}{k} \tag{11}$$

The prior probability that nearest particles to i have dissimilar label sets given that they are far in the feature space is computed as follows:

$$P^i_{disY|dis\mathcal{F}} = \frac{\sum_{j \in N_i} d_{\mathcal{L}}(i,j) \cdot d_{\mathcal{F}}(i,j)}{k} \tag{12}$$

Note that the neighborhood-density and neighborhood-weight are similar because they consider the feature and label set based distances. However, neighborhood-density is oriented to the distribution of label sets around the particle i, whereas the neighborhood-weight is more focused on the probabilities of encountering particles with similar label sets in the neighborhood.

Once all training multi-label instances are converted in to particles, a test instance is classified as follows:

Given a test instance i, the k-nearest particles are retrieved. The gravitational force between the test instance i and the jth particle is computed as follows:

$$f(i,j) = \frac{g_j}{d_{\mathcal{F}}(i,j)^2} \tag{13}$$

Note that the test instance is considered as an atomic data particle, therefore its mass is equal to 1. The classic formula for gravitational force $f(i,j) = G\frac{m_i m_j}{r^2}$ is modified, where m_i and m_j are the mass of the objects i and j, respectively. G is the gravitational constant and r is the distance between the two objects. In our case, $m_i = m_j = 1$, where the distance between the particles is calculated by the $d_{\mathcal{F}}$ function and the gravitational constant G is replaced by the coefficient g_j that strengthens or weakens the gravitational force.

Once the gravitational forces between a test instance i and its k-nearest particles have been computed, the strength of the data gravitation field for each label is calculated. Neighbor particles that belong to the ℓth label exert a positive force

Algorithm 1: MLDGC algorithm.

```
Input  : E → training set of multi-label instances
         T_s → test set of multi-label instances
         k → number of nearest neighbors
begin
    // Training phase
    foreach i ∈ E do
        N_i ← kNearestNeighbors (i,E,k );
        g_i ← ngc (i,N_i);     // Eq. (5)
    end
    // Test phase
    foreach i ∈ T_s do
        N_i ← kNearestNeighbors (i,E,k );
        Y_i ← ∅;
        foreach ℓ ∈ L do
            f_ℓ^+ ← positiveGF (i,N_i,ℓ);     // Eq. (14)
            f_ℓ^- ← negativeGF (i,N_i,ℓ);     // Eq. (15)
            if f_ℓ^+ > f_ℓ^- then
                Y_i ← Y_i ∪ ℓ;
            end
            c_ℓ^i ← confidence (f_ℓ^+,f_ℓ^-);     // Eq. (16)
        end
    end
end
```

over the test instance, so the test instance is attracted for belonging to the ℓth label. On the other hand, the neighbor particles that do not belong to the ℓth label exert a negative force over the test instance, so the test instance is repulsed for belonging to the ℓth label. Given a test instance i, the positive and negative forces on the ℓth label are computed as follows:

$$f_\ell^+(i) = \sum_{j \in N_i} f(i,j) \mid \ell \in Y_j \tag{14}$$

$$f_\ell^-(i) = \sum_{j \in N_i} f(i,j) \mid \ell \notin Y_j \tag{15}$$

A test instance i belongs to the ℓth label if $f_\ell^+(i) > f_\ell^-(i)$, otherwise, the test instance does not belong to the ℓth label. The confidence that the test instance i belongs to the ℓth label is computed as follows:

$$c_\ell^i = \frac{f_\ell^+(i)}{f_\ell^+(i) + f_\ell^-(i)} \tag{16}$$

We call this approach Multi-label Data Gravitation Classification (MLDGC). The Algorithm (1) describes the main steps that follow the MLDGC algorithm.

The function `kNearestNeighbors` returns the k-nearest neighbors of i on the training set E. The function `ngc` computes the NGC value of the particle i. The functions `positiveGF` and `negativeGF` compute the positive and negative gravitational forces in the ℓth label, respectively. The function `confidence` calculates the confidence that the particle i belongs to the ℓth label.

The MLDGC algorithm does not use a problem transformation method, i.e. it directly handles the multi-label data, and therefore it falls into category of algorithm adaptation methods for multi-label learning. Given a test instance, MLDGC is able to return a bipartition of the label space into relevant and irrelevant label sets. Alternatively, still based on a test instance, MLDGC can return a label ranking by calculating the confidences on each label. Therefore, MLDGC can resolve multi-label classification and label ranking tasks simultaneously, resolving multi-label ranking generalization.

In the training phase, MLDGC needs to retrieve the k-nearest neighbors for each multi-label instance, in order to convert an instance into a particle. However, if the distance between each pair of multi-label instances is pre-calculated and an adequate data structure for the searching process is employed, the computing of the k-nearest neighbors for each multi-label instance can be efficiently performed. Supposing that a linear search to retrieve the k-nearest neighbors of an instance i is used, the time complexity to form the corresponding particle of i is at most $O(n \cdot d)$, n being the number of training samples and d the cardinality of the feature space. Therefore, in the training phase, MLDGC needs at most $O(n^2 \cdot d)$ to create all data particles.

In the test phase, MLDGC only needs to retrieve k-nearest particles to perform the classification of a test instance i. Supposing that a linear search to retrieve the k-nearest neighbors of a test instance i is used, the time complexity to classify a test instance is at most $O(n \cdot d)$. Therefore, in the test phase, MLDGC needs at most $O(m \cdot n \cdot d)$ steps to classify m test instances.

4. Experimental study

In this section, we explain the means by which multi-label lazy algorithms are evaluated, whilst providing a description of the multi-label datasets and other settings used in the experimental study. Finally, the experimental results on different datasets and the statistical analysis are discussed.

4.1. Evaluation of multi-label lazy algorithms

In this work, several evaluation measures suggested in studies [21,34,51] were used. In multi-label setting, additional degrees of freedom are introduced. Therefore it is essential to include several measures that enable the analysis of different viewpoints for evaluating multi-label learning methods. In case [21], the measures to evaluate multi-label algorithms are divided into two categories: label-based and example-based measures. The example-based measures are further categorized into bipartition-based and ranking-based measures.

Based on a test set $T_s = \{\langle \mathbf{X_i}, Y_i\rangle,\ i = 1 \dots m\}$ of multi-label instances, a multi-label classification method predicts a set of labels Z_i for a given test instance i. On the other hand, a label ranking method ranks labels R_i for a given test instance i, $R_i(\ell)$ being the rank predicted for the label ℓ.

The label-based measure used in this work is the Micro-Average F_1-Measure (M_{iF_1}). The micro approach aggregates the true positive, true negative, false positive, and false negative values of all labels and then calculates the measure

$$M_{iF_1} = F_1\left(\sum_{\ell=1}^{q} tp_\ell, \sum_{\ell=1}^{q} fp_\ell, \sum_{\ell=1}^{q} tn_\ell, \sum_{\ell=1}^{q} fn_\ell\right) \tag{17}$$

where q is the number of labels, and F_1 function computes the F_1-Measure given the true positive (tp), false positive (fp), true negative (tn), and false negative (fn) values.

The bipartition-based measures used in this work are the Hamming Loss (H_L) and Example-based F_1-Measure (F_{1Ex}). H_L averages the symmetrical differences between the predicted and actual label sets, while F_{1Ex} calculates the F_1-Measure for all examples in the test set

$$H_L = \frac{1}{m}\sum_{i=1}^{m} \frac{|\ Y_i \triangle Z_i\ |}{q} \tag{18}$$

$$F_{1Ex} = \frac{1}{m}\sum_{i=1}^{m} \frac{2\,|\ Y_i \cap Z_i\ |}{|\ Y_i\ | + |\ Z_i\ |} \tag{19}$$

where $\triangle$ denotes the symmetric difference between 2 sets and m is the number of test instances.

The ranking-based measures used in this work are the Ranking Loss (R_L), Average Precision (A_P) and One Error (O_E). R_L averages the proportion of label pairs that are incorrectly ordered. A_P averages how many times a particular label is ranked over another label which is in the true label set. O_E averages how many times the top-ranked label is not in the set of true labels for the given instance

$$R_L = \frac{1}{m}\sum_{i=1}^{m} \frac{|\ (\ell_a, \ell_b) : R_i(\ell_a) > R_i(\ell_b),\ (\ell_a, \ell_b) \in Y_i \times \bar{Y}_i\ |}{|\ Y_i\ ||\ \bar{Y}_i\ |} \tag{20}$$

$$A_P = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|\ Y_i\ |} \sum_{\ell \in Y_i} \frac{|\ \{\ell' \in Y_i : R_i(\ell') \le R_i(\ell)\}\ |}{R_i(\ell)} \tag{21}$$

$$O_E = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \delta(\operatorname*{argmin}_{\ell \in \mathcal{L}} R_i(\ell)) \qquad \delta(\ell) = \begin{cases} 1 & \ell \notin Y_i \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where $\bar{Y}_i$ denotes the complementary set of Y_i in $\mathcal{L}$ and m is the number of test instances.

The H_L, F_{1Ex} and M_{iF_1} evaluation measures are associated with the multi-label classification task, whereas R_L, A_P and O_E are associated with the label ranking task. The higher the value of F_{1Ex}, M_{iF_1} and A_P, and the lower the value of H_L, R_L and O_E, the better the performance of a multi-label learning algorithm.

To analyze and validate the results, non-parametric statistical tests were used, as proposed in studies [12,19,20]. Friedman's test [18] was performed to evaluate whether there were significant differences in performance of the lazy algorithms. If Friedman's test indicated that the results were significantly different, a post-hoc test was used to perform multiple comparisons between all methods. In this work, Shaffer's test [44] was used in order to perform multiple comparisons between all methods. In case [20], it was proved that Nemenyi's test [36] is conservative and many of the obvious differences could

Table 1
Statistics of the benchmark datasets, number of instances (n), number of features (d), number of labels (q), different subsets of labels (d_s), label cardinality (l_c) and label density (l_d). The datasets are ordered by their complexity calculated as $n \cdot d$.

Dataset	Domain	n	d	q	d_s	l_c	l_d
Flags	Image	194	19	7	54	3.392	0.485
Cal500	Music	502	68	174	502	26.044	0.150
Emotions	Music	593	72	6	27	1.869	0.311
Birds	Audio	645	260	19	133	1.014	0.053
Yeast	Biology	2417	103	14	198	4.237	0.303
Scene	Image	2407	294	6	15	1.074	0.179
Genbase	Biology	662	1186	27	32	1.252	0.046
Medical	Text	978	1449	45	94	1.245	0.028
Enron	Text	1702	1001	53	753	3.378	0.064
Mediamill	Video	43907	120	101	6555	4.376	0.043
TMC2007-500	Text	28596	500	22	1341	2.16	0.098
Corel5k	Image	5000	499	374	3175	3.522	0.009
Corel16k (10 samples)	Image	13811	500	161	4937	2.867	0.018
Bibtex	Text	7395	1836	159	2856	2.402	0.015
Arts	Text	7484	23146	26	599	1.654	0.064
Science	Text	6428	37187	40	457	1.450	0.036
Business	Text	11214	21924	30	233	1.599	0.053
Health	Text	9250	30605	32	335	1.644	0.051
Reference	Text	8027	39679	33	275	1.174	0.035
Education	Text	12030	27534	33	511	1.463	0.044
Recreation	Text	12828	30324	22	530	1.429	0.065
Entertainment	Text	12730	32001	21	337	1.414	0.067
Computers	Text	12444	34096	33	428	1.507	0.046
Society	Text	14512	31802	27	1054	1.670	0.062
Social	Text	12111	52350	39	361	1.279	0.033

not be detected. The authors proposed to use Bergmann–Hommel's test [2] or Shaffer's test to perform multiple comparisons between all methods. Bergmann–Hommel's test is the most powerful, but it requires intensive computation when numerous classifiers are involved. Shaffer's test can be used instead of Bergmann–Hommel's test in these cases.

In the statistical analysis, the Adjusted p-values (APVs) [60] were considered. APVs provide further information on a statistical analysis. APVs take into account the fact that multiple tests are conducted and can be compared directly with any significance level [20]. In this work, a significance level equal to $\alpha = 0.05$ was considered.

4.2. Multi-label datasets

In the experiments, 34 real-world multi-label datasets were used[1]. Multi-label datasets with different scales and from different application domains were included to analyze the behavior of the multi-label lazy algorithms in datasets with diverse properties.

The datasets come from 6 domains: image, music, audio, biology, text and video. "Birds" [4] contains examples of bird species for acoustic classification. "Cal500" [54] contains pieces of music for semantic annotation. "Emotions" [49] stores examples of songs based on the emotions that they evoke. "Flags" [11] stores examples about nations and their national flags. "Scene" [3] contains a series of patterns concerning kinds of landscapes. The "Corel5k" [15] and "Corel16k" [1] datasets contain Corel images. "Mediamill" [46] contains examples for the automatic detection of semantic concepts in videos. The "Yeast" [16] and "Genbase" [14] datasets include information about the function of genes and proteins, respectively. "Medical" [40] was used in the Medical Natural Language Processing Challenge in 2007. "Enron" [28] contains emails from 151 users. "TMC2007-500" [48,53] stores examples of reports of aviation safety. "Bibtex" [27] contains bibtex examples for automatic tag suggestion. The other 11 datasets come from the Yahoo text collection [55].

Table 1 shows statistics regarding the multi-label datasets. The values of the properties in the Corel16k dataset are averaged over all 10 samples used. The label cardinality is the average number of labels per example. The label density is the label cardinality divided by the total number of labels. The datasets vary in size: from 194 up to 43, 907 instances, from 19 up to 52, 350 features, from 6 to 374 labels, from 15 to 6, 555 different subset of labels, from 1.014 to 26.044 label cardinality, and from 0.009 to 0.485 label density.

4.3. Experimental setting

In the experiments, our proposal MLDGC was compared to MLkNN [70], kNNMLR [6], BRkNN [47], BRkNN-α [47], BRkNN-β [47], LPkNN [47], DMLkNN [66], IBLRML [10], FVkNN [68], EMLkNN [67], MLCWkNN [62] and FSkNN [24] algorithms. Due

[1] All these datasets are available at http://mulan.sourceforge.net/datasets.html .

Table 2
M_{iF_1} (↑) results. Friedman's test rejected the null hypothesis with a p-value equal to 6.712E−11.

Dataset	ML*k*NN	BR*k*NN	BR*k*NN-α	BR*k*NN-β	DML*k*NN	MLCW*k*NN	IBLRML	LP*k*NN	*k*NNMLR	FV*k*NN	FS*k*NN	EML*k*NN	MLDGC
Flags	0.746	0.733	0.729	0.725	0.747	0.715	0.748	0.726	0.733	0.742	0.736	0.458	**0.759**
Cal500	0.321	0.357	0.343	**0.421**	0.315	0.355	0.323	0.353	0.301	0.402	0.335	0.120	0.360
Emotions	0.667	0.670	0.666	0.655	0.671	0.684	**0.687**	0.677	0.670	0.658	0.666	0.558	0.682
Birds	0.377	0.476	0.379	0.381	0.342	0.528	0.449	0.430	0.476	0.540	0.461	0.403	**0.551**
Yeast	0.649	0.654	**0.665**	0.661	0.648	0.660	0.653	0.638	0.654	0.637	0.616	0.510	0.660
Scene	0.743	0.721	0.723	0.729	0.744	0.743	**0.751**	0.724	0.721	0.701	0.746	0.732	0.731
Genbase	0.968	0.968	0.956	0.958	0.959	0.987	0.978	0.959	0.982	0.734	0.976	0.956	**0.989**
Medical	0.685	0.622	0.652	0.639	0.648	**0.687**	0.649	0.647	0.644	0.618	0.673	0.642	0.673
Enron	0.478	0.340	0.352	0.441	0.473	0.432	0.473	0.362	0.378	0.440	0.495	0.331	**0.485**
Mediamill	0.612	0.627	0.626	0.605	0.610	0.674	0.617	0.573	0.604	0.647	0.658	0.502	**0.680**
TMC2007-500	0.643	0.613	0.597	0.618	0.648	**0.696**	0.650	0.606	0.635	0.651	0.634	0.471	0.647
Corel5k	0.033	0.024	0.097	0.154	0.006	0.089	0.058	0.109	0.020	0.132	0.129	0.103	**0.155**
Corel16k01	0.015	0.045	0.112	0.125	0.009	0.089	0.056	0.117	0.021	0.131	0.124	0.135	**0.143**
Corel16k02	0.018	0.049	0.080	0.085	0.011	0.094	0.062	0.117	0.024	0.137	0.141	0.124	**0.147**
Corel16k03	0.013	0.053	0.055	0.058	0.009	0.095	0.053	0.115	0.028	0.138	0.100	0.102	**0.147**
Corel16k04	0.019	0.049	0.035	0.049	0.012	0.094	0.063	0.120	0.028	0.135	0.101	0.064	**0.149**
Corel16k05	0.018	0.051	0.069	0.087	0.013	0.100	0.074	0.111	0.022	0.138	0.110	0.068	**0.150**
Corel16k06	0.017	0.053	0.068	0.074	0.012	0.102	0.069	0.124	0.030	0.138	0.135	0.120	**0.149**
Corel16k07	0.016	0.052	0.042	0.048	0.011	0.098	0.058	0.121	0.025	0.132	0.087	0.060	**0.147**
Corel16k08	0.016	0.051	0.060	0.087	0.010	0.095	0.060	0.115	0.027	0.132	0.140	0.065	**0.147**
Corel16k09	0.009	0.043	0.040	0.042	0.005	0.087	0.049	0.113	0.022	0.134	0.140	0.038	**0.143**
Corel16k10	0.011	0.061	0.074	0.082	0.014	0.099	0.055	0.124	0.023	0.139	0.120	0.075	**0.148**
Bibtex	0.227	0.159	0.211	0.226	0.188	0.239	**0.272**	0.183	0.147	0.226	0.241	0.219	0.253
Arts	0.078	0.145	0.218	0.243	0.078	0.191	0.079	0.221	0.145	0.232	0.248	0.227	**0.275**
Science	0.059	0.116	0.112	0.106	0.056	0.191	0.077	0.173	0.116	0.225	0.064	0.098	**0.226**
Business	0.686	0.687	0.689	0.666	0.686	**0.690**	0.685	0.679	0.687	0.635	0.672	0.679	0.689
Health	0.362	0.393	0.380	0.367	0.362	0.426	0.372	0.421	0.393	0.416	0.411	0.378	**0.464**
Reference	0.353	0.437	0.444	0.441	0.351	0.443	0.361	0.437	0.437	0.413	0.351	0.321	**0.454**
Education	0.074	0.177	0.170	0.180	0.074	0.224	0.074	0.235	0.177	0.274	0.070	0.141	**0.311**
Recreation	0.150	0.205	0.203	0.214	0.150	0.258	0.138	0.262	0.205	0.274	0.241	0.255	**0.301**
Entertainment	0.251	0.300	0.301	0.298	0.249	0.347	0.249	0.355	0.300	0.352	0.164	0.235	**0.394**
Computers	0.442	0.434	0.410	0.412	0.441	0.454	0.448	0.471	0.434	0.432	0.430	0.401	**0.490**
Society	0.233	0.305	0.300	0.298	0.233	0.324	0.284	0.342	0.305	0.318	0.301	0.252	**0.378**
Social	0.351	0.382	0.390	0.378	0.350	0.404	0.383	0.411	0.382	0.381	0.374	0.314	**0.435**
Ave. Rank	9.456	8.029	7.882	7.471	10.250	4.382	7.176	5.676	8.662	5.162	6.073	9.118	1.662
Pos.	12	9	8	7	13	2	6	4	10	3	5	11	1

to the exposed limitations of the Mr.*k*NN [33] method, it was not included in the experimentation. The EML*k*NN algorithm was only included in the comparison of methods regarding the multi-label classification task, as it is not able to resolve the label ranking task.

The multi-label lazy algorithms were implemented on MULAN [52]. MULAN is a Java library which contains several algorithms, evaluation methods and measures for multi-label learning. For each possible combination of algorithms and datasets, a stratified 10-fold cross validation strategy was used. The methods proposed in study [43] were used to stratify the multi-label data. The whole training set was used to retrieve the k-nearest neighbors of an instance. The best number of neighbors (k) was determined for each classifier on each dataset.

4.4. Results and discussion

The algorithms as standalone runnable files and all the results of the experimental study are available in order to facilitate the replicability of the experiments[2]. In this manuscript, only a summary of the results is provided. In all cases, the best results are highlighted in bold typeface in the tables, "↓" indicates "the smaller the better", and "↑" indicates "the larger the better". In the tables, the last 2 rows show the average rank (Ave. Rank) and the ranking position (Pos.) for each lazy learning algorithm according to Friedman's test.

Tables 2–4 show the results of the multi-label lazy algorithms for the M_{iF_1}, H_L and F_{1Ex} measures. The results for R_L, O_E and A_P measures can be consulted on the available web page. Generally speaking, the results show that the MLDGC method demonstrated good performance levels on both simple datasets as well as large-scale datasets.

A statistical analysis was carried out to detect significant differences in the performance levels between multi-label lazy algorithms. Friedman's test rejected the null hypothesis in all cases analyzed, considering a significance level equal to $\alpha = 0.05$. The p-values returned by Friedman's test can be consulted on Tables 2–4.

[2] http://www.uco.es/grupos/kdis/kdiswiki/MLL/MLDGC

Table 3
$H_L(\downarrow)$ results. Friedman's test rejected the null hypothesis with a p-value equal to 1.317E−10.

Dataset	ML*k*NN	BR*k*NN	BR*k*NN-α	BR*k*NN-β	DML*k*NN	MLCW*k*NN	IBLRML	LP*k*NN	*k*NNMLR	FV*k*NN	FS*k*NN	EML*k*NN	MLDGC
Flags	0.254	0.255	0.262	0.262	0.256	0.272	0.253	0.269	0.255	0.283	0.263	0.371	**0.252**
Cal500	**0.138**	0.141	0.143	0.163	**0.138**	0.145	0.228	0.195	0.139	0.206	0.145	0.146	**0.138**
Emotions	0.193	0.193	0.195	0.205	0.190	0.184	0.185	0.204	0.193	0.237	0.197	0.224	**0.181**
Birds	0.046	0.044	0.070	0.060	0.046	**0.041**	0.047	0.054	0.044	0.055	0.045	0.065	**0.041**
Yeast	0.192	0.193	0.190	0.192	0.192	0.189	0.193	0.214	0.193	0.248	0.214	0.221	**0.187**
Scene	0.084	0.090	0.094	0.093	0.086	0.084	**0.083**	0.095	0.091	0.119	0.085	0.092	0.086
Genbase	0.003	0.003	0.004	0.003	0.004	0.002	0.002	0.004	0.002	0.020	0.003	0.004	**0.001**
Medical	**0.015**	0.017	0.019	0.020	0.016	0.016	0.019	0.019	0.018	0.024	**0.015**	0.018	**0.015**
Enron	**0.053**	0.058	0.061	0.063	**0.053**	0.055	0.055	0.068	0.058	0.077	0.054	0.061	0.054
Mediamill	0.028	0.028	0.028	0.027	0.028	0.025	0.028	0.036	0.029	0.033	0.029	0.029	**0.024**
TMC2007-500	0.065	0.067	0.070	0.074	0.064	0.038	0.064	0.072	0.066	0.051	0.055	0.079	**0.037**
Corel5k	0.010	0.010	0.011	0.015	0.010	0.010	0.021	0.016	0.010	0.019	0.010	0.011	**0.009**
Corel16k01	0.019	0.019	0.023	0.031	0.019	0.019	0.019	0.032	0.019	0.043	0.021	0.023	**0.018**
Corel16k02	**0.018**	**0.018**	0.019	0.022	**0.018**	**0.018**	**0.018**	0.030	**0.018**	0.039	0.022	0.019	**0.018**
Corel16k03	0.018	0.018	**0.017**	0.023	0.018	0.019	0.019	0.032	0.019	0.040	0.019	0.021	0.018
Corel16k04	0.018	0.018	0.024	0.026	0.018	0.018	0.018	0.031	0.018	0.038	0.020	0.021	**0.017**
Corel16k05	0.019	0.018	0.024	0.028	0.023	0.020	0.020	0.019	0.018	0.038	0.019	0.022	**0.017**
Corel16k06	0.020	0.022	0.021	0.023	0.021	0.020	0.019	0.018	0.018	0.039	0.021	0.022	**0.017**
Corel16k07	0.020	0.021	0.021	0.022	0.021	0.020	0.019	0.018	0.017	0.036	0.020	0.022	**0.016**
Corel16k08	**0.017**	**0.017**	0.018	0.018	**0.017**	**0.017**	0.018	0.031	**0.017**	0.039	0.018	0.030	**0.017**
Corel16k09	0.018	**0.017**	**0.017**	0.021	**0.017**	**0.017**	**0.017**	0.029	**0.017**	0.037	0.019	0.031	**0.017**
Corel16k10	0.018	0.018	0.024	0.023	0.020	0.019	0.018	0.018	0.025	0.039	0.018	0.020	**0.016**
Bibtex	0.014	0.015	0.017	0.024	0.014	**0.013**	0.016	0.022	0.014	0.029	0.019	0.018	0.014
Arts	**0.062**	0.063	0.080	0.091	**0.062**	0.065	**0.062**	0.082	0.064	0.104	0.074	0.080	**0.062**
Science	**0.036**	**0.036**	0.039	0.038	**0.036**	**0.036**	**0.036**	0.052	**0.036**	0.057	0.039	0.040	**0.036**
Business	**0.028**	**0.028**	**0.028**	0.034	**0.028**	0.029	**0.028**	0.029	0.029	0.039	0.031	**0.028**	**0.028**
Health	0.047	0.048	0.049	0.052	0.047	0.047	**0.046**	0.050	0.047	0.064	0.047	0.072	**0.046**
Reference	**0.033**	0.034	0.038	0.040	0.034	0.034	**0.033**	0.037	0.034	0.043	0.035	0.040	0.034
Education	**0.043**	0.044	**0.043**	0.046	0.044	0.045	**0.043**	0.059	**0.043**	0.067	0.051	0.055	**0.043**
Recreation	**0.061**	**0.061**	0.063	0.063	**0.061**	0.062	**0.061**	0.083	**0.061**	0.097	0.065	0.063	**0.061**
Entertainment	0.060	0.060	0.059	0.061	0.060	0.061	0.060	0.076	0.060	0.092	0.078	0.065	**0.059**
Computers	0.039	0.039	0.043	0.044	0.039	0.039	**0.038**	0.041	0.039	0.052	0.042	0.048	**0.038**
Society	0.058	0.059	0.060	0.064	0.059	0.059	**0.057**	0.067	0.058	0.091	0.062	0.060	**0.057**
Social	0.029	0.029	0.030	0.030	0.029	0.029	**0.028**	0.035	0.029	0.044	0.033	0.041	**0.028**
Ave. Rank	4.309	5.485	8.059	10.000	5.176	5.073	4.912	10.103	5.265	12.471	7.750	10.176	2.221
Pos.	2	7	9	10	5	4	3	11	6	13	8	12	1

MLDGC obtained the first position in 4 of the 6 ranking of methods (6 rankings = 1 ranking of methods for each evaluation measure), and also obtained the second and the third position on the rankings for measures A_P and R_L, respectively.

A Shaffer's post-hoc test for all pairwise comparisons was then carried out. The results of Shaffer's test are showed in Table 5. Evaluation measures are provided in each cell of the table with the APVs resulting from Shaffer's test, for which the algorithm located in the row significantly outperforms the algorithm located in the column. In a cell appears a dash (-) for those cases where Shaffer's test does not detect significant differences between 2 algorithms for any of the 6 evaluation measures used.

For the label-based measure M_{iF_1}, Shaffer's test detected that MLDGC algorithm significantly outperformed the other lazy algorithms, except the MLCW*k*NN method, for the significance level considered. The MLCW*k*NN, FV*k*NN and LP*k*NN algorithms showed a good performance levels on this label-based measure. The BR*k*NN-α, BR*k*NN-β, BR*k*NN, *k*NNMLR, EML*k*NN, ML*k*NN and DML*k*NN algorithms demonstrated poor performance levels.

For the bipartition-based measure H_L, Shaffer's test detected that the MLDGC algorithm significantly outperformed the BR*k*NN, BR*k*NN-α, BR*k*NN-β, LP*k*NN, *k*NNMLR, FV*k*NN, FS*k*NN and EML*k*NN algorithms. The ML*k*NN, IBLRML, MLCW*k*NN and DML*k*NN algorithms showed good performance levels on this bipartition-based measure. The BR*k*NN-α, BR*k*NN-β, LP*k*NN, EML*k*NN and FV*k*NN algorithms showed poor performance levels.

For the bipartition-based measure F_{1Ex}, Shaffer's test detected that the MLDGC algorithm significantly outperformed the other lazy algorithms, except the FV*k*NN method. The FV*k*NN, LP*k*NN and MLCW*k*NN algorithms showed good performance levels on this bipartition-based measure. The EML*k*NN, BR*k*NN, *k*NNMLR, ML*k*NN and DML*k*NN algorithms demonstrated poor performance levels.

We recall that the EML*k*NN method was not included in the comparison regarding the label ranking task. For the ranking-based measure R_L, Shaffer's test detected that the MLDGC algorithm significantly outperformed the BR*k*NN, BR*k*NN-α, BR*k*NN-β, MLCW*k*NN, LP*k*NN, *k*NNMLR and FV*k*NN algorithms. The DML*k*NN, ML*k*NN and IBLRML algorithms showed good performance levels on this ranking-based measure. The FV*k*NN, *k*NNMLR, MLCW*k*NN and LP*k*NN algorithms demonstrated poor performance levels.

Table 4
$F_{1Ex}(\uparrow)$ results. Friedman's test rejected the null hypothesis with a p-value equal to 1.139E−10.

Dataset	ML*k*NN	BR*k*NN	BR*k*NN-α	BR*k*NN-β	DML*k*NN	MLCW*k*NN	IBLRML	LP*k*NN	*k*NNMLR	FV*k*NN	FS*k*NN	EML*k*NN	MLDGC
Flags	0.729	0.715	0.703	0.712	0.731	0.691	0.729	0.707	0.715	0.729	0.722	0.402	**0.744**
Cal500	0.325	0.353	0.345	0.437	0.319	0.350	0.320	0.348	0.298	**0.395**	0.328	0.102	0.355
Emotions	0.627	0.627	0.638	0.655	0.630	0.639	0.636	0.655	0.627	0.643	0.628	0.539	**0.668**
Birds	0.600	0.639	0.248	0.230	0.582	0.660	0.631	0.628	0.639	0.652	0.622	0.550	**0.669**
Yeast	0.624	0.627	0.614	0.621	0.624	**0.632**	0.628	0.620	0.627	0.628	0.592	0.579	**0.632**
Scene	0.702	0.689	0.733	0.732	0.705	0.701	0.707	0.729	0.689	0.724	0.723	**0.738**	0.735
Genbase	0.974	0.971	0.966	0.968	0.969	0.989	0.984	0.970	0.985	0.508	0.979	0.971	**0.993**
Medical	0.609	0.531	0.654	0.645	0.552	0.619	0.627	0.653	0.570	0.643	0.601	0.652	**0.657**
Enron	0.444	0.285	0.354	0.445	0.434	0.401	0.445	0.359	0.321	0.435	0.472	0.329	**0.480**
Mediamill	0.595	0.607	0.577	0.600	0.594	0.652	0.598	0.565	0.581	0.645	0.620	0.512	**0.658**
TMC2007-500	0.626	0.584	0.581	0.614	0.625	**0.772**	0.627	0.597	0.611	0.764	0.632	0.489	0.640
Corel5k	0.021	0.015	0.097	0.140	0.004	0.065	0.051	0.108	0.013	0.120	0.099	0.106	**0.142**
Corel16k01	0.010	0.031	0.107	0.111	0.006	0.066	0.038	0.113	0.013	**0.130**	0.100	0.109	0.128
Corel16k02	0.011	0.032	0.045	0.056	0.008	0.069	0.040	0.110	0.016	**0.133**	0.054	0.054	**0.133**
Corel16k03	0.009	0.035	0.040	0.042	0.006	0.069	0.034	0.109	0.018	**0.133**	0.068	0.068	**0.133**
Corel16k04	0.012	0.032	0.028	0.031	0.008	0.068	0.039	0.114	0.017	**0.132**	0.042	0.071	0.131
Corel16k05	0.011	0.033	0.045	0.058	0.008	0.070	0.037	0.016	0.135	0.134	0.121	0.054	**0.136**
Corel16k06	0.013	0.028	0.030	0.030	0.034	0.069	0.071	0.036	0.019	**0.132**	0.035	0.038	0.131
Corel16k07	0.014	0.030	0.028	0.031	0.036	0.071	0.071	0.072	0.017	**0.133**	0.040	0.063	**0.133**
Corel16k08	0.010	0.032	0.054	0.050	0.007	0.067	0.039	0.109	0.016	0.130	0.041	0.067	**0.131**
Corel16k09	0.006	0.030	0.035	0.033	0.004	0.065	0.033	0.108	0.015	**0.132**	0.058	0.058	0.129
Corel16k10	0.007	0.033	0.100	0.102	0.005	0.066	0.032	0.120	0.017	**0.130**	0.020	0.039	**0.130**
Bibtex	0.174	0.113	0.195	0.215	0.144	0.178	0.227	0.188	0.105	**0.240**	0.188	0.228	0.235
Arts	0.059	0.110	0.231	0.244	0.059	0.161	0.059	0.240	0.115	0.248	0.198	0.248	**0.285**
Science	0.040	0.083	0.090	0.070	0.038	0.151	0.051	0.186	0.083	**0.244**	0.180	0.089	**0.244**
Business	**0.746**	0.744	0.743	0.709	**0.746**	0.741	0.745	0.741	0.744	0.686	0.680	0.743	0.745
Health	0.341	0.399	0.387	0.384	0.341	0.418	0.349	0.450	0.374	0.434	0.368	0.314	**0.475**
Reference	0.281	0.427	0.431	0.429	0.279	0.418	0.292	0.442	0.414	0.430	0.297	0.324	**0.463**
Education	0.038	0.125	0.128	0.145	0.038	0.172	0.037	0.234	0.125	0.271	0.298	0.200	**0.309**
Recreation	0.093	0.141	0.160	0.154	0.092	0.197	0.084	0.271	0.141	0.290	0.102	0.120	**0.308**
Entertainment	0.160	0.213	0.235	0.211	0.157	0.270	0.151	0.360	0.213	0.359	0.198	0.190	**0.402**
Computers	0.408	0.390	0.398	0.395	0.406	0.424	0.405	0.487	0.390	0.458	0.411	0.350	**0.508**
Society	0.197	0.281	0.297	0.302	0.197	0.308	0.254	0.373	0.281	0.344	0.240	0.158	**0.404**
Social	0.280	0.326	0.378	0.368	0.280	0.366	0.315	0.437	0.326	0.411	0.324	0.253	**0.465**
Ave. Rank	9.956	8.794	7.471	6.647	10.397	5.426	7.956	5.088	9.235	3.279	7.147	8.191	1.412
Pos.	12	10	7	5	13	4	8	3	11	2	6	9	1

For the ranking-based measure O_E, Shaffer's test detected that the MLDGC algorithm significantly outperformed the BR*k*NN, BR*k*NN-α, BR*k*NN-β, LP*k*NN, *k*NNMLR, FV*k*NN and FS*k*NN algorithms. The ML*k*NN, DML*k*NN, IBLRML and MLCW*k*NN algorithms showed good performance levels on this ranking-based measure. The *k*NNMLR, BR*k*NN-β, BR*k*NN-α, FV*k*NN and LP*k*NN algorithms demonstrated poor performance levels.

For the ranking-based measure A_P, the Shaffer's test detected that our proposal MLDGC significantly outperformed BR*k*NN, BR*k*NN-α, BR*k*NN-β, MLCW*k*NN, LP*k*NN, *k*NNMLR and FV*k*NN algorithms. The IBLRML, ML*k*NN and DML*k*NN algorithms showed a good performance on this ranking-based measure. The BR*k*NN-α, BR*k*NN-β, FV*k*NN and LP*k*NN algorithms showed a poor performance.

Generally speaking, our proposal – MLDGC – boasted good performance levels on both datasets with different properties as well as the multi-label classification and label ranking tasks.

4.4.1. Discussion

From a statistical point of view, the MLDGC and IBLRML classifiers were the two algorithms that performed the best. The MLDGC method was not significantly outperformed by other lazy algorithm in any of the six evaluation measures used. The IBLRML method was only outperformed by the FV*k*NN method in the F_{1Ex} measure, and by the MLDGC method in the F_{1Ex} and M_{iF_1} measures.

For the M_{iF_1} measure, the MLDGC, MLCW*k*NN and FV*k*NN algorithms showed the best performance levels. For the H_L measure, the best results were obtained by the MLDGC, ML*k*NN and IBLRML algorithms. For the F_{1Ex} measure, the best results were obtained by the MLDGC, FV*k*NN and LP*k*NN algorithms. Regarding the label-ranking measures (R_L, O_E and A_P), the best results were obtained by the ML*k*NN, DML*k*NN, MLDGC and IBLRML algorithms.

The results showed that the LP*k*NN, FV*k*NN and *k*NNMLR methods demonstrated poor performance levels on the ranking-based measures. For the H_L measure, the LP*k*NN, EML*k*NN and FV*k*NN algorithms showed the worst results. However, for the F_{1Ex} and M_{iF_1} measures, the worst results were obtained by the ML*k*NN and DML*k*NN algorithms.

Based on these results and the statistical analysis, we concluded that the MLDGC algorithm performed well in the resolution of the multi-label classification and label ranking tasks. However, it is relevant to note that it obtained better results

Table 5
Significant differences of the performance levels between all multi-label lazy algorithms according to Shaffer's test.

	ML*k*NN	BR*k*NN	BR*k*NN-α	BR*k*NN-β	DML*k*NN	MLCW*k*NN	IBLRML	LP*k*NN	*k*NNMLR	FV*k*NN	FS*k*NN	EML*k*NN
ML*k*NN	-		$H_L = 0.0$	$H_L = 0.0$	-		-	$H_L = 0.0$		$H_L = 0.0$		
		$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$		$R_L = 0.0$		$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$	$H_L = 0.1$	$H_L = 0.0$
		$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.0$		$A_P = 0.1$		$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.2$	
		$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$				$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$		
BR*k*NN	-	-	-	$H_L = 0.0$	-	-	-	$H_L = 0.0$	-		-	$H_L = 0.0$
								$R_L = 0.0$		$H_L = 0.0$		
								$O_E = 0.0$		$O_E = 0.3$		
								$A_P = 0.0$				
BR*k*NN-α	-	-	-	-	-	-	-	$R_L = 0.0$	-	$H_L = 0.0$	-	-
								$O_E = 0.0$				
								$A_P = 0.0$				
BR*k*NN-β	$F_{1Ex} = 0.02$	-	-	-	$F_{1Ex} = 0.0$	-	-	$R_L = 0.0$	-	-	-	-
								$O_E = 0.0$				
								$A_P = 0.0$				
DML*k*NN	-			$H_L = 0.0$	-		-	$H_L = 0.0$		$H_L = 0.0$		
		$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$		$R_L = 0.0$		$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.2$	
		$O_E = 0.1$	$O_E = 0.0$	$O_E = 0.0$		$A_P = 0.1$		$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.3$	
		$A_p = 0.0$	$A_p = 0.0$	$A_P = 0.0$				$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$		
MLCW*k*NN						-	-	$H_L = 0.0$		$H_L = 0.0$	-	
	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$M_{iF_1} = 0.1$	$H_L = 0.0$	$F_{1Ex} = 0.0$			$O_E = 0.0$	$F_{1Ex} = 0.0$	$O_E = 0.0$		$H_L = 0.0$
	$M_{iF_1} = 0.0$	$M_{iF_1} = 0.01$	$O_E = 0.01$	$O_E = 0.01$	$M_{iF_1} = 0.00$			$A_P = 0.0$	$M_{iF_1} = 0.00$	$A_P = 0.01$		$M_{iF_1} = 0.0$
IBLRML	-		$H_L = 0.04$	$H_L = 0.0$	-		-	$H_L = 0.0$		$H_L = 0.0$	-	$H_L = 0.0$
		$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$		$R_L = 0.0$	-	$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$		
		$O_E = 0.01$	$O_E = 0.0$	$O_E = 0.0$		$A_P = 0.01$		$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.0$		
		$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$				$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$		
LP*k*NN	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	-	-	$F_{1Ex} = 0.0$	-	-	-	$F_{1Ex} = 0.0$	-	-	$F_{1Ex} = 0.05$
	$M_{iF_1} = 0.0$				$M_{iF_1} = 0.0$							$M_{iF_1} = 0.01$
*k*NNMLR	-	-	-	$H_L = 0.0$	-	-	-	$H_L = 0.0$	-	$H_L = 0.0$	-	$H_L = 0.0$
								$R_L = 0.0$				
								$O_E = 0.0$				
								$A_P = 0.0$				
FV*k*NN	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.02$	$F_{1Ex} = 0.0$	-	$F_{1Ex} = 0.0$	$R_L = 0.0$	$F_{1Ex} = 0.0$	-	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$
	$M_{iF_1} = 0.0$				$M_{iF_1} = 0.0$				$M_{iF_1} = 0.01$			$M_{iF_1} = 0.0$
FS*k*NN	$M_{iF_1} = 0.02$	-	-	-	$F_{1Ex} = 0.03$	$R_L = 0.0$	-	$R_L = 0.0$	$R_L = 0.03$	$H_L = 0.0$	-	-
					$M_{iF_1} = 0.0$			$O_E = 0.0$		$O_E = 0.0$		
								$A_P = 0.0$		$A_P = 0.0$		
MLDGC		$H_L = 0.02$	$H_L = 0.0$	$H_L = 0.0$				$H_L = 0.0$	$H_L = 0.05$			
		$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$				$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$M_{iF_1} = 0.01$	$H_L = 0.0$	
	$F_{1Ex} = 0.0$	$M_{iF_1} = 0.0$	$M_{iF_1} = 0.0$	$M_{iF_1} = 0.0$	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$F_{1Ex} = 0.0$	$M_{iF_1} = 0.0$	$M_{iF_1} = 0.0$	$M_{iF_1} = 0.01$	$F_{1Ex} = 0.0$	$H_L = 0.0$
	$M_{iF_1} = 0.0$	$R_L = 0.0$	$R_L = 0.0$	$R_L = 0.0$	$M_{iF_1} = 0.0$	$R_L = 0.0$	$M_{iF_1} = 0.0$	$R_L = 0.0$	$R_L = 0.0$			$F_{1Ex} = 0.0$
		$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.0$		$A_P = 0.01$		$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.0$	$O_E = 0.01$	$M_{iF_1} = 0.0$
		$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$				$A_P = 0.0$	$A_P = 0.0$	$A_P = 0.0$		

on metrics to evaluate bipartitions (H_L and F_{1Ex}) and label-based metrics (M_{iF_1}) than on metrics to evaluate rankings (O_E, R_L and A_P).

The MLkNN and DMLkNN algorithms obtained better results on metrics to evaluate rankings than on metrics to evaluate bipartitions and label-based metrics, i.e. they obtained better results for the label ranking task than for the multi-label classification task. The FVkNN and LPkNN algorithms performed well on the multi-label classification task. However, they demonstrated poor performance levels for the label ranking task. IBLRML showed better performance levels on the label ranking task than on the multi-label classification task. According to the results, the kNNMLR, BRkNN and its extensions, BRkNN-α and BRkNN-β, showed poor performance levels in both tasks. MLCWkNN showed better performance in the resolution of the multi-label classification task than for the resolution of the label ranking task. The FSkNN algorithm had a similar performance in both tasks. The EMLkNN method showed poor performance levels in the multi-label classification task.

The proposed MLDGC algorithm performed well for simple and large-scale multi-label datasets. For the multi-label classification task, MLDGC tended to perform better in datasets with a low label density and a large number of distinct label sets at the same time, e.g. the Corel16k dataset and the Yahoo text collection. For the label ranking task, MLDGC tended to perform better in datasets with a low label density.

Based on the statistical analysis, we can conclude that the proposed MLDGC algorithm was competitive with respect to state-of-the-art multi-label lazy algorithms. MLDGC outperformed the other 12 lazy algorithms included in the experimental study with regards to several evaluation metrics, confirming the effectiveness of our proposal for multi-label lazy learning.

Regarding computational complexity, multi-label lazy algorithms that use the Binary Relevance approach as a problem transformation method, e.g. the IBLRML method, are computationally expensive for multi-label datasets that have a large number of labels. The IBLRML algorithm is expensive in terms of computational time, since it creates new training data with label information as features, and for every label created it must train a logistic regression classifier. The evidence suggests that lazy algorithms which use the Label Power Set approach as their problem transformation method, e.g. the LPkNN method, deplore poor performance levels in the label ranking task. This situation is probably caused by the loss of multi-label information that occurs when a multi-label dataset is converted into a multi-class dataset.

Our MLDGC algorithm boasts similar computational complexity to the MLkNN, DMLkNN and FVkNN algorithms, as they compute the k-nearest neighbors for every instance during the training phase. The IBLRML and FSkNN algorithms have the highest computational complexity. The kNNMLR, BRkNN and its extensions, BRkNN-α and BRkNN-β, have the lowest computational complexity.

The state-of-the-art data gravitation classification (DGC) algorithms are designed to work on single-label data [7,38,39,45,58]. For this reason, MLDGC is the only method based on the DGC approach which we included in our experimental study. Note that single-label DGC algorithms can be evaluated with regards to multi-label data via problem transformation methods, such as the Label Power Set, Binary Relevance or Ranking By Pairwise Comparison methods. However, problem transformation methods can generate highly imbalanced single-label datasets. Consequently, single-label DGC algorithms can be known to perform poorly. Moreover, the use of problem transformation methods in multi-label datasets with a large number of labels and/or a large number of distinct label sets involves a very high computational cost. To avoid the aforementioned drawbacks when using problem transformation methods, MLDGC directly handles the multi-label data, using simple principles with high performance.

In Section 2.3, we described the drawbacks that the particle creation process presents for various instances on single-label data. Later, in Section 3, we described other challenges that arise when creating artificial particles from multiple instances on multi-label data. MLDGC differs from traditional DGC works [38,39] as it considers each multi-label instance as an atomic data particle. Not only does it consider each instance as an atomic data particle, it also provides an accurate local classification and a good generalization where there are no closer training instances. MLDGC also differs from traditional DGC methods in that it uses a new concept to calculate gravitational force, known as the Neighborhood-based Gravitation Coefficient, instead of the mass of the particle. This coefficient strengthens or weakens the gravitational force of a particle over a test instance, avoiding the high impact that particles have when faced with big masses. MLDGC uses only the k-nearest particles for classifying a test instance, instead of using all training particles, avoiding the negative impact that far particles demonstrate when faced with huge masses.

5. Conclusions

In this work, a new multi-label lazy algorithm named MLDGC has been presented. MLDGC is based on the principles of a data gravitation model. It directly handles multi-label data and considers each training instance as a new atomic data particle. A new concept – the Neighborhood-based Gravitation Coefficient of a particle – is defined and used in the gravitational force calculation instead of the particle's mass.

MLDGC is efficient in terms of computational complexity. In the training phase, it needs to compute the k-nearest neighbors for each instance in order to create the corresponding particle. The creation process of atomic particles can be computed efficiently if the distance between each pair of multi-label instances is pre-calculated, and an adequate data structure for the searching process is utilized.

The proposed algorithm was extensively compared to 12 state-of-the-art multi-label lazy algorithms. The statistical analysis showed that our proposal significantly outperformed the other lazy algorithms in several evaluation metrics. MLDGC

performed well for the multi-label classification and label ranking tasks, and obtained good results on simple and large-scale multi-label datasets with different properties. However, the evidence suggested that MLDGC tended to perform better in datasets with a low label density. The results also showed that considering each training instance as an atomic data particle is a good solution for avoiding the problems that may arise in the creation of artificial particles from various instances.

The experimental study confirmed the benefits of a data gravitation model for multi-label lazy learning, the main motivation for the present work. We recommend that when new lazy algorithms for multi-label learning are proposed, they should be compared with state-of-the-art lazy algorithms and the method proposed in this work, using several evaluation measures and a considerable number of datasets with different properties.

Future research will study the use of other approaches to adapt the data gravitation model to multi-label learning. It is essential that the effectiveness of the proposal be assessed on synthetic multi-label datasets. Furthermore, it is important to conduct a comparative study on how the proposed lazy algorithm scales the state-of-the-art model-based multi-label algorithms.

Acknowledgments

This research was supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds, project TIN-2014-55252-P.

References

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, M.I. Jordan, Matching words and pictures, J. Mach. Learn. Res. 3 (2003) 1107–1135.
[2] G. Bergmann, G. Hommel, Improvements of general multiple test procedures for redundant systems of hypotheses, in: P. Bauer, G. Hommel, E. Sonnemann (Eds.), Multiple Hypotheses Testing, Springer-Verlag, Berlin, 1988, pp. 100–115.
[3] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.
[4] F. Briggs, et. al., The ninth annual MLSP competition: new methods for acoustic classification of multiple simultaneous bird species in a noisy environment, in: Proceedings of the International Workshop on Machine Learning for Signal Processing (MLSP'13), IEEE, 2013.
[5] K. Brinker, J. Furnkranz, E. Hullermeier, A unified model for multilabel classification and ranking, in: Proceedings of the Seventeenth European Conference on Artificial Intelligence (ECAI'06), 2006, pp. 489–493.
[6] K. Brinker, E. Hullermeier, Case-based multilabel ranking, in: Proceedings of the 20th International Conference on Artificial Intelligence (IJCAI'07), 2007, pp. 702–707.
[7] A. Cano, A. Zafra, S. Ventura, Weighted data gravitation classification for standard and imbalanced data, IEEE Trans. Cybern. 43 (6) (2013) 1672–1687.
[8] N. Cesa-Bianchi, G. Valentini, Hierarchical cost-sensitive algorithms for genome-wide gene function prediction, J. Mach. Learn. Res. 8 (2010) 14–29.
[9] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, Addressing imbalance in multilabel classification: measures and random resampling algorithms, Neurocomputing 163 (2015) 3–16.
[10] W. Cheng, E. Hullermeier, Combining instance-based learning and logistic regression for multilabel classification, Mach. Learn. 76 (2–3) (2009) 211–225.
[11] E. Correa, A. Plastino, A. Freitas, A genetic algorithm for optimizing the label ordering in multi-label classifier chains, in: Proceedings of the 25th International Conference on Tools with Artificial Intelligence (ICTAI'13), IEEE, 2013.
[12] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[13] T. Denceux, A *k*-nearest neighbor classification rule based on Dempster–Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 804–813.
[14] S. Diplarisa, G. Tsoumakas, P. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: Proceedings of the Tenth Panhellenic Conference on Informatics (PCI'05), 2005, pp. 448–456.
[15] P. Duygulu, K. Barnard, N. de Freitas, D. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: Proceedings of the Seventh European Conference on Computer Vision, 2002, pp. 97–112.
[16] A. Elisseeff, J. Weston, A Kernel method for multi-labelled classification, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, vol. 14, MIT Press, 2001, pp. 681–687.
[17] Y. Endo, H. Iwata, Dynamic clustering based on universal gravitation model, in: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence, LNCS, vol. 3558, Springer-Verlag, Berlin/Heidelberg, 2005, pp. 183–193.
[18] M. Friedman, A comparison of alternative tests of significance for the problem of *m* rankings, Ann. Math. Stat. 11 (1940) 86–92.
[19] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[20] S. García, F. Herrera, An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[21] E. Gibaja, S. Ventura, Multi-label learning: a review of the state of the art and ongoing research, WIREs Data Min. Knowl. Discov. 4 (2014) 411–444.
[22] J. Gómez, D. Dasgupta, O. Nasraoui, A new gravitational clustering algorithm, in: Proceedings of the SIAM International Conference on Data Mining (SDM), 2013.
[23] E. Hullermeier, J. Furnkranz, W. Cheng, K. Brinker, Label ranking by learning pairwise preferences, Artif. Intell. 172 (2008) 1897–1916.
[24] J. Jiang, S. Tsai, S.J. Lee, FS*k*NN: multi-label text classification based on fuzzy similarity and *k*-nearest neighbors, Expert Syst. Appl. 39 (2012) 2813–2821.
[25] L. Junlin, F. Hongguang, Data classification based on supporting data gravity, in: Proceedings of the International Conference on Intelligent Computing and Intelligent Systems (ICIS'09), IEEE, Shanghai, China, 2009, pp. 22–28.
[26] J. Wang, Y. Zhao, X. Wu, X. Hua, A transductive multi-label learning approach for video concept detection, Pattern Recognit. 44 (2010) 2274–2286.
[27] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), Antwerp, Belgium, 2008.
[28] B. Klimt, Y. Yang, The Enron corpus: a new dataset for email classification research, in: Proceedings of the Fifteenth European Conference on Machine Learning (ECML'04), 2004, pp. 217–226.
[29] I. Kononenko, M. Robnik-Sikonja, Computational Methods of Feature Selection, chapter Non-Myopic feature quality evaluation with (R)ReliefF, Chapman & Hall/CRC, 2008, pp. 169–191.
[30] I. kononenko, E. Simec, M.R. Sikonja, Overcoming the myopia of inductive learning algorithms with RELIEFF, Appl. Intell. 7 (1997) 39–55.
[31] M. Larese, P. Granitto, J. Gómez, Spot defects detection in cDNA microarray images, Pattern Anal. Appl. 16 (2013) 307–319.
[32] T. Li, M. Ogihara, Detecting emotion in music, in: Proceedings of the International Symposium on Music Information Retrieval, Washington DC, United States of America, 2003, pp. 239–240.
[33] X. Lin, X.W. Chen, Mr.KNN: Soft Relevance for Multi-label Classification, in: Proceedings of the Nineteenth International Conference on Information and Knowledge Management (CIKM'10), ACM, New York, United States of America, 2010, pp. 349–358.

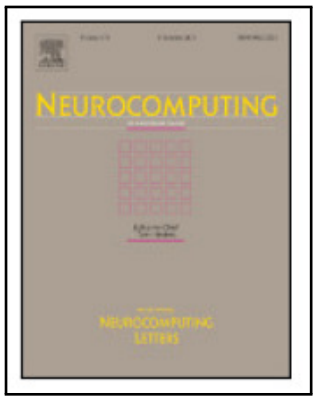
[34] G. Madjarov, D. Kocev, D. Gjorgjevikj, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (2012) 3084–3104.
[35] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: Proceedings of the Working Notes of the AAAI'99 Workshop on Text Learning, 1999.
[36] P.B. Nemenyi, Distribution-Free Multiple-Comparisons, Princeton University, United States of America, 1963 Ph.D. thesis.
[37] F. Otero, A. Freitas, C. Johnson, A hierarchical multi-label classification ant colony algorithm for protein function prediction, Memet. Comput. 2 (2010) 165–181.
[38] L. Peng, B. Yang, Y. Chen, A. Abraham, Data gravitation based classification, Inf. Sci. 179 (2009) 809–819.
[39] L. Peng, H. Zhang, B. Yang, Y. Chen, A new approach for imbalanced data classification based on data gravitation, Inf. Sci. 288 (2014) 347–373.
[40] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (2011) 1–27.
[41] M. Robnik-Sikonja, I. Kononenko, An adaptation of relief for attribute estimation in regression, in: Proceedings of the International Conference on Machine Learning (ICML'97), 1997, pp. 296–304.
[42] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (2003) 23–69.
[43] K. Sechidis, G. Tsoumakas, I. Vlahavas, On the stratification of multi-label data, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), Springer-Verlag, Berlin/Heidelberg, 2011, pp. 145–158.
[44] J. Shaffer, Modified sequentially rejective multiple test procedures, J. Am. Stat. Assoc. 81 (395) (1986) 826–831.
[45] P. Shafigha, S.Y. Hadi, E. Sohrab, Gravitation based classification, Inf. Sci. 220 (2013) 319–330.
[46] C. Snoek, M. Worring, J.C. Van-Gemert, J. Geusebroek, A.Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of ACM Multimedia, ACM, Santa Barbara, United States of America, 2006, pp. 421–430.
[47] E. Spyromitros, G. Tsoumakas, I. Vlahavas, An empirical study of lazy multilabel classification algorithms, in: Proceedings of the Fifth Hellenic Conference on Artificial Intelligence (SETN'08), in: LNAI, vol. 5138, Springer-Verlag, Berlin/Heidelberg, 2008, pp. 401–406.
[48] A. Srivastava, B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: Proceedings of the Aerospace Conference, IEEE, 2005, pp. 55–63.
[49] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas, Multilabel classification of music into emotions, in: Proceedings of the Ninth International Conference on Music Information Retrieval (ISMIR'08), 2008, pp. 325–330.
[50] G. Tsoumakas, I. Katakis, Multi-label classification: an overview, Int. J. Data Warehous. Min. 3 (2007) 1–13.
[51] G. Tsoumakas, I. Katakis, I. Vlahavas, Data Mining and Knowledge Discovery Handbook, chapter Mining Multi-label Data, second ed., Springer-Verlag, New York, United States of America, 2010, pp. 667–686.
[52] G. Tsoumakas, E. Spyromitros-Xioufi, J. Vilcek, I. Vlahavas, MULAN: a java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.
[53] G. Tsoumakas, I. Vlahavas, Random *k*-labelsets: an ensemble method for multilabel classification, in: Proceedings of the Eighteenth European Conference on Machine Learning (ECML'07), Warsaw, Poland, 2007, pp. 406–417.
[54] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Trans. Audio Speech Lang. Process. 16 (2) (2008) 467–476.
[55] N. Ueda, K. Saito, Parametric mixture models for multi-labeled text, in: Proceedings on Neural Information Processing Systems (NIPS'15), MIT Press, 2002, pp. 721–728.
[56] E. Ukwatta, J. Samarabandu, Vision based metal spectral analysis using multi-label classification, in: Proceedings of the Canadian Conference on Computer and Robot Vision (CRV'09), 2009, pp. 132–139.
[57] C. Wang, Y.Q. Chen, Improving nearest neighbor classification with simulated gravitational collapse, in: L. Wang, K. Chen, Y. Ong (Eds.), Proceedings of the International Conference on Advances in Natural Computation (ICNC), LNCS, vol. 3612, Springer-Verlag, 2005, pp. 845–854.
[58] G. Wena, J. Wei, J. Wang, T. Zhoub, L. Chen, Cognitive gravitation model for classification on small noisy data, Neurocomputing 118 (2013) 245–252.
[59] D. Wilson, T.R. Martínez, Improved heterogeneous distance functions, J. Artif. Intell. Res. (JAIR) 6 (1997) 1–34.
[60] S.P. Wright, Adjusted *p*-values for simultaneous inference, Biometrics 48 (1992) 1005–1013.
[61] W.E. Wright, Gravitational clustering, Pattern Recognit. 9 (1977) 151–166.
[62] J. Xu, Multi-label weighted *k*-nearest neighbor classifier with adaptive weight estimation, in: Proceedings of the International Conference on Neural Information Processing (ICONIP), in: LNCS, vol. 7073, Springer-Verlag, Berlin/Heidelberg, 2011, pp. 79–88.
[63] R.R. Yager, Veristic variables, IEEE Trans. Syst. Man Cybern. 30 (2000) 71–84.
[64] B. Yang, L. Peng, Y. Chen, H. Liu, R. Yuan, A DGC-based data classification method used for abnormal network intrusion detection, in: I. King, J. Wang, L.W. Chan, D. Wang (Eds.), Proceedings of the International Conference on Neural Information Processing (ICONIP), LNCS, vol. 4234, Springer-Verlag, Berlin/Heidelberg, 2006, pp. 209–216.
[65] S. Yang, S. Kim, Y. Ro, Semantic home photo categorization, IEEE Trans. Circuits Syst. Video Technol. 17 (2007) 324–335.
[66] Z. Younes, F. Abdallah, T. Denceux, Multi-label classification algorithm derived from *k*-nearest neighbor rule with label dependencies, in: Proceedings of the Sixteenth European Signal Processing Conference, Lausanne, Switzerland, 2008, pp. 297–308.
[67] Z. Younes, F. Abdallah, T. Denceux, Evidential multi-label classification approach to learning from data with imprecise labels, in: Computational Intelligence for Knowledge-Based Systems Design, in: LNCS, vol. 6178, Springer-Verlag, Berlin/Heidelberg, 2010b, pp. 119–128.
[68] Z. Younes, F. Abdallah, T. Denceux, Fuzzy multi-label learning under veristic variables, in: Proceedings of the International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2010a, pp. 1–8.
[69] M.L. Zhang, Z.H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (2006) 1338–1351.
[70] M.L. Zhang, Z.H. Zhou, ML-*k*NN: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

TITLE:

Effective active learning strategy for multi-label learning

AUTHORS:

O. Reyes, C. Morell, and S. Ventura



Neurocomputing, *submitted, 2015*

RANKING:

Impact factor (JCR 2015): 2.392

Knowledge area:

Computer Science, Artificial Intelligence: 31/130

Effective active learning strategy for multi-label learning

Oscar Reyes[a], Carlos Morell[b], Sebastián Ventura[c,d,*]

[a]*Department of Computer Science, University of Holguín, Cuba*
[b]*Department of Computer Science, Universidad Central de Las Villas, Cuba*
[c]*Department of Computer Science and Numerical Analysis, University of Córdoba, Spain*
[d]*Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia*

Abstract

Data labelling is an expensive process that requires expert handling. In multi-label data, data labelling is further complicated owing to experts must label each example several times, as each example belongs to various categories. Active learning is concerned with learning accurate classifiers by choosing which examples will be labelled, reducing the labelling effort and the cost of training an accurate model. This paper presents a new active learning strategy for working on multi-label data. Two uncertainty measures based on the predictions of base classifier and the inconsistency of a predicted label set regarding the label dimension of the labelled dataset, respectively, are defined to select the most uncertain examples. The proposed strategy was evaluated and compared to several state-of-the-art strategies on 18 datasets. The experimental results were validated using non-parametric statistical tests and confirmed the effectiveness of the proposal for better multi-label active learning.

Keywords: Multi-label classification, label ranking, multi-label active learning, active learning strategy, pool-based scenario, rank aggregation problem

1. Introduction

In recent years, the study of problems that involve data associated with more than one label at the same time has attracted a great deal of attention. Particular multi-label problems include text categorization [1–3], classification of emotions evoked by music [4], semantic annotation of images [5–7], classification of music and videos [8–10], classification of protein and gene function [11–16], acoustic classification [17], chemical data analysis [18] and many more.

Multi-label learning is concerned with learning a model that correctly generalizes unseen multi-label data. In multi-label learning, two tasks have been studied [19–21]: Multi-label Classification and Label Ranking. Multi-label Classification task aims to find a model where, for a given test instance, the label space is divided into relevant and irrelevant label sets. On the other hand, Label Ranking task aims to provide, for a given test instance, a ranking of labels according to their relevance values. In the literature, is named Multi-label Ranking task [22] the generalization of Multi-label Classification and Label Ranking tasks. Multi-label Ranking aims to produce, at the same time, both a bipartition of label space and a consistent ranking of labels.

Most multi-label learning algorithms that have been proposed in the literature are designed for working on supervised learning environments, i.e. scenarios where all training instances are labelled. However, data labelling is a very expensive process that requires expert handling. In multi-label data, experts must label each example several times, as each example belongs to various categories. The situation is further complicated when a multi-label problem with a large number of examples and label classes is analyzed. Consequently, several real scenarios nowadays contain a small number of labelled data and a large number of unlabelled data simultaneously.

To date, there are two main areas that are concerned with learning models from labelled and unlabelled data, known as Semi-Supervised Learning [23] and Active Learning [24]. Active Learning is concerned with learning better classifiers by choosing which instances are labelled for training. Consequently, the labelling effort

*Corresponding author. Tel:+34957212218; fax:+34957218630.
Email addresses: oreyesp@facinf.uho.edu.cu (Oscar Reyes), cmorellp@uclv.edu.cu (Carlos Morell), sventura@uco.es (Sebastián Ventura)

and the cost of training an accurate model are reduced. Active learning methods are involved in the acquisition of their own training data. A selection strategy iteratively selects instances from the unlabelled set that seem to be the most informative. Afterwards, an oracle annotates the selected instances and they are inserted in the set of labelled data. [24]

In this work, we focus in active learning scenarios in which a large collection of unlabelled data and a small set of labelled data are available, known as pool-based active learning [25]. In a pool-based scenario, the entire unlabelled set is evaluated and the instances are ranked before the selection of the most informative instances.

Active learning has proved be useful in several domains [26–31]. For more than a decade, a considerable number of active learning methods for single-label data have been proposed, for an interesting survey see [24]. However, the development of active learning methods for multi-label data has been scarce. The main challenge in performing active learning on multi-label data is designing effective strategies that measure the unified informative potential of an unlabelled instance across all labels [32]. The most significant works related to multi-label active learning have appeared in the following literature: [32–40].

Most state-of-the-art multi-label active learning strategies employ the Binary Relevance [19] approach to break down a multi-label problem into several binary classification tasks. The scarce literature on multi-label active learning is primarily focused on text and image classification. In addition, the active learning strategies are generally tested on the Multi-label Classification task. However, how they perform regarding the Label Ranking task has not been considered.

In this work, an effective multi-label active learning strategy is proposed, known as Uncertainty Sampling based on Category Vector Inconsistency and Ranking of Scores (CVIRS). Two measures of uncertainty from the perspectives of the predictions of base classifier and the inconsistency of a predicted label set regarding the label dimension of the labelled dataset, respectively, are defined. To compute the unified uncertainty of an unlabelled instance, a rank aggregation problem based in the difference margin on predictions of classifier is formulated. On the other hand, the inconsistency of a predicted label set, for a given unlabelled instance, is computed by means of the distance to the label sets of the labelled instances.

The experiments were carried out on 18 multi-label datasets, considering different problem domains, numbers of instances, features and labels. Several multi-label evaluation measures were used to analyze different viewpoints. The experimental stage showed the effectiveness of the proposal, obtaining significantly better results than previous multi-label active learning strategies. The experimental study included a statistical analysis based on non-parametric tests as proposed in [41, 42].

To the best of our knowledge, this paper presents a first attempt to compare the most significant multi-label active learning strategies over a sizeable number of multi-label datasets and two multi-label learning tasks, Multi-label Classification and Label Ranking. Also, not only the learning curves of the active learning strategies were observed through a visual inspection, but the active learning strategies were also compared through non-parametric statistical tests, resulting in a more robust analysis. As far as we understand, this is the first attempt to propose a multi-label active learning strategy that computes the uncertainty of unlabelled instances by means of a rank aggregation method.

This paper is arranged as follows: Section 2 briefly describes the state-of-the-art multi-label active learning strategies that have appeared in the literature. Section 3 presents the basis of our proposal. Section 4 describes the experimental set-up and analyses the experimental results. Finally, section 5 provides some concluding remarks.

2. Related works

In [33], two multi-label active learning strategies, named Max Loss (ML) and Mean Max Loss (MML), are proposed. The two active learning strategies select the unlabelled instances which have the maximum or mean loss value over the predicted classes. A threshold is computed, for each label, to predict whether an unlabelled instance belongs to the associated label or not. MML strategy considers the multi-label information taking into account the loss produced in each label. ML strategy calculates the loss value only on the label predicted with the most certainty. The effectiveness of the approach was proved in two multi-label datasets for image classification.

In [34], the Binary Minimum (BinMin) strategy is proposed. BinMin selects the unlabelled instance which, considering a target label, minimises the distance among the restricting hyperplane and the center of the maximum radius hyperball. The effectiveness of the approach was proved in one multi-label dataset for text classification.

In [36], an active learning strategy named Maximum Loss Reduction with Maximal Confidence (MMC) is

suggested. MMC is based on the principles of Expected Error Reduction [24]; it selects those unlabelled instances that maximize the reduction rate of the expected model loss. MMC predicts the number of labels for an unlabelled instance following a process known as "LR-based prediction method". In each iteration, the labelled set is transformed into a single-label dataset and a Logistic Regression classifier is trained. For each unlabelled instance, a label vector is predicted which is used to calculate the uncertainty of the instance. MMC is computationally expensive in large-scale multi-label datasets. The effectiveness of the approach was proved in seven multi-label datasets for text classification.

In [35], the Two Dimensional Active Learning (2DAL) strategy for image classification is presented. 2DAL selects the sample-label pairs to minimize a multi-label bayesian classification error bound, reducing the sample and label redundancies simultaneously. The authors presume that for a selected unlabelled instance, only some labels need to be annotated, whilst the other labels can be inferred through the label correlations. 2DAL tries to reduce the sample and label redundancies simultaneously by selecting the most informative sample-label pairs. 2DAL annotates a subset of labels of an instance; to do this, the authors propose a method known as "Kernelized Maximum Entropy Model" which models label correlations. The effectiveness of the approach was evaluated in two benchmark datasets.

In later work [37], the 2DAL strategy is extended. An online multi-label classification algorithm is introduced to avoid the continuous retraining of the classifier. The effectiveness of this extension was evaluated on two benchmark datasets and one real-world image collection. In theory, 2DAL can be used in any multi-label scenario. However, it is ideal in scenarios where the instances are partially labelled, that is to say, when the set of labels of an instance is not completely unknown.

Furthermore, in [38], the 2DAL strategy is extended to multi-view setting. The extension combines multi-view learning and active learning for image classification. It takes into account information on labels and view dimensions to select the most informative instances. In multi-view learning, multiple classifiers are trained similar to Query by Committee methods [43], in that the disagreement among classifiers is used to improve the classification performance. The authors define an intra-view uncertainty that is used to determine the most informative sample-label pairs for each view. On the other hand, an inter-view uncertainty is defined which represents the uncertainty along the views. The overall uncertainty of a sample-label pair is computed as the summation of the intra-view and inter-view uncertainties. This extension of the 2DAL strategy was proved in one multi-label image dataset.

In [39], a general framework for multi-label active learning is proposed. In this framework, three dimensions are defined: Evidence, Class and Weight. A strategy is constructed selecting one alternative for each dimension. Two choices are available for the Evidence dimension, three for the Class dimension and two for the Weight dimension, giving a total of twelve different active learning strategies. The alternatives in the Evidence dimension represent the type of evidence to use. The alternatives of the Class dimension represent how to combine the values of a vector. The alternatives in the Weight dimension show whether all labels are treated equally or not. The effectiveness of the twelve active learning strategies was proved on two multi-label datasets for text classification. The results showed that the CMN strategy obtains the best results. CMN is constructed by selecting the confidence of predictions (C) as type of evidence, the minimum (M) choice which selects the minimum value of a confidence vector, and the no-weighting (N) choice which treats all labels alike.

In [32], the Max-Margin Prediction Uncertainty (MMU) and Label Cardinality Inconsistency (LCI) strategies are proposed. MMU models the uncertainty of an instance taking into account the separation margin between the predicted groups of positive and negative labels. MMU defines a measure named "Max-Margin Prediction Uncertainty" for computing the uncertainty of an instance. On the other hand, the LCI strategy uses the average number of positive labels assigned to each instance (label cardinality) to measure the uncertainty of an unlabelled instance from the label space perspective. The uncertainty of an instance is defined as the distance between the number of predicted positive labels and the label cardinality of the current labelled set. The effectiveness of these active learning strategies was proved on three multi-label datasets for image classification and one dataset for text classification.

In [40], the strategy known as Active Learning based on Uncertainty and Diversity for Incremental Multi-label Learning (AUDI) is proposed. The authors propose an incremental multi-label classification model to avoid the retraining of the classifier with all labelled examples after every iteration. The criterion used by LCI strategy [32] is extended in order to identify the most uncertain instances in the instance space. AUDI selects the instance-label pairs which can improve the designed multi-label classification model. The effectiveness of the AUDI strategy was proved on nine multi-

label datasets.

Generally speaking, many state-of-the-art multi-label active learning strategies employ the Binary Relevance [19] approach to break down a multi-label problem into several binary classification tasks. Some of them simply extend the binary uncertainty concept for multi-label scenarios by aggregating the value associated with each label, e.g. taking the minimum [34] or the average value over all labels [33, 36]. The bibliographic revision reveals that most multi-label active learning strategies have been tested with BR-SVM as base classifier, i.e. the Binary Relevance approach using binary SVM classifiers [32–34, 36].

Commonly, previous works use the Micro-Average F_1-measure [32, 36] and Macro-Average F_1-Measure [32, 35, 37–39] to evaluate the effectiveness of the active learning strategies. The Micro-Average F_1-measure and Macro-Average F_1-Measure are measures associated with Multi-label Classification task. However, the way that these active learning strategies perform on different evaluation measures for the Multi-label Classification task has not been analyzed. On the other hand, evaluation measures for the Label Ranking task have not been considered.

3. A new multi-label active learning strategy

On multi-label data, let $\mathcal{F}$ be a feature space and $\mathcal{L}$ a label space with cardinality equal to q (number of labels). A multi-label instance i is represented as a tuple $\langle \mathbf{X_i}, \mathbf{Y_i} \rangle$, where $\mathbf{X_i}$ is the feature vector and $\mathbf{Y_i}$ the category vector of the instance i. Let us say $\mathbf{Y_i}$ is a binary vector that contains q components, where component $Y_{i\ell}$ represents whether the instance i belongs to the ℓ-th label or not.

A multi-label classifier Φ that resolves Multi-label Classification and Label Ranking tasks at the same time, for a given test instance, (i) partitions the label space $\mathcal{L}$ into relevant label set (positive labels) and irrelevant label set (negative labels), and also (ii) returns a ranking of labels according to their relevance.

The multi-label learning algorithms can be divided into two main categories [19, 21]: the problem transformation methods and algorithm adaptation methods. The problem transformation methods transform a multi-label dataset into one or more single-label datasets. Afterwards, for each transformed dataset a single-label classifier is executed, and finally, an aggregation strategy is performed. The algorithm adaptation category groups together algorithms designed to directly handle the multi-label data.

On pool-based active learning scenarios, we have a small set of labelled data L_s and a large set of unlabelled data U_s.

Uncertainty measure based on rank aggregation

Let Φ be a multi-label classifier which, for a given unseen instance, returns probabilities for each possible label $\ell \in \mathcal{L}$. On multi-label data, we have a probability that an instance i belongs to the ℓ-th label ($P_\Phi(\ell{=}1|i)$) and a probability that i does not belong to the ℓ-th label ($P_\Phi(\ell{=}0|i)$).

So, the difference margin in predictions of classifier Φ with respect to whether the given instance i belongs or does not belong to the ℓ-th label can be computed as follows:

$$m_\Phi^{i,\ell} = |P_\Phi(\ell{=}1|i) - P_\Phi(\ell{=}0|i)| \quad (1)$$

An instance with large margin value on ℓ-th label means that the classifier Φ has little doubt in differentiating whether the instance belongs or does not belong to label ℓ. On the other hand, an instance with small margin value on ℓ-th label means that it is more ambiguous for the current classifier to predict whether the instance belongs or does not belong to label ℓ. So, given an unlabelled instance i and a classifier Φ, we can obtain a vector of margin values $\mathbf{M_\Phi^i} = \left\langle m_\Phi^{i,1}, m_\Phi^{i,2}, \ldots, m_\Phi^{i,q} \right\rangle$, one margin value for each label $\ell \in \mathcal{L}$. The problem is therefore how to properly aggregate the multi-label information for computing the unified informative value of an unlabelled instance.

We consider that for computing the utility of an unlabelled instance, not only should its margin vector be considered, but also information regarding all unlabelled instances. We focus on pool-based active learning scenarios, where a vector of margin values for each unlabelled instance $i \in U_s$ can be obtained.

In this work, we propose to handle the aggregation of the margin values as a *Rank Aggregation* problem [44, 45]. Therefore, implicitly, multi-label information will be taken into account.

Let $\mathbf{M_\Phi^{i_1}}, \mathbf{M_\Phi^{i_2}}, \ldots, \mathbf{M_\Phi^{i_{|U_s|}}}$ be the margin vectors of the unlabelled instances $i_1, i_2, \ldots, i_{|U_s|}$, respectively. Given the margin vector of each unlabelled instance, q rankings of instances $\tau_1, \tau_2, \ldots, \tau_q$ are computed; one ranking for each label $\ell \in \mathcal{L}$. Given a label ℓ, the ranking of unlabelled instances τ_ℓ is computed as follows:

$$\tau_\ell = (i_{\pi 1}, i_{\pi 2}, \ldots, i_{\pi |U_s|}) \mid m_\Phi^{i_{\pi 1},\ell} < m_\Phi^{i_{\pi 2},\ell} \ldots < m_\Phi^{i_{\pi |U_s|},\ell} \quad (2)$$

A ranking $\tau_\ell = (i_{\pi 1}, i_{\pi 2}, \ldots, i_{\pi |U_s|})$ is an ordering (permutation or full list) of the unlabelled instances according to their margin values on the ℓ-th label. So, we want

to find a ranking of instances τ' which combines the information from the rankings $\tau_1, \tau_2, \ldots, \tau_q$, in such a manner that the instances placed in the first positions of the final ranking τ' correspond to the most uncertain instances.

Several rank aggregation methods have been proposed in the literature [44–46]. There is a long history of theoretical works arising from the rank aggregation problem. The rank aggregation obtained is called *Kemeny optimal aggregation (KOA)* when the Kendall's tau distance among the final ranking τ' and the source rankings is optimized. On the other hand, if the Spearman's footrule distance is minimised the rank aggregation obtained is called *Footrule optimal aggregation (FOA)*. In [44] a polynomial time algorithm to compute *FOA* for full lists is proposed. They also prove that the *KOA* problem is NP-hard even when the number of lists is equal to four.

Since we focus on pool-based active learning where a large number of unlabelled instances are available, and where it is also nowadays common to find multi-label datasets with a large number of labels, we do not consider it practical to use sophisticated rank aggregation methods. The oracle, e.g. a human annotator, would have to wait a considerable time before labelling the most uncertain instances. Instead, we propose to use the simplest rank aggregation method, Borda's method.

Borda's method is a positional method, assigning a score corresponding to the positions in which an instance appears within each ranking [44]. The advantage of the positional methods for rank aggregation is that they are computationally efficient. However, the positional methods neither optimise any distance criteria nor satisfy *Condorcet*'s criterion. *Condorcet*'s criterion states that if an element defeats every other element in pairwise majority voting, this element should be ranked first [44].

Based on Borda's method, the score of an instance i is computed as follows:

$$s(i)=\frac{\sum_{\ell\in\mathcal{L}}(|U_s|-\tau_\ell(i))}{q(|U_s|-1)} \tag{3}$$

where $\tau_\ell(i)$ returns the position of the instance i in the ranking τ_ℓ. The greater the value of $s(i)$, the greater uncertainty of the instance i taking the information across all labels.

Category vector inconsistency

The defined uncertainty measure based on rank aggregation does not take into account the possible ties, instances which have equal margin values, that can appear when a ranking of unlabelled instances is computed for a certain label. On the other hand, the uncertainty measure based on rank aggregation does not satisfy *Condorcet*'s criterion, therefore if a multi-label dataset with a large number of labels is considered, then the score function used to compute the uncertainty of an unlabelled instance may fail in representing the unified uncertainty across all labels.

Consequently, in addition to the defined uncertainty measure based on rank aggregation, we consider important to take into account the inconsistency of a predicted label set in computing the uncertainty of an unlabelled instance, by exploiting some measure with respect to the label dimension of current labelled set. As the labelled set and unlabelled set are drawn from the same underlying distribution, is expected that predicted label sets and the label sets of labelled instances share common properties.

In [32], a similar idea is proposed. The LCI strategy defines a measure based on label cardinality inconsistency. The authors reveal that the multi-label instances usually have similar number of positive labels (label cardinality); therefore the LCI strategy selects the unlabelled instances that have maximum difference among the number of predicted positive labels and the label cardinality of the current labelled set. The measure based on label cardinality inconsistency works well when the base classifier predicts positive labels that are actually true positive labels. However, the measure fails when the number of predicted positive labels is approximately equal to the label cardinality of labelled set, and the prediction of base classifier is totally wrong, i.e. predicted positive labels are actually negative labels. In this case, a small value of label cardinality inconsistency is assigned, and the LCI strategy considers that the instance is not informative for the current classifier.

Instead of using the label cardinality as measure of inconsistency, in this work we propose to use a measure based on category vector inconsistency, by taking the difference among the predicted label set of an unlabelled instance and the label sets of labelled instances. As the labelled set and unlabelled set are drawn from the same underlying distribution, it is expected that a predicted label set is similar to label sets existing in the labelled set L_s.

Table 1 shows a contingency table created given the category vectors $\mathbf{Y_i}$ and $\mathbf{Y_j}$ of the instances i and j, respectively. Let a be the number of components where $Y_{i\ell}=Y_{j\ell}=1$, b is the number of components where $Y_{i\ell}=1$ and $Y_{j\ell}=0$, c is the number of components where the $Y_{i\ell}=0$ and $Y_{j\ell}=1$, and d is the number of components

where $Y_{i\ell}$=$Y_{j\ell}$=0.

$\mathbf{Y_i} \diagdown \mathbf{Y_j}$	1	0
1	a	b
0	c	d

Table 1: Contingency table given two category vectors.

Several distances and similarity functions to compare binary vectors have been proposed in the literature. The Hamming distance is one of the most popular and simple distance for binary data. Given the category vectors $\mathbf{Y_i}$ and $\mathbf{Y_j}$, the normalized Hamming distance is computed as follows:

$$d_H(\mathbf{Y_i},\mathbf{Y_j})=\frac{b+c}{q} \tag{4}$$

where q is the number of labels.

The Hamming distance is an important tool in algebraic code theory. The Hamming distance counts the number of components for which two binary vectors differ. In our case, given the category vectors of two instances, the Hamming distance returns the numbers of cases for which an instance belongs to a label and the other instance does not belong to the same label. However, we do not only want to count the number of components for which two category vectors differ, but we also consider it important to measure the difference among the structures of category vectors. The label sets that more commonly appear in a multi-label dataset form structures (combinations of zeros and ones) that we can commonly found in the category vectors of the labelled instances.

In [47], an entropy distance is defined to compute the differences among the structures of two binary vectors. It implicitly uses the Hamming distance. The normalized entropy distance (d_E) among two category vectors $\mathbf{Y_i}$ and $\mathbf{Y_j}$ is computed as follows:

$$d_E(\mathbf{Y_i},\mathbf{Y_j})=\frac{2H(\mathbf{Y_i},\mathbf{Y_j})-H(\mathbf{Y_i})-H(\mathbf{Y_j})}{H(\mathbf{Y_i},\mathbf{Y_j})} \tag{5}$$

where the joint entropy among $\mathbf{Y_i}$ and $\mathbf{Y_j}$ is computed as follows:

$$H(\mathbf{Y_i},\mathbf{Y_j})=H_4(\frac{a}{q},\frac{b}{q},\frac{c}{q},\frac{d}{q})$$

According to the properties of the discrete entropy, H_4 is equal to:

$$H_4(\frac{a}{q},\frac{b}{q},\frac{c}{q},\frac{d}{q})=H_2(\frac{b+c}{q},\frac{a+d}{q})+\frac{b+c}{q}H_2(\frac{b}{b+c},\frac{c}{b+c})+\frac{a+d}{q}H_2(\frac{a}{a+d},\frac{d}{a+d})$$

The entropy of a category vector $\mathbf{Y}$ is computed as follows:

$$H(\mathbf{Y})=H_2(\frac{w}{q},\frac{s}{q})=-\frac{w}{q}log_2(\frac{w}{q})-\frac{s}{q}log_2(\frac{s}{q})$$

where w and s are the numbers of ones (positive labels) and zeros (negative labels), respectively, in the category vector $\mathbf{Y}$.

Based in the d_H and d_E distance functions, the inconsistency of the predicted category vector, for a given unlabelled instance i, is computed as follows:

$$v(i)=\frac{1}{|L_s|}\sum_{j\in L_s} f_u(\mathbf{Y_i},\mathbf{Y_j}) \tag{6}$$

$$f_u(\mathbf{Y_i},\mathbf{Y_j})=\begin{cases} d_E(\mathbf{Y_i},\mathbf{Y_j}) & d_H(\mathbf{Y_i},\mathbf{Y_j})<1 \\ 1 & d_H(\mathbf{Y_i},\mathbf{Y_j})=1 \end{cases}$$

where $\mathbf{Y_i}$ is the category vector of the instance i predicted by the current classifier Φ. $\mathbf{Y_j}$ is the category vector of the instance j that belongs to the labelled set of instances L_s.

The entropy distance recognizes the existing structures (patterns) among two binary vectors. It is more flexible than Hamming distance. For example, given two category vectors $\mathbf{Y_i}$=[010101] and $\mathbf{Y_j}$=[101010], $d_H(\mathbf{Y_i},\mathbf{Y_j})$=1, owing to $\mathbf{Y_i}$ and $\mathbf{Y_j}$ differ in all their components. However, $d_E(\mathbf{Y_i},\mathbf{Y_j})$ is equal to 0 so $\mathbf{Y_i}$ and $\mathbf{Y_j}$ have the same structure, the alternation of two symbols. Hence, f_u function returns the maximum value when $d_H(\mathbf{Y_i},\mathbf{Y_j})$=1.

Active learning strategy

Uncertainty sampling is one of the most simple and commonly used active learning strategies [25]. This type of strategy selects those unlabelled instances which are least certain for the current base classifier. Based in the two measures above defined, the most uncertain instance from the unlabelled set of instances U_s is selected as follows:

$$\underset{i\in U_s}{\operatorname{argmax}} \quad s(i)\cdot v(i) \tag{7}$$

We call this new active learning strategy, Uncertainty sampling based on Category Vector Inconsistency and Ranking of Scores (CVIRS). This active learning strategy selects the unlabelled instance that has the most unified uncertainty computed by means of the rank aggregation problem formulated, and at the same time, it has the most dissimilar category vector predicted with respect to the category vector of the labelled instances.

This approach must be used with probabilistic learning models, although it can be used with base classifiers which can obtain proper probability estimates from their outputs. Our proposal is not restricted to base classifiers that use problem transformation methods, it can also be used with multi-label learning algorithms that belong to algorithm adaptation category.

Note that our proposal is in some ways related with Density-Weighted methods. Density-Weighted methods consider that the most informative instance should not only be uncertain, but it should also be "representative" of the underlying distribution [24].

Regarding the computational complexity to compute the score of an unlabelled instance by means of the rank aggregation problem formulated, let $f_{ts}(\Phi)$ be the cost function for multi-label classifier Φ to classify an unlabelled instance. To compute the margin vector for each unlabelled instance, $i \in U_s$, $O(|U_s| \cdot f_{ts}(\Phi))$ steps are needed. To compute the q rankings of unlabelled instances, $O(q \cdot |U_s|^2)$ steps are needed. Although, if an efficient sort algorithm is used, then the computational complexity could be reduced to $O(q \cdot |U_s| \cdot \log(|U_s|))$ steps. In addition, in order to reduce the computational complexity of computing the q rankings, only a subset of the U_s could be considered.

Regarding the computational complexity to compute the inconsistency of a category vector predicted for a given unlabelled instance, $O(f_{ts}(\Phi))$ steps are needed to predict the category vector of an unlabelled instance. To compute the differences among a predicted category vector and the category vector of each labelled instance, $i \in L_s$, $O(q \cdot |L_s|)$ steps are needed.

The CVIRS strategy requires $O(max(|U_s| \cdot f_{ts}(\Phi), q \cdot |U_s|^2))$ steps to determine the utility of an unlabelled instance, owing to $|U_s| \cdot f_{ts}(\Phi) >> f_{ts}(\Phi)$ and $q \cdot |U_s|^2 >> q \cdot |L_s|$.

4. Experimental study

In this section, the means by which multi-label models and active learning strategies are evaluated, a description of the multi-label datasets, and other settings used in the empirical study are explained. Finally, the experimental results on different datasets and the statistical analysis are discussed.

4.1. Evaluation of multi-label models and active learning strategies

In this work, several evaluation measures that have been suggested in [19–21] were used. In multi-label setting additional degrees of freedom are introduced, therefore for evaluating multi-label methods it is essential to include several measures that allow analyse different point of views. In [19], the measures to evaluate multi-label learning algorithms are divided into two categories: Label-based measures and Example-based measures. The Example-based measures are further categorized into Ranking-based and Bipartition-based measures. Label-based measures compute a single-label measure for each label and then an average value is obtained. Example-based measures are calculated for each test instance and an average value across the test set is computed.

The Label-based measures used in this work are the Micro-Average F_1-Measure (M_{iF_1}) and Macro-Average F_1-Measure (M_{aF_1}). The micro approach aggregates the true positive, true negative, false positive, and false negative values of all labels and then calculates the measure. The macro approach computes one measure for each label and then the values are averaged over all labels. The Bipartition-based measures used in this work are the Hamming Loss (H_L) and Example-based F_1-Measure (F_{1Ex}). H_L averages the symmetrical differences among the predicted and actual label sets, while F_{1Ex} calculates the F_1-Measure over all examples in the test set. The Ranking-based measures used in this work are the Ranking Loss (R_L), Average Precision (A_P) and One Error (O_E). R_L averages the proportion of label pairs that are incorrectly ordered. A_P averages how many times a particular label is ranked above another label which is in the true label set. O_E averages how many times the top-ranked label is not in the set of true labels of the instance.

The M_{iF_1}, M_{aF_1}, H_L and F_{1Ex} evaluation measures are associated with the Multi-label Classification task, whereas R_L, A_P and O_E are related with the Label Ranking task. The higher the value of M_{iF_1}, M_{aF_1}, F_{1Ex} and A_P, and the lower the value of H_L, R_L and O_E, the better the performance of a multi-label learning algorithm. A formal definition of these evaluation measures can be consulted in [19–21].

Generally speaking, active learning strategies are evaluated by constructing learning curves, plotting an evaluation measure as a function of the number of labelled instances that exist in the labelled set. Through visual inspection, a strategy is superior to the alternatives if it dominates them for most of the points along their learning curves [24]. This is a mere visual inspection of the learning curves, providing a qualitative intuition of which active learning strategy performs better. However, visually comparing several learning curves can be very confusing, as several intersections among the learning curves could occur.

In addition to a visual inspection, the area under learning curve to compare several active learning strategies in a quantitative manner was used. To analyse and validate the results, several non-parametric statistical tests were used, as proposed in [41, 42]. Friedman's test [48] was performed to evaluate whether there was significant differences in the results. If Friedman's test indicated that the results were significantly different, the Shaffer post-hoc test [49] was used to perform multiple comparisons among all methods, as proposed in [42].

4.2. Experimental setting

In the experimental study, our proposal CVIRS was compared to BinMin [34], ML [33], MML [33], MMC [36], CMN [39], MMU [32], LCI [32] and Random strategies. Random strategy randomly chosees the instances from the available unlabelled set.

For the sake of fairness, the 2DAL [35] and AUDI [40] strategies were not included in the comparison. 2DAL strategy is optimal in scenarios where the instances are partially labelled. On the other hand, these two AL strategies have been designed to work in conjunction with their incremental multi-label classification models to avoid the retraining of the classifier with all labelled examples after every iteration.

In this work, the conventional batch learning scenario was used, where all labelled instances are required as a prerequisite for training. For the sake of fairness, in the experiments, the active learning strategies used the Binary Relevance method with binary Support Vector Machine in each label (BR-SVM) as their base classifier, since most state-of-the-art multi-label active learning strategies have been tested with BR-SVM as base classifier. A linear kernel and a penalty parameter equal to 1.0 were used, as proposed in [36].

The active learning strategies and BR-SVM algorithm were implemented on MULAN [50]. MULAN is a Java library which contains several methods and evaluation measures for the multi-label learning paradigm. We use the SMO algorithm for SVMs as presented by [51]. Logistic regression models are fitted to the outputs of SVMs to obtain proper probability estimates. The algorithms as standalone runnable files are available in order to facilitate the replicability of the experiments[1].

For each possible combination of strategy and dataset, a 10-fold cross validation was used and the average value was calculated. In this work, for each fold execution the iterative experimental protocol described in Algorithm 1 was adopted. The 5% of the training set T_r was randomly selected to construct the labelled set L_s. Therefore, the initial classifier was trained with few labelled instances. The non-selected instances of T_r formed the unlabelled set U_s. For each instance $i \in U_s$, its label set was hidden. The maximum number of iteration β was set to 750. In each iteration, the multi-label classifier Φ determined the significance of the current L_s set in classifying the test set T_s. This experimental protocol is similar to previous experimental protocols used in [32, 36, 39, 40].

Algorithm 1: Experimental protocol.

```
Input : T_r → training set of multi-label instances
        T_s → test set of multi-label instances
        γ → multi-label active learning strategy
        θ → oracle for labelling unlabelled instances
        s → number of sampling instances
        β → maximum number of iterations
1  begin
2      //Construct the labelled and unlabelled set from T_r
3      L_s ← Resample (s, T_r);
4      U_s ← T_r / L_s;
5      for iter ← 1 to β do
6          //Train Φ with L_s
7          Φ ← Train (L_s,Φ);
8          //Evaluate the effectiveness of Φ on T_s
9          Test (T_s, Φ);
10         //Select the most informative instance from U_s
11         i ← SelectInformativeInstance (γ, Φ, U_s);
12         //Label the selected instance
13         Label (θ,i);
14         //Update the labelled and unlabelled sets
15         L_s ← L_s ∪ {i} ;
16         U_s ← U_s / {i} ;
17     end
18 end
```

The state-of-the-art active learning strategies discussed in section 2 and our proposal are not optimal when working in batch-mode active learning scenarios. Batch-mode active learning allows the learner to select a set of instances in each iteration. Minimizing the redundancy of information among the selected instances is one of main challenges in batch-mode active learning. The strategies considered in this work can select a set of unlabelled instances following a myopic approach, e.g. by selecting the "p-best" instances from the unlabelled set. However, following a myopic approach the redundancy of information among the "p-best" instances is not considered [24]. For the sake of fairness, in this work the active learning strategies selected only one unlabelled instance in each iteration.

The labelling process was done in a simulated environment, i.e. the oracle reveals the hidden label set of an unlabelled instance and the instance is added to the L_s set. A pool-based scenario for selecting the most

[1] http://www.uco.es/grupos/kdis/kdiswiki/MLAL

informative instance in each iteration was used. In a pool-based scenario the entire U_s set is tested and the instances are ranked before selection of the most informative instance [24].

In the experiments, 18 real multi-label datasets were used[2]. Multi-label datasets with different scales and from different application domains were included to analyse the behaviour of the multi-label active learning strategies in datasets with diverse properties.

The datasets come from five domains: image, text, biology, music and audio. The Flags [7] dataset stores examples about nations and their national flags. Emotions [52] stores examples of songs according to the emotions that they evoke. Birds [17] contains examples of multiple bird species for acoustic classification. Cal500 [9] contains pieces of music for semantic annotation. Scene [8] contains a series of patterns about kinds of landscapes. The Corel5k [53] and Corel16k [5] datasets contain Corel images. The Yeast [11] and Genbase [12] datasets come from the biological domain, including information about the function of genes and proteins. Medical [54] was used in the Medical Natural Language Processing Challenge in 2007. The Enron [55] dataset contains emails from 151 users. TMC2007-500 [2, 56] stores examples of reports of aviation safety. Bibtex [3] dataset contains bibtex examples for automatic tag suggestion. The Arts, Business, Entertainment, Recreation and Health datasets come from the Yahoo text collection [57].

On the context of multi-label active learning, the Corel5k dataset was previously used in [32, 33, 40]. Emotions, Enron, Medical and Genbase datasets were used in [40]. Scene and Yeast datasets were previously used in [35, 37, 40]. Datasets from Yahoo text collection were used in [36].

Table 2 shows some statistics of the datasets. The label cardinality is the average number of labels per example. The label density is the label cardinality divided by the total number of labels. The label cardinality, label density and different subsets of labels are measures that represent the complexity of a multi-label dataset. A dataset which has a large number of labels may challenge a multi-label algorithm in many ways [19]. The datasets vary in size: from 194 up to 28,596 examples, from 10 up to 32,001 features, from 6 up to 374 labels, from 15 up to 4,937 different subset of labels, from 1.014 up to 26.044 label cardinality, and from 0.009 up to 0.485 label density.

[2]The datasets are available to download at http://mulan.sourceforge.net/datasets.html

Dataset	n	d	q	d_s	l_c	l_d
Flags	194	10	7	54	3.932	0.485
Emotions	593	72	6	27	1.869	0.311
Birds	645	260	19	133	1.014	0.053
Yeast	2417	103	14	198	4.237	0.303
Scene	2407	294	6	15	1.074	0.179
Cal500	502	68	174	502	26.044	0.150
Genbase	662	1186	27	32	1.252	0.046
Medical	978	1449	45	94	1.245	0.028
Enron	1702	1001	53	753	3.378	0.064
TMC2007-500	28596	500	22	1341	2.160	0.098
Corel5k	5000	499	374	3175	3.522	0.009
Corel16k	13811	500	161	4937	2.867	0.018
Bibtex	7395	1836	159	2856	2.402	0.015
Arts	7484	23146	26	599	1.654	0.064
Business	11214	21924	30	233	1.599	0.053
Entertainment	12730	32001	21	337	1.414	0.067
Recreation	12828	30324	22	530	1.429	0.065
Health	9205	30605	32	335	1.644	0.051

Table 2: Statistics of the benchmark datasets, number of instances (n), number of features (d), number of labels (q), different subsets of labels (d_s), label cardinality (l_c) and label density (l_d). The datasets are ordered by their complexity calculated as $n \times d \times q$.

4.3. Results and discussion

The empirical study was divided into two parts: a comparative study between the active learning strategies on the Multi-label Classification task and a comparison of the strategies on the Label Ranking task.

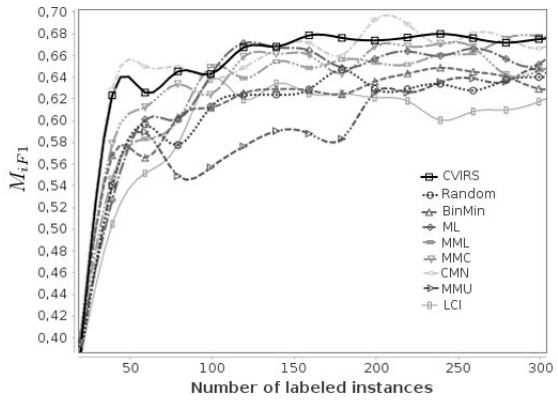
4.3.1. Multi-label Classification task

Figures 1-4 represent the learning curves for the active learning strategies considered on the Emotions, Medical, Yeast and TMC2007-500 datasets. Each graph represents a plot of a function on which its x-axis is the number of labelled instances and the y-axis is the value reached by the multi-label classifier for a certain evaluation measure.

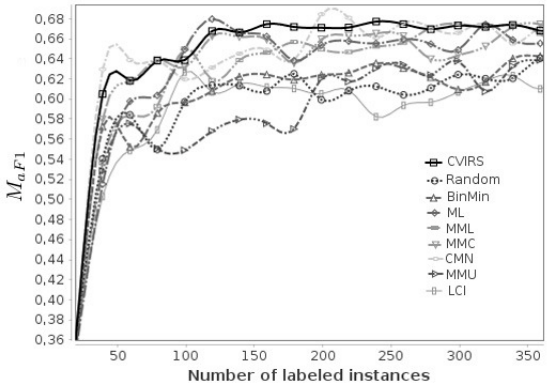
Through visual inspection, Figure 1 shows that CVIRS strategy obtained the best results for the M_{iF_1} and M_{aF_1} measures on the Emotions dataset. The CMN, ML, MML and MMC strategies performed better than Random strategy. The LCI and MMU strategies showed a poor performance on this dataset.

Figure 2 shows that the MMC, ML and MML strategies obtained worse results than the Random strategy on the Medical dataset. The CVIRS, CMN, MMU and LCI strategies showed the best performance.

Figure 3 shows that the CVIRS, CMN, LCI and BinMin strategies outperformed the rest of the strategies on the Yeast dataset for the H_L measure, their learning curves were under the learning curve of the Random strategy. The ML, MML and MMC strategies showed the worst performance. For the F_{1Ex} measure, the CVIRS, LCI, BinMin and MMU strategies showed the best performance, whereas MMC showed the worst results.

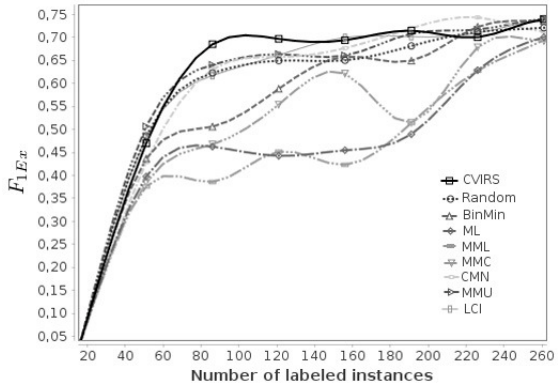


(a) Performance for M_{iF_1} measure.

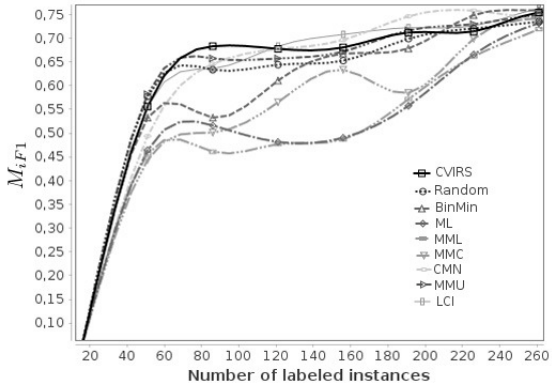


(b) Performance for M_{aF_1} measure.

Figure 1: Performance of the multi-label active learning strategies on the Emotions dataset.

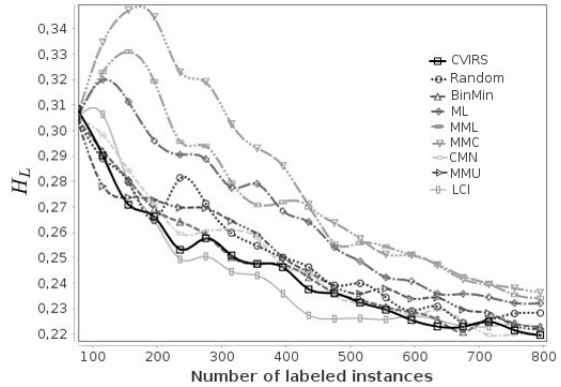


(a) Performance for F_{1Ex} measure.

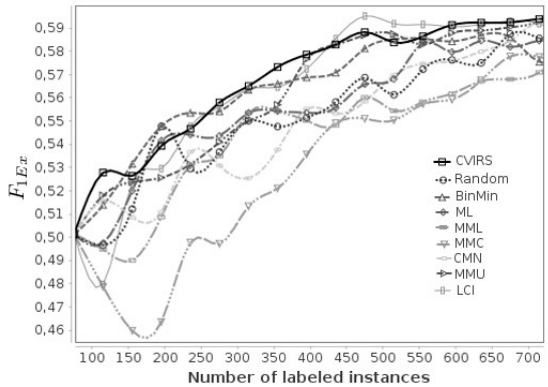


(b) Performance for M_{iF_1} measure.

Figure 2: Performance of the multi-label active learning strategies on the Medical dataset.



(a) Performance for H_L measure.



(b) Performance for F_{1Ex} measure.

Figure 3: Performance of the multi-label active learning strategies on the Yeast dataset.

Figure 4 shows that CVIRS strategy obtained a significantly better performance with respect to the rest of the active learning strategies on the TMC2007-500 dataset. For the F_{1Ex} measure, the CMN, BinMin, ML,

MML and MMC strategies showed a poor performance on this dataset - their learning curves were dominated by the learning curve of Random strategy. For the $M_{aF_{1Ex}}$ measure, the MMU, ML, MML and MMC strategies showed a poor performance.

Visual inspection of the learning curves provides a qualitative intuition of which active learning strategy performs better. However, it is often not a simple process, due to the overlapping among several learning curves. In order to compare several active learning strategies in a quantitative manner, the Area Under learning Curve (ALC) values were calculated, and statistical tests were carried out.

Tables 3-6 show the ALC results obtained by the nine strategies considered in the experimental study. In all cases, the best results are highlighted in bold typeface in the tables, "↓" indicates "the smaller the better", and "↑" indicates "the larger the better". In the tables, the last two rows show the average rank (Ave. Rank) and the ranking position (Pos.) for each strategy according to Friedman's test.

For the label-based (M_{iF_1} and M_{aF_1}) measures, CVIRS strategy generally showed a good performance on the 18 multi-label datasets. The MMU and LCI strategies showed a good performance on the Medical, Enron and Yeast datasets. CMN strategy obtained good results on the Birds, Genbase, Medical and Enron datasets. The BinMin and ML strategies obtained good results on the Corel5k, Corel16k and Yahoo Collection datasets.

For the bipartition-based (H_L and F_{1Ex}) measures, CVIRS strategy generally showed a good performance on the 18 multi-label datasets. The MMU and LCI strategies showed a good performance on the Medical and Yeast datasets. CMN strategy obtained good results on the Emotions, Birds, Genbase, Medical and Enron datasets. BinMin strategy obtained good results on the Flags, Yeast, Corel5k, Corel16k and Yahoo Collection datasets.

According to the average rankings returned by the Friedman's test, the three active learning strategies that obtained the best performance for the label-based and bipartition-based measures were CVIRS, CMN and BinMin, in this order. CVIRS strategy had the first position in the average rankings returned by the Friedman's test. The CMN and BinMin strategies had the second and third positions in the rankings, respectively. The MMC and Random strategies obtained the worst results.

Given that the p-values of Friedman's test were lower than the level of significance set as α=0.05, we concluded that there were significant differences among the observed ALC's values in the bipartition-based and label-based measures considered. Afterwards, a Shaffer's post-hoc test for all pairwise comparisons was carried out. In the statistical analysis, the adjusted p-values [58] were considered. The adjusted p-values take into account the fact that multiple tests are conducted and they can be compared directly with any significance level [42].

The statistical information obtained from the Shaffer's test was illustrated as a graph. An edge $\gamma_1 \rightarrow \gamma_2$ shows that strategy γ_1 outperforms strategy γ_2. Each edge was labelled with the evaluation measures which γ_1 outperformed the γ_2 method. The adjusted p-values of the Shaffer's test were indicated between parentheses. Figure 5 shows the results of the Shaffer's test for the bipartition-based and label-based measures.

From a statistical point of view, the proposed active learning strategy (CVIRS) significantly outperformed Random, ML, MML, MMC, LCI and MMU strategies on all the label-based and bipartition-based measures considered.

In the case of $M_{iF_{1Ex}}$ measure, the BinMin, CMN and LCI strategies significantly outperformed the Random strategy. The CMN and BinMin strategies performed better than MMC and MML strategies. Significant differences among the Random, MMC, ML, MML and MMU strategies were not detected for the significance level considered.

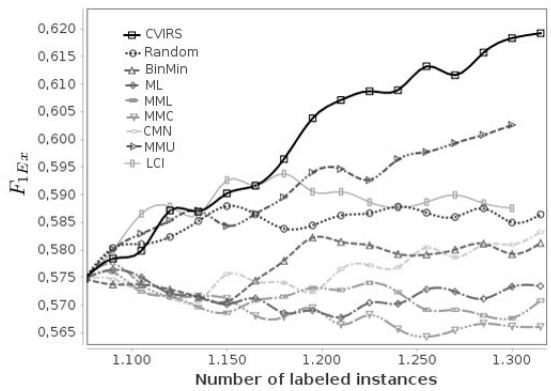
In the case of $M_{aF_{1Ex}}$ measure, the BinMin and CMN strategies significantly outperformed Random strategy. Furthermore, CMN strategy performed better than ML and MMC strategies. Shaffer's test did not detect significant differences among ML, MML, MMC, MMU, LCI and Random strategies for the significance level considered.

In the case of H_L measure, the CMN and BinMin strategies significantly outperformed Random strategy. The Shaffer's test did not detect significant differences among ML, MML, MMC, MMU, LCI and Random strategies for the significance level considered.

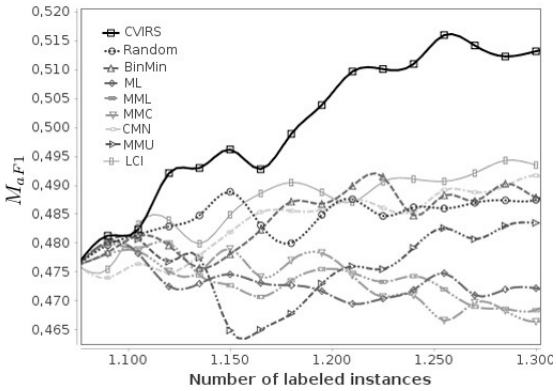
In the case of F_{1Ex} measure, the CMN and BinMin performed better than MML, MMC and Random strategies. Furthermore, LCI strategy outperformed Random and MMC strategies. Shaffer's test did not detect significant differences among ML, MML, MMC, MMU and Random strategies for the significance level considered.

4.3.2. Label Ranking task

Figures 6-9 represent the learning curves of the active learning strategies on the Emotions, Cal500, Genbase, Medical and Yeast datasets. Through visual inspection, Figure 6 shows that the CVIRS and CMN strategies obtained the best results for the A_P and R_L measures on the



(a) Performance for F_{1Ex} measure.



(b) Performance for M_{aF_1} measure.

Figure 4: Performance of the multi-label active learning strategies on the TMC2007-500 dataset.

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.541	0.691	0.668	0.671	0.671	0.683	0.688	0.681	**0.692**
Emotions	0.616	0.621	0.640	0.643	0.644	0.658	0.601	0.607	**0.659**
Birds	0.265	0.333	0.384	0.385	0.387	0.412	0.326	0.396	**0.415**
Genbase	0.945	0.949	0.952	0.946	0.923	0.956	0.921	0.940	**0.963**
Cal500	0.330	0.336	0.331	0.330	0.332	0.332	0.329	0.328	**0.346**
Medical	0.648	0.648	0.570	0.556	0.609	0.665	0.665	0.665	**0.667**
Yeast	0.575	0.630	0.618	0.608	0.616	0.640	0.780	**0.784**	0.658
Scene	0.630	0.634	0.618	0.608	0.616	0.640	0.642	0.630	**0.643**
Enron	0.420	0.436	0.372	0.378	0.384	0.457	0.447	0.450	**0.464**
Corel5k	0.101	**0.168**	0.126	0.128	0.120	0.158	0.154	0.157	0.160
Corel16k	0.099	**0.161**	0.145	0.146	0.149	0.152	0.155	0.154	0.158
TMC2007-500	0.598	0.608	0.589	0.584	0.584	0.608	0.597	0.600	**0.620**
Bibtex	0.203	0.299	0.274	0.286	0.289	0.312	0.298	0.314	**0.321**
Arts	0.200	**0.266**	0.260	0.262	0.259	0.265	0.249	0.260	0.264
Business	0.305	0.366	**0.476**	0.391	0.375	0.387	0.411	0.422	0.436
Entertainment	0.259	0.343	0.323	0.304	0.298	0.332	0.334	0.333	**0.350**
Recreation	0.199	0.268	0.265	0.264	0.258	0.268	0.261	0.255	**0.273**
Health	0.301	0.359	0.347	0.332	0.315	0.347	0.357	0.341	**0.371**
Ave. Rank	7.806	3.583	6.000	6.528	6.639	3.278	5.056	4.722	1.389
Pos.	9	3	6	7	8	2	5	4	1

Table 3: ALC results for the $M_{iF_1}(\uparrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 6.121E-11.

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.569	0.583	0.572	0.576	0.562	**0.592**	0.575	0.567	0.588
Emotions	0.517	0.520	0.608	0.636	0.636	0.642	0.495	0.498	**0.654**
Birds	0.304	0.255	0.309	0.310	0.311	**0.332**	0.239	0.311	0.330
Genbase	0.751	0.785	**0.806**	0.753	0.699	0.794	0.735	0.788	0.785
Cal500	0.161	0.156	0.154	0.154	0.151	0.162	0.156	0.146	**0.170**
Medical	0.352	0.348	0.310	0.312	0.317	0.376	0.370	0.369	**0.383**
Yeast	0.385	0.413	**0.416**	0.408	0.396	0.393	0.398	0.396	0.400
Scene	0.645	0.640	0.624	0.612	0.628	0.650	0.647	0.634	**0.651**
Enron	0.152	0.173	0.147	0.152	0.154	0.171	0.170	0.166	**0.185**
Corel5k	0.274	0.315	0.303	0.310	0.300	**0.321**	0.300	0.309	0.314
Corel16k	0.033	0.059	0.048	0.054	0.051	0.062	0.060	0.061	**0.065**
TMC2007-500	0.485	0.497	0.479	0.473	0.467	0.500	0.476	0.487	**0.521**
Bibtex	0.111	0.145	0.149	0.154	0.152	0.152	0.150	0.151	**0.156**
Arts	0.132	**0.171**	0.147	0.148	0.147	0.167	0.155	0.159	0.170
Business	0.135	0.158	0.159	0.161	0.158	0.158	0.148	0.149	**0.170**
Entertainment	0.154	0.200	0.191	0.195	0.187	0.197	0.190	0.194	**0.201**
Recreation	0.142	0.209	0.207	0.205	0.204	0.198	0.197	0.190	**0.218**
Health	0.123	0.188	0.171	0.169	0.155	0.174	0.188	0.185	**0.194**
Ave. Rank	7.528	3.972	5.778	5.306	6.667	2.972	5.750	5.389	1.639
Pos.	9	3	7	4	8	2	6	5	1

Table 4: ALC results for the $M_{aF_1}(\uparrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 1.029E-10.

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.365	0.301	0.313	0.311	0.313	0.304	0.304	0.310	**0.294**
Emotions	0.234	0.235	0.228	0.226	0.224	0.222	0.244	0.241	**0.221**
Birds	0.200	0.117	0.091	0.091	0.089	**0.078**	0.117	0.085	0.083
Genbase	0.006	0.005	0.004	0.005	0.007	0.004	0.008	0.006	**0.003**
Cal500	0.196	0.197	0.200	0.201	0.196	0.199	0.199	0.190	**0.185**
Medical	0.019	**0.018**	0.021	0.023	0.020	**0.018**	**0.018**	0.019	**0.018**
Yeast	0.251	0.247	0.266	0.272	0.281	0.248	0.250	0.248	**0.243**
Scene	0.145	**0.136**	0.142	0.141	0.144	0.140	0.142	0.139	0.137
Enron	0.086	0.082	0.095	0.095	0.091	**0.076**	0.085	0.080	0.078
Corel5k	0.045	**0.017**	0.023	0.022	0.020	**0.017**	0.019	0.019	**0.017**
Corel16k	0.052	0.036	0.039	0.040	0.042	0.036	0.037	0.042	**0.035**
TMC2007-500	0.084	**0.078**	0.084	0.085	0.085	0.079	0.087	0.082	**0.078**
Bibtex	0.029	0.017	0.021	0.020	0.023	0.017	0.021	0.019	**0.014**
Arts	0.295	0.180	0.140	**0.138**	0.139	0.198	0.213	0.158	0.205
Business	0.188	0.115	**0.084**	0.098	0.110	0.105	0.099	0.102	0.094
Entertainment	0.245	0.170	0.168	0.172	0.194	0.182	0.177	0.178	**0.165**
Recreation	0.289	0.239	**0.210**	0.225	0.234	0.231	0.221	0.233	0.213
Health	0.187	0.128	**0.112**	0.124	0.129	0.133	0.154	0.161	0.119
Ave. Rank	7.639	4.000	5.083	5.556	6.389	3.667	5.806	5.028	1.833
Pos.	9	3	5	6	8	2	7	4	1

Table 5: ALC results for the $H_L(\downarrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 5.829E-9.

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.601	**0.677**	0.644	0.647	0.648	0.663	0.662	0.657	0.674
Emotions	0.555	0.563	0.587	0.592	0.587	0.615	0.539	0.547	**0.616**
Birds	0.488	0.522	0.518	0.520	0.520	**0.605**	0.513	0.576	0.590
Genbase	0.955	0.958	0.958	0.953	0.930	0.958	0.941	0.950	**0.960**
Cal500	0.335	0.336	0.331	0.330	0.332	0.330	0.328	0.327	**0.345**
Medical	0.625	0.616	0.521	0.510	0.575	0.639	0.640	0.645	**0.650**
Yeast	0.553	0.565	0.558	0.547	0.533	0.554	0.565	0.566	**0.568**
Scene	0.599	0.594	0.573	0.554	0.578	0.610	0.614	0.597	**0.634**
Enron	0.424	0.432	0.377	0.382	0.388	0.455	0.440	0.450	**0.465**
Corel5k	0.099	**0.158**	0.124	0.123	0.113	0.146	0.130	0.147	**0.158**
Corel16k	0.086	**0.147**	0.135	0.139	0.137	0.139	0.140	0.141	0.144
TMC2007-500	0.600	0.589	0.574	0.570	0.567	0.592	0.605	0.607	**0.622**
Bibtex	0.203	0.271	0.268	0.271	0.269	0.290	0.274	0.283	**0.295**
Arts	0.198	0.272	0.255	0.256	0.253	0.273	0.254	0.272	**0.275**
Business	0.374	0.393	**0.540**	0.421	0.458	0.438	0.477	0.485	0.498
Entertainment	0.296	**0.366**	0.340	0.322	0.301	0.357	0.354	0.358	0.362
Recreation	0.221	0.281	0.274	0.270	0.268	0.281	0.284	0.286	**0.290**
Health	0.332	0.373	0.358	0.331	0.320	0.361	0.374	0.371	**0.386**
Ave. Rank	7.444	3.778	6.139	6.722	7.056	3.639	4.833	4.083	1.306
Pos.	9	3	6	7	8	2	5	4	1

Table 6: ALC results for the $F_{1Ex}(\uparrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 4.523E-11.

Emotions dataset. The MMU strategy obtained worse results than Random strategy.

Figure 7 shows that CVIRS strategy obtained the best results for the A_P and R_L measures on the Cal500 dataset. The CMN, ML, MML and MMU strategies showed a poor performance on this dataset - their learning curves were dominated by the learning curve of the Random strategy.

Figure 8 shows that the CVIRS and LCI strategies obtained the best results for the A_P and R_L measures on the Medical dataset. The ML, MML, BinMin and MMC strategies showed a poor performance on this dataset, their learning curves were dominated by the learning curve of the Random strategy.

Figure 9 shows that the CVIRS, BinMin, LCI and CMN strategies obtained the best results for the A_P and R_L measures on the Yeast dataset. The MML and MMC strategies showed a poor performance on this dataset.

Tables 7-9 show the ALC results obtained by the active learning strategies for the ranking-based measures (O_E, R_L and A_P). In all cases, the best results are highlighted in bold typeface in the tables. In the tables, the last two rows show the average rank (Ave. Rank) and the ranking position (Pos.) for each strategy according to Friedman's test.

For the ranking-based measures, CVIRS strategy generally showed a good performance on the 18 multi-label datasets. BinMin strategy obtained good results on the Flags, Corel16k and TMC2007-500 datasets. LCI strategy obtained good results on the Arts dataset. CMN strategy showed a good performance on Birds and Emotions datasets.

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.332	0.252	0.347	0.344	0.272	0.274	0.239	0.276	**0.238**
Emotions	0.324	0.322	0.305	0.305	0.299	0.290	0.321	0.322	**0.276**
Birds	0.901	0.853	0.785	0.784	0.782	0.776	0.845	0.801	**0.775**
Genbase	0.037	**0.022**	0.033	0.050	0.065	0.039	0.040	0.045	0.029
Cal500	0.800	0.847	0.845	0.848	0.824	0.840	0.808	0.780	**0.769**
Medical	0.303	0.320	0.416	0.432	0.356	0.299	0.299	0.294	**0.283**
Yeast	0.382	0.342	0.392	0.406	0.456	0.351	0.372	0.358	**0.326**
Scene	0.328	0.332	0.349	0.357	0.345	0.336	0.328	0.335	**0.321**
Enron	0.699	**0.618**	0.791	0.781	0.768	0.640	0.650	0.687	**0.618**
Corel5k	0.930	0.851	0.928	0.913	0.915	**0.849**	0.866	0.874	0.852
Corel16k	0.902	0.840	0.848	0.846	0.851	0.848	0.844	0.842	**0.838**
TMC2007-500	0.339	0.313	0.350	0.360	0.361	0.321	0.366	0.349	**0.308**
Bibtex	0.621	0.586	0.615	0.609	0.612	0.566	0.588	0.571	**0.549**
Arts	0.841	0.760	0.739	**0.736**	0.739	0.756	0.780	**0.736**	0.771
Business	0.799	0.726	**0.500**	0.654	0.700	0.679	0.659	0.675	0.641
Entertainment	0.842	0.738	0.754	0.775	0.788	0.750	0.723	0.735	**0.703**
Recreation	0.823	0.773	0.757	0.787	0.777	0.781	0.788	0.772	**0.747**
Health	0.789	0.764	0.697	0.674	0.642	0.773	0.642	0.640	**0.634**
Ave. Rank	7.083	4.444	5.972	6.389	6.167	4.333	4.861	4.167	1.583
Pos.	9	4	6	8	7	3	5	2	1

Table 7: ALC results for the $O_E(\downarrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 1.601E-8.

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.302	**0.261**	0.293	0.282	0.279	0.288	0.266	0.289	0.266
Emotions	0.201	0.206	0.194	0.192	0.190	**0.184**	0.211	0.207	**0.184**
Birds	0.198	0.158	0.132	0.132	0.134	0.128	0.156	0.133	**0.126**
Genbase	0.009	0.006	0.008	0.008	0.029	0.014	0.008	0.007	**0.005**
Cal500	0.249	0.250	0.254	0.255	0.251	0.256	0.253	0.248	**0.234**
Medical	0.089	0.118	0.171	0.183	0.113	0.089	0.087	0.085	**0.083**
Yeast	0.229	0.220	0.231	0.237	0.254	0.220	0.226	0.221	**0.216**
Scene	0.131	0.137	0.147	0.154	0.145	0.133	0.127	0.136	**0.123**
Enron	0.189	0.225	0.188	0.185	0.183	0.214	0.192	0.218	**0.174**
Corel5k	0.501	0.425	0.377	**0.372**	0.374	0.457	0.431	0.426	0.406
Corel16k	0.399	0.346	0.351	0.348	0.350	0.345	0.349	0.356	**0.341**
TMC2007-500	0.088	0.078	0.084	0.085	0.085	0.078	0.079	**0.077**	**0.077**
Bibtex	0.301	0.268	0.273	0.274	0.270	0.261	0.264	0.267	**0.248**
Arts	0.365	0.262	0.285	0.283	0.300	0.259	0.262	0.254	**0.250**
Business	0.299	0.280	**0.166**	0.190	0.296	0.284	0.254	0.231	0.226
Entertainment	0.350	0.240	0.224	**0.221**	0.249	0.261	0.257	0.264	0.225
Recreation	0.311	0.248	0.258	0.267	0.274	0.258	0.254	0.255	**0.240**
Health	0.287	0.221	0.241	0.239	0.230	0.238	0.233	0.236	**0.210**
Ave. Rank	7.417	4.528	5.667	5.444	5.806	4.917	4.889	4.806	1.528
Pos.	9	2	7	6	8	5	4	3	1

Table 8: ALC results for the $R_L(\downarrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 1.732E-7.

According to the average rankings returned by the Friedman's test, for the O_E measure the strategies that obtained the best results were CVIRS, LCI and CMN, in this order. In the case of R_L measure, the strategy that obtained the best results was CVIRS. In the case of A_P measure, the strategies that obtained the best results were CVIRS, BinMin and CMN.

Given that the p-values of Friedman's test were lower than the level of significance set as α=0.05, we concluded that there were significant differences among the observed ALC's values in the ranking-based measures considered. Afterwards, a Shaffer's post-hoc test for all pairwise comparisons was carried out. The statistical information obtained from the Shaffer's test was illustrated as a graph (see Figure 10).

From a statistical point of view, CVIRS was the strategy that obtained the best performance for the three ranking-label measures considered. CVIRS took the first position in the average rankings of methods returned by the Friedman's test.

In the case of O_E measure, CVIRS strategy significantly outperformed Random, ML, MML, MMC, MMU and BinMin strategies. Furthermore, LCI strategy performed better than Random strategy. The statistical test did not detect significant differences among Random, MMC, ML, MML, MMU, BinMin and CMN strategies for the significance level considered.

In the case of R_L measure, CVIRS strategy significantly outperformed all the strategies. Also, BinMin strategy outperformed to Random strategy. The Shaffer's test did not detect significant differences among Random, ML, MML, MMC, MMU, CMN and LCI

Dataset	Multi-label AL strategy								
	Random	**BinMin**	**ML**	**MML**	**MMC**	**CMN**	**MMU**	**LCI**	**CVIRS**
Flags	0.685	**0.792**	0.757	0.761	0.775	0.774	0.778	0.771	0.787
Emotions	0.759	0.768	0.778	0.778	0.782	0.789	0.758	0.765	**0.790**
Birds	0.399	0.418	0.507	0.507	0.507	0.514	0.424	0.492	**0.519**
Genbase	0.960	**0.980**	0.975	0.966	0.943	0.968	0.968	0.960	0.978
Cal500	0.345	0.340	0.332	0.331	0.340	0.332	0.338	0.349	**0.367**
Medical	0.757	0.725	0.642	0.628	0.704	0.754	0.755	0.764	**0.775**
Yeast	0.681	0.695	0.677	0.664	0.644	0.692	0.680	0.690	**0.699**
Scene	0.790	0.792	0.780	0.775	0.782	0.792	**0.807**	0.789	0.804
Enron	0.448	0.453	0.387	0.392	0.401	0.474	0.447	0.450	**0.479**
Corel5k	0.113	0.163	0.124	0.122	0.129	**0.152**	0.134	0.138	0.150
Corel16k	0.166	0.183	0.168	0.170	0.175	0.178	0.175	0.174	**0.184**
TMC2007-500	0.726	0.740	0.720	0.714	0.714	0.738	0.717	0.722	**0.744**
Bibtex	0.301	0.356	0.337	0.341	0.354	0.375	0.377	0.374	**0.387**
Arts	0.296	0.392	0.390	0.394	0.385	0.394	0.376	**0.408**	0.386
Business	0.444	0.436	**0.618**	0.592	0.302	0.462	0.498	0.488	0.525
Entertainment	0.364	0.439	0.431	0.395	0.374	0.424	0.428	0.439	**0.460**
Recreation	0.302	0.398	0.405	0.390	0.378	0.388	0.411	0.401	**0.414**
Health	0.356	0.412	**0.443**	0.421	0.400	0.400	0.402	0.410	0.425
Ave. Rank	7.250	3.806	5.667	6.417	6.611	4.139	4.833	4.556	1.722
Pos.	9	2	6	7	8	3	5	4	1

Table 9: ALC results for the $A_P(\uparrow)$ measure. Friedman's test rejected the null hypothesis with a p-value equal to 3.137E-9.

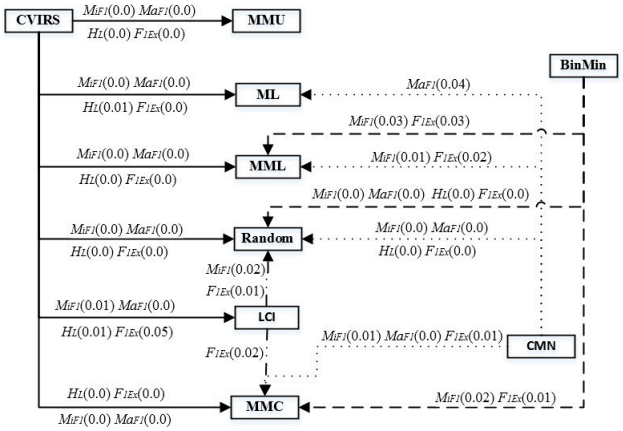


Figure 5: Significant differences in performance among active learning strategies according to the Shaffer's test.

strategies for the significance level considered.

In the case of A_P measure, CVIRS strategy significantly outperformed Random, MMU, LCI, ML, MML and MMC strategies. The BinMin and CMN strategies performed better than Random strategy. Significant differences among CVIRS, BinMin and CMN strategies were not detected for the significance level considered. Shaffer's test did not detect significant differences among Random, MMU, LCI, ML, MML and MMC strategies for the significance level considered.

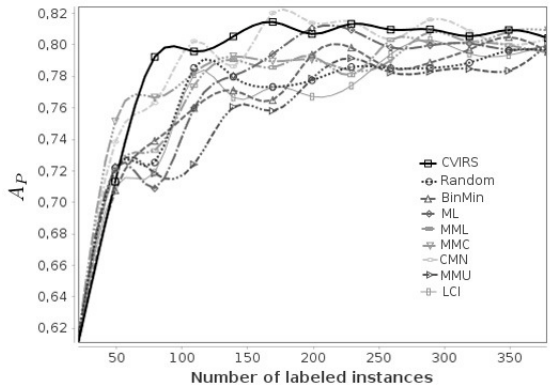
4.3.3. Discussion

The main goal of the experimental study was to compare several state-of-the-art multi-label active learning strategies and our proposal over a sizeable number of multi-label datasets and two multi-label learning tasks. The evidence suggested that our proposal (CVIRS) performed well for the two tasks analysed, the Multi-label Classification and Label Ranking tasks. Taking into account the average rankings returned by Friedman's test, the CVIRS, CMN and BinMin strategies obtained the best results in Multi-label Classification task. In Label Ranking task, the strategies that obtained the best results were CVIRS, BinMin, CMN and LCI.

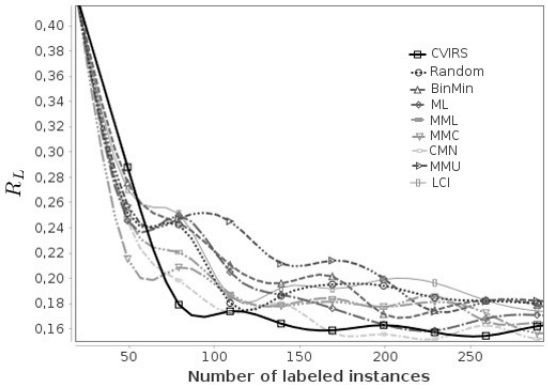
Although the Shaffer's test did not detect significant differences among the CVIRS, CMN and BinMin strategies in some of the evaluation measures considered, the CVIRS strategy was ranked before CMN and BinMin in all the average rankings returned by Friedman's test. This situation revealed the importance of considering the information for all labels.

Generally speaking, CVIRS strategy worked well on multi-label datasets with diverse characteristics and from different domains. The evidences suggested that CVIRS strategy obtained better results on multi-label datasets that have a small number of labels, e.g. the Emotions, Birds, Yeast and datasets from Yahoo Collection, than on datasets that have a large number of labels, e.g. the Cal500, Corel5k and Bibtex datasets. In datasets that have a large number of labels, the performance of CVIRS strategy can be affected due to the positional method used to resolve the rank aggregation problem formulated.

It is relevant to note that although the BinMin, ML and CMN strategies do not take into consideration the information for all labels, they worked well in multi-label datasets that have a small label cardinality and a small label density, e.g. the Corel5k and datasets from Yahoo Collection.

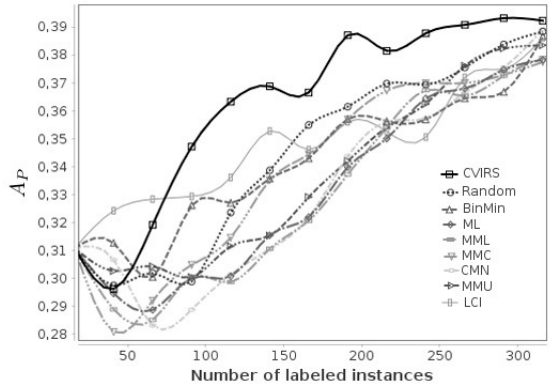


(a) Performance for A_P measure.

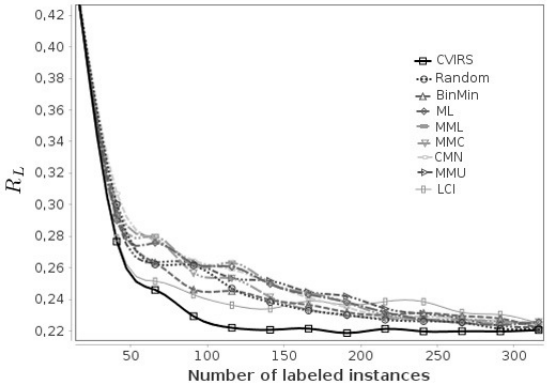


(b) Performance for R_L measure.

Figure 6: Performance of the multi-label active learning strategies on the Emotions dataset.

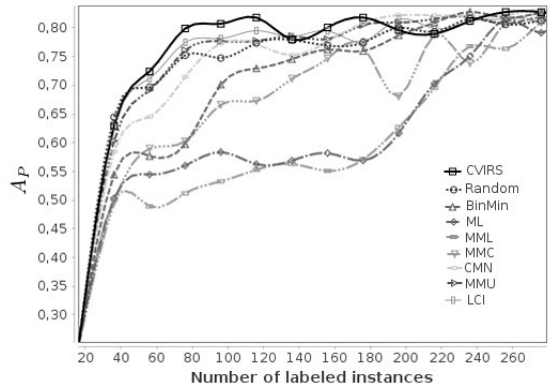


(a) Performance for A_P measure.

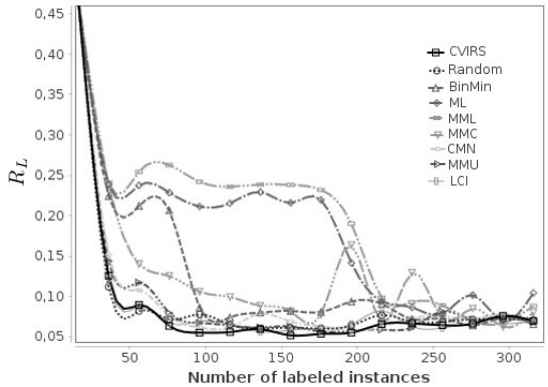


(b) Performance for R_L measure.

Figure 7: Performance of the multi-label active learning strategies on the Cal500 dataset.



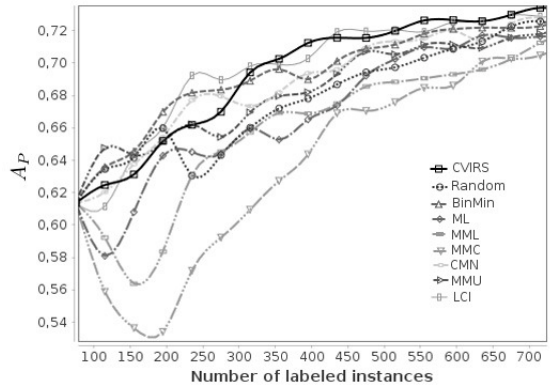
(a) Performance for A_P measure.



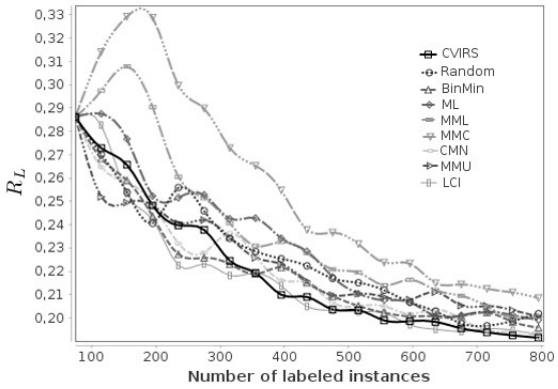
(b) Performance for R_L measure.

Figure 8: Performance of the multi-label active learning strategies on the Medical dataset.

It appears that CVIRS strategy performs well using BR-SVM as its base classifier and the experimental settings set. To ensure fairness, a similar experimental protocol to those used in related works was employed.



(a) Performance for A_P measure.



(b) Performance for R_L measure.

Figure 9: Performance of the multi-label active learning strategies on the Yeast dataset.

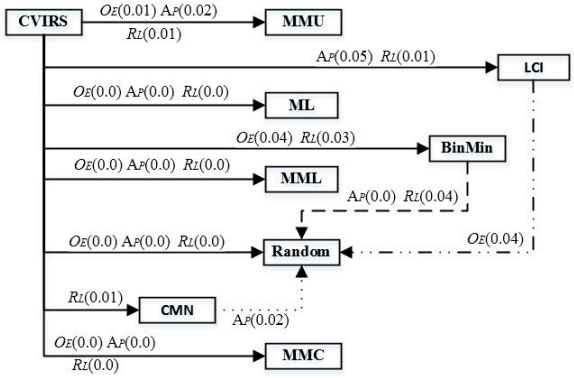


Figure 10: Significant differences in performance among active learning strategies according to the Shaffer's test for the ranking-based measures.

However, in future researches it would be important to test the state-of-the-art active learning strategies and our proposal with other multi-label algorithms as base classifiers and other experimental settings. It is also important to study how these strategies perform with base classifiers that do not follow the BR approach, e.g. multi-label classifiers that belong to algorithm adaptation category.

Regarding the computational cost to determine the utility of an unlabelled instance, let $f_{ts}(\Phi)$ be the cost function for multi-label classifier Φ to classify an instance. $f_{ts}(LRBP)$ denotes the cost function for the "LR-based prediction method" used by the MMC strategy [36] to classify an instance. L_s is the labelled set, U_s is the unlabelled set, and q is the number of labels.

The BinMin, ML, CMN and MMU strategies require $O(f_{ts}(\Phi))$ steps to determine the utility of an unlabelled instance. BinMin computes the minimum distance among the restricting hyperplane and the center of the maximum radius hyperball over the outputs of each binary SVM classifier. ML computes the loss value only on the label predicted with the most certainty. CMN calculates the minimum confidence value over the vector of confidences returned by classifier Φ. MMU computes the maximum margin prediction taking the set of positive and negative labels predicted by classifier Φ. The BinMin, ML, CMN and MMU strategies have the lowest computational cost.

MML strategy requires $O(max(q^2, f_{ts}(\Phi)))$ steps, as MML computes the loss produced in each label. The MMC strategy requires $O(max(f_{ts}(LRBP), f_{ts}(\Phi)))$ steps, as MMC needs the prediction of the classifier Φ and the prediction of the "LR-based prediction method". The LCI strategy requires $O(max(q \cdot |L_s|, f_{ts}(\Phi)))$ steps, as LCI needs the prediction of the classifier Φ and it also computes the label cardinality inconsistency over the current labelled set.

CVIRS strategy requires $O(max(|U_s| \cdot f_{ts}(\Phi), q \cdot |U_s|^2))$ steps, as CVIRS first needs to compute the difference margin of prediction on each label for all unlabelled instances, and then computes q rankings of unlabelled instances in order to calculate the final score. The cost of computing q rankings of unlabelled instances can be reduced if only a subset of the unlabelled set is considered and an efficient order algorithm is used.

5. Conclusions

In this paper, an active learning strategy for working on multi-label data was proposed, known as CVIRS. CVIRS selects the most uncertain instances based on two defined measures. The first measure computes the unified uncertainty of an unlabelled instance based on

difference margins on the predictions of base classifier. In order to aggregate the multi-label information, a rank aggregation problem was defined and a simple rank aggregation method for computing the score of an unlabelled instance was used. The second measure computes the inconsistency of a predicted label set with respect to the label set of labelled instances. CVIRS can be used with base classifiers which can obtain proper probability estimates from their outputs. CVIRS is not restricted to base classifiers that use problem transformation methods, it can also be used with multi-label learning algorithms that belong to algorithm adaptation category.

Several active learning strategies were tested over 18 multi-label datasets that belong to different domains. The empirical study showed that CVIRS strategy performed well on multi-label datasets with diverse characteristics. Also, CVIRS strategy obtained good results for the two tasks analysed; the Multi-label Classification and Label Ranking tasks. The experimental study showed that CVIRS strategy is competitive with respect to the state-of-the-art multi-label active learning strategies. The evidence suggested that the use of rank aggregation methods can be beneficial in developing multi-label active learning strategies.

In the experimental study, visual inspection of the learning curves and the area under the learning curves have been used as processes to provide a qualitative and quantitative intuition, respectively, to decide which strategy performs better. In cases where the learning curves of several strategies overlapped, the area under the learning curve can be used to decide which active learning strategy performs better with a statistical support.

Future research will study more effective approaches to resolve the rank aggregation problem formulated into the CVIRS strategy. In addition, we will study active learning strategies for working on batch-mode multi-label active learning, an area in which, in contrast to batch-mode single-label active learning, far less research has been carried out. It would also be useful to analyse the effectiveness of the approach using incremental learning algorithms to speed up the updating of the base classifiers in each iteration.

Acknowledgements

This research was supported by the Spanish Ministry of Economy and Competitiveness, project TIN-2014-55252-P, and by FEDER funds.

References

[1] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: Working Notes of the AAAI'99 Workshop on Text Learning, 1999.

[2] A. Srivastava, B. Zane-Ulman, Discovering recurring anomalies in text reports regarding complex space systems, in: Proceedings of the Aerospace Conference, IEEE, 2005, pp. 55–63.

[3] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Proceedings of the ECML/PKDD Discovery Challenge, 2008.

[4] T. Li, M. Ogihara, Detecting emotion in music, in: Proceedings of the International Symposium on Music Information Retrieval, Washington DC, United States of America, 2003, pp. 239–240.

[5] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, M. I. Jordan, Matching words and pictures, J. Mach. Learn. Res. 3 (2003) 1107–1135.

[6] S. Yang, S. Kim, Y. Ro, Semantic home photo categorization, IEEE Trans. Circuits Syst. Video Technol. 17 (2007) 324–335.

[7] E. Correa, A. Plastino, A. Freitas, A Genetic Algorithm for Optimizing the Label Ordering in Multi-Label Classifier Chains, in: Proceedings of the 25th International Conference on Tools with Artificial Intelligence (ICTAI'13), IEEE, 2013.

[8] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.

[9] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Trans. Audio Speech Lang. Process. 16 (2) (2008) 467–476.

[10] J.Wang, Y. Zhao, X. Wu, X. Hua, A transductive multi-label learning approach for video concept detection, Pattern Recognit. 44 (2010) 2274–2286.

[11] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Vol. 14, MIT Press, 2001, pp. 681–687.

[12] S. Diplarisa, G. Tsoumakas, P. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: Proceedings of the 10th Panhellenic Conference on Informatics (PCI'05), 2005, pp. 448–456.

[13] M. L. Zhang, Z. H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (2006) 1338–1351.

[14] N. Cesa-Bianchi, G. Valentini, Hierarchical cost-sensitive algorithms for genome-wide gene function prediction, J. Mach. Learn. Res. 8 (2010) 14–29.

[15] F. Otero, A. Freitas, C. Johnson, A hierarchical multi-label classification ant colony algorithm for protein function prediction, Memetic Comput. 2 (2010) 165–181.

[16] M. G. Larese, P. Granitto, J. Gómez, Spot defects detection in cDNA microarray images, Pattern Anal. Applic. 16 (2013) 307–319.

[17] F. Briggs, et. al., The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment, in: Proceedings of the International Workshop on Machine Learning for Signal Processing (MLSP'13), IEEE, 2013.

[18] E. Ukwatta, J. Samarabandu, Vision based metal spectral analysis using multi-label classification, in: Canadian Conference on Computer and Robot Vision (CRV'09), 2009, p. 132139.

[19] G. Tsoumakas, I. Katakis, I. Vlahavas, Data Mining and Knowledge Discovery Handbook, 2nd Edition, Springer-Verlag, New York, United States of America, 2010, Ch. Mining Multi-label Data, pp. 667–686.

[20] G. Madjarov, D. Kocev, D. Gjorgjevikj, An extensive experi-

mental comparison of methods for multi-label learning, Pattern Recognit. 45 (2012) 3084–3104.
[21] E. Gibaja, S. Ventura, Multi-label learning: a review of the state of the art and ongoing research, WIREs Data Mining Knowl. Discov. 4 (2014) 411–444.
[22] K. Brinker, J. Furnkranz, E. Hullermeier, A unified model for multilabel classification and ranking, in: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06), 2006, pp. 489–493.
[23] X. Zhu, A. B. Goldberg, Introduction to Semi-Supervised Learning, Morgan & Claypool Publishers, 2009.
[24] B. Settles, Active Learning, 1st Edition, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012.
[25] D. Lewis, W. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Springer-Verlag, 1994, pp. 3–12.
[26] J. Fu, S. Lee, Certainty-based active learning for sampling imbalanced datasets, Neurocomputing 119 (2013) 350–358.
[27] W. Wu, Y. Liu, M. Guo, C. Wang, X. Liu, A probabilistic model of active learning with multiple noisy oracles, Neurocomputing 118.
[28] S. Jones, L. Shao, K. Du, Active learning for human action retrieval using query pool selection, Neurocomputing 124 (2014) 89–96.
[29] X. Zhang, S. Wang, X. Zhu, X. Yun, G. Wu, Y. Wang, Update vs. upgrade: Modeling with indeterminate multi-class active learning, Neurocomputing 162 (2015) 163–170.
[30] J. Zhou, S. Sun, Gaussian process versus margin sampling active learning, Neurocomputing 167 (2015) 122–131.
[31] H. Yu, C. Sun, W. Yang, X. Yang, X. Zuo, AL-ELM: One uncertainty-based active learning algorithm using extreme learning machine, Neurocomputing 166 (2015) 140–150.
[32] X. Li, Y. Guo, Active Learning with Multi-Label SVM Classification, in: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, AAAI Press, 2013, pp. 1479–1485.
[33] X. Li, L. Wang, E. Sung, Multi-label SVM active learning for image classification, in: Proceedings of the International Conference on Image processing (ICIP'04), Vol. 4, IEEE, 2004, pp. 2207–2210.
[34] K. Brinker, From Data and Information Analysis to Knowledge Engineering, Springer, 2006, Ch. On Active Learning in Multi-label Classification, pp. 206–213.
[35] G. Qi, X. Hua, Y. Rui, J. Tang, H. Zhang, Two-dimensional active learning for image classification, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–25.
[36] B. Yang, J. Sun, T. Wang, Z. Chen, Effective multi-label active learning for text classification, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, Paris, France, 2009, pp. 917–926.
[37] G. Qi, X. Hua, Y. Rui, J. Tang, H. Zhang, Two-dimensional multi-label active learning with an efficient online adaptation model for image classification, IEEE Trans. Pattern Anal. Machine Intell. 99 (1).
[38] X. Zhang, J. Cheng, C. Xu, H. Lu, S. Ma, Multi-view multi-label active learning for image classification, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME-2009), IEEE, 2009, pp. 258–261.
[39] A. Esuli, F. Sebastiani, Active Learning Strategies for Multi-Label Text Classification, in: Advances in Information Retrieval, Springer, 2009, pp. 102–113.
[40] S. Huang, Z. Zhou, Active query driven by uncertainty and diversity for incremental multi-label learning, in: Proceedings of the 13th International Conference on Data Mining, IEEE, 2013, pp. 1079–1084.
[41] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[42] S. García, F. Herrera, An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets'' for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[43] H. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proceedings of the ACM Workshop on Computational Learning Theory, ACM, 1992, pp. 287–294.
[44] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank Aggregation Methods for the Web, in: Proceedings of the 10th World Wide Web Conference, ACM, 2001, pp. 613–622.
[45] N. Ailon, M. Charikar, A. Newman., Aggregating inconsistent information: ranking and clustering, in: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC), ACM, Baltimore, Maryland, United States of America, 2005, pp. 684–693.
[46] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, E. Vee, Comparing and aggregating rankings with ties, in: Proceedings of the ACM Symposium on Principles of Database Systems (PODS), ACM, 2004, pp. 47–58.
[47] S. Guiasu, C. Reischer, Some remarks on entropic distance, entropic measure of connexion and hamming distance, RAIRO-Theoretical Informatics 13 (4) (1979) 395–407.
[48] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[49] J. Shaffer, Modified sequentially rejective multiple test procedures, J. Amer. Stat. Assoc. 81 (395) (1986) 826–831.
[50] G. Tsoumakas, E. Spyromitros-Xioufi, J. Vilcek, I. Vlahavas, MULAN: A Java Library for Multi-Label Learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.
[51] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.
[52] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas, Multilabel classification of music into emotions, in: Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08), 2008, pp. 325–330.
[53] P. Duygulu, K. Barnard, N. de Freitas, D. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: Proceedings of the 7th European Conference on Computer Vision, 2002, pp. 97–112.
[54] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, W. Duch, A shared task involving multi-label classification of clinical free text, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing (BioNLP-07). Association for Computational Linguistics, Stroudsburg, PA, United States of America, 2007, pp. 97–104.
[55] B. Klimt, Y. Yang, The Enron corpus: a new dataset for email classification research, in: Proceedings of the 15th European conference on Machine Learning (ECML-2004), Springer, 2004, pp. 217–226.
[56] G. Tsoumakas, I. Vlahavas, Random k-labelsets: An ensemble method for multilabel classification, in: Proceedings of the 18th European Conference on Machine Learning (ECML'07), Warsaw, Poland, 2007, pp. 406–417.
[57] N. Ueda, K. Saito, Parametric mixture models for multi-labeled text, in: Proceedings on Neural Information Processing Systems (NIPS'15), MIT Press, 2002, pp. 737–744.
[58] S. P. Wright, Adjusted p-values for simultaneous inference, Biometrics 48 (1992) 1005–1013.

TITLE:

Evolutionary Strategy to perform Batch-Mode Active Learning on Multi-Label Data

AUTHORS:

O. Reyes and S. Ventura

ACM Transactions on Intelligent Systems and Technology, *submitted, 2016*

RANKING:

Impact factor (JCR 2015): 2.414

Knowledge area:

Computer Science, Artificial Intelligence: 30/130

Computer Science, Information Systems: 20/143

A

Evolutionary Strategy to perform Batch-Mode Active Learning on Multi-Label Data

OSCAR REYES, University of Córdoba, Spain
SEBASTIÁN VENTURA, University of Córdoba, Spain, and King Abdulaziz University, Jeddah, Saudi Arabia

Multi-label learning has become an important area of research, owing to the increasing number of real-world problems that contain multi-label data. Data labeling is a very expensive process that requires expert handling. Consequently, numerous modern problems involve a small number of labeled examples and a large number of unlabeled examples simultaneously. Batch-mode active learning focusses on constructing accurate classifiers by means of choosing which instances will be labeled, in such a way that the selected instances are informative and the overlapping of information between them is minimal, reducing the labeling effort and the cost of training an accurate model. This paper presents a new strategy, named ESBMAL, to perform batch-mode active learning on multi-label data. ESBMAL formulates batch-mode active learning as a multi-objective problem and solves it by means of an evolutionary algorithm. Extensive experiments were conducted to validate the effectiveness of the proposal. The experimental results were validated using non-parametric statistical tests and confirmed the effectiveness of the proposal for better batch-mode multi-label active learning.

CCS Concepts: •**Theory of computation** → **Active learning;**

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Batch-mode multi-label active learning, multi-objective problem, genetic algorithm, multi-label classification, label ranking

ACM Reference Format:
Oscar Reyes and Sebastián Ventura. Evolutionary Strategy to perform Batch-Mode Active Learning on Multi-Label Data. *ACM Trans. Intell. Syst. Technol.* V, N, Article A (January YYYY), 20 pages.
DOI: -

1. INTRODUCTION

Multi-label problems concern problems where examples belong to multiple labels at the same time. The goal of the Multi-Label Learning paradigm is to develop a model that correctly generalizes unseen multi-label data. Multi-Label Classification and Label Ranking are two tasks that have been studied in the context of multi-label learning. The Multi-Label Classification (MLC) task aims to find a model where, for a given test instance, the labels are divided into relevant and irrelevant label sets. On the other hand, the Label Ranking (LR) task provides, for a given test instance, a permutation of the labels; the labels are ordered according to their relevance values. [Gibaja and Ventura 2014; Tsoumakas et al. 2010]

Particular real-world problems that involve multi-label data include text categorization [Katakis et al. 2008; Pestian et al. 2007], classification of emotions evoked by music [Li and Ogihara 2003], semantic annotation of images [Barnard et al. 2003], classi-

Author's addresses: O. Reyes, Department of Computer Science and Numerical Analysis, University of Córdoba, Spain; S. Ventura, Department of Computer Science and Numerical Analysis, University of Córdoba, Spain, and Department of Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia.

DOI: -

fication of music and videos [Boutell et al. 2004; Turnbull et al. 2008], classification of protein and gene function [Elisseeff and Weston 2001; Diplarisa et al. 2005], and many more. Generally speaking, the majority of the aforementioned problems originate from domains from which a huge amount of data is commonly available. Data labeling is a very expensive process that requires expert handling. In multi-label settings, experts must label each instance several times, as each instance belongs to various categories. The situation is further complicated when a multi-label dataset with a large number of examples and label classes (large-scale dataset) is analyzed. Consequently, several real scenarios nowadays contain a small number of labeled data and a large number of unlabeled data simultaneously.

Most multi-label learning algorithms that have been proposed in the literature are designed for working on supervised learning environments, i.e. scenarios where all training examples are labeled. However, the current techniques that allow us to create models from labeled and unlabeled data have attracted the attention of researchers. Active Learning (AL) focuses on constructing better classifiers by iteratively choosing which instances will be labeled for training [Settles 2012; Zhang et al. 2014b]. Consequently, the labeling effort and the cost of training an accurate model are reduced. In this work, we focus on AL scenarios in which a large collection of unlabeled data and a small set of labeled data are available, known as pool-based AL [Lewis and Gale 1994].

For more than a decade, a large number of AL methods for single-label data have been proposed (for an interesting survey, see [Settles 2012]). However, compared to single-label AL, the AL problem within a multi-label context is far less studied. The main challenge in performing AL on multi-label data is designing effective strategies that measure the unified informative potential of unlabeled instances across all labels.

Most state-of-the-art multi-label AL strategies employ the Binary Relevance (BR) approach [Tsoumakas et al. 2010] to break down a multi-label problem into several binary classification problems. Multi-label AL strategies are generally tested on the MLC task. However, their performance with regard to the LR task has not been considered. Most multi-label AL strategies only use informativeness-based criteria (e.g uncertainty measures) to select the most useful unlabeled instances, leading to a sub-optimal performance [Huang et al. 2014]. Other types of selection criteria, such as representativeness and diversity, have rarely been considered in the multi-label AL. Few works have combined two selection criteria to select the best unlabeled instances [Chakraborty et al. 2011; Li and Guo 2013; Huang and Zhou 2013; Zhang et al. 2014a; Huang et al. 2014], notably informativeness and diversity.

In several domains, such as in the speeding up of the process of inducting classifiers with slow training procedures or in systems where a parallel annotation environment is available, the selection of batches of unlabeled instances is preferred. Batch-mode AL methods select a batch of k unlabeled instances in each iteration, in such a way that the selected instances are informative and the overlapping of information between them is minimal [Fu et al. 2013]. Most state-of-the-art multi-label AL strategies were designed to select one unlabeled instance at a time, consequently they may have a sub-optimal performance in the resolution of the batch-mode AL problem. The most significant works related to performing batch-mode AL on multi-label data appeared in [Chakraborty et al. 2011; Zhang et al. 2014a]. However, the use of these methods is difficult, practically speaking, for application to large-scale multi-label datasets.

In this work, an efficient batch-mode AL strategy, named Evolutionary Strategy for Batch-Mode Multi-Label Active Learning (ESBMAL), is proposed. ESBMAL formulates batch-mode multi-label AL as a multi-objective problem, and resolves it by means of a genetic algorithm. ESBMAL aims to optimize three criteria (informativeness, representativeness and diversity) to select batches of unlabeled instances. ESBMAL can be used with any base multi-label classifier which can obtain proper probability es-

timates from its outputs. Several experiments were conducted to measure the effectiveness of our proposal. The experimental stage showed the effectiveness of our proposal, obtaining significantly better results than several state-of-the-art multi-label AL strategies on MLC and LR tasks.

To the best of our knowledge, for the first time, a multi-label AL strategy that combines informativeness, representativeness and diversity criteria is proposed. Also, it is the first attempt to formulate the batch-mode multi-label AL as a multi-objective problem, and resolve this type of AL problem by means of evolutionary algorithms.

This paper is arranged as follows: Section 2 briefly describes the state-of-the-art on multi-label AL. Section 3 presents our proposal. Section 4 describes the experimental study and analyses the experimental results. Finally, Section 5 provides some concluding remarks.

2. RELATED WORK

Several criteria have been proposed for selecting the most useful unlabeled instances to be annotated for an oracle. Selection criteria can be categorized into the following three groups: informativeness-based, representativeness-based and diversity-based.

Most AL strategies use informativeness-based criteria to select the most useful unlabeled instances. Informativeness measures the effectiveness of an instance by reducing the uncertainty of a model. AL strategies that only select informative instances usually do not exploit the structure of unlabeled data, leading to a sub-optimal performance [Huang et al. 2014]. In [Li et al. 2004; Brinker 2006; Qi et al. 2009; Zhang et al. 2009; Yang et al. 2009; Esuli and Sebastiani 2009; Singh et al. 2009; Hung and Lin 2011; Wang et al. 2012; Tang et al. 2012; Li and Guo 2013; Wu et al. 2014; Huang et al. 2015] several multi-label AL strategies that only consider informativeness-based selection criteria were proposed.

Representativeness-based selection criteria measure whether an unlabeled instance is representative of the underlying distribution [Huang et al. 2014], i.e. lies in a dense region of the input space. The studies of representativeness-based criteria applied to multi-label data are still scarce. In [Huang et al. 2014], to the best of our knowledge, appears the first attempt to desig a selection criterion that measures the representativeness of the unlabeled instances on multi-label data. Representativeness of unlabeled instances is a good criterion for improving the generalization of the learning model. However, the AL strategies that only consider representative instances may have to query a relatively large number of instances before an optimal decision boundary is reached [Huang et al. 2014].

Diversity-based selection criteria measure the diversity among a set of instances in order to reduce information overlapping. In [Li and Guo 2013], a selection criterion that measures the diversity between the instances from the label space perspective is proposed. On the other hand, diversity-based criteria have been widely used in the batch-mode AL scheme [Chakraborty et al. 2011; Zhang et al. 2014a].

In the multi-label context, most AL strategies use informativeness-based criteria, leaving the other types of selection criteria rarely considered. Few works have combined two of the three selection criteria above-mentioned. In [Huang et al. 2014], there is mention of a systematic way for measuring and combining informativeness and representativeness criteria. In [Li and Guo 2013; Huang and Zhou 2013], the authors combine informativeness and diversity criteria; the diversity criterion is analyzed from the label space perspective. On the other hand, in [Chakraborty et al. 2011; Zhang et al. 2014a], batch-mode multi-label AL strategies that use informativeness and diversity criteria to select batches of unlabeled instances are proposed.

Multi-label AL strategies can be also classified so that the labels of unlabeled instances are queried. Most AL strategies are designed to query all the label assign-

ments of the selected unlabeled instances [Li et al. 2004; Brinker 2006; Yang et al. 2009; Esuli and Sebastiani 2009; Singh et al. 2009; Chakraborty et al. 2011; Hung and Lin 2011; Wang et al. 2012; Tang et al. 2012; Li and Guo 2013; Wu et al. 2014]. On the other hand, several AL strategies query the relevance of an instance-label pair [Qi et al. 2009; Zhang et al. 2009; Huang and Zhou 2013; Zhang et al. 2014a; Huang et al. 2014; Huang et al. 2015; C.Ye et al. 2015], i.e. the strategy analyzes whether a specific label is relevant to the selected instance. AL strategies that query all labels of an instance may lead to information redundancy and more annotation effort in real problems with a large number of labels. On the other hand, AL strategies that select instance-label pairs avoid information redundancy, but they may ignore the interaction between labels and obtain limited supervision from each query [Huang et al. 2015].

Most state-of-the-art multi-label AL strategies were not designed to focus on batch-mode scenarios. However, these AL strategies can easily select a batch of unlabeled instances, e.g. by selecting the k most informative examples from the unlabeled set, k being the batch size. When a batch of unlabeled instances is selected in a greedy manner, the selected unlabeled instances may be similar, resulting in information overlapping. In the following cases, we refer to this type of strategies as myopic AL strategies. More information about myopic multi-label AL strategies that have been proposed in the literature can be found on the electronic appendix.

To date, few works focusing on batch-mode multi-label AL have been noted [Chakraborty et al. 2011; Zhang et al. 2014a]. In [Chakraborty et al. 2011], a batch-mode multi-label AL strategy is proposed, we call this strategy BMAL. To compute the informativeness of an unlabeled instance i, the current learning model is applied to the instance i to yield posterior probabilities over all labels. The uncertainty of the instance i is computed as the average entropy over individual labels. Regarding diversity-based criteria, a matrix that contains the redundancy values between each pair of unlabeled instances is computed. Symmetric Kullback Leibler divergence is used to compute the level of information overlap between two points. The authors formulated the batch mode AL problem as an integer programming problem that includes the vector of uncertainty values and the redundancy matrix. The optimization problem is subject to two constraints, and the authors relaxed the constraints to transform the integer programming problem into a continuous optimization problem. The authors proposed to resolve the continuous optimization problem using a solver for quadratic problems, such as the Quasi-Newton method. Once the optimization problem is resolved, a vector that shows whether an unlabeled instance is selected or not is obtained. BMAL queries all the label assignments of the selected unlabeled instances. The authors proved the effectiveness of the strategy on two image datasets.

The BMAL method is difficult to apply to large-scale datasets, since the computation of the decision vector on problems with a large number of unlabeled instances is very costly. The length of the decision vector is equal to the cardinality of the unlabeled set. On the other hand, BMAL requires the computation of the redundancy matrix in each iteration, which can be computationally expensive when a large pool of unlabeled instances is available.

In [Zhang et al. 2014a], a batch-mode multi-label AL strategy, named High-order label correlation driven Active Learning (HoAL), is proposed. HoAL uses selection criteria based on examples and label dimensions, i.e. the annotation selection granularity is based on instance-label pairs instead of only instances. HoAL defines a score function to measure the informative potential of instance-label pairs; it is designed based on both likelihood maximization (on labeled data) and uncertainty minimization (on unlabeled data). The authors used an efficient association rule mining algorithm to discover informative high-order label correlations. The authors defined a cross-label uncertainty which measures the disagreement between the mined label correlation and

the label co-occurrence possibility from the detected classification model. The batch-mode multi-label AL problem is formulated as an NP-hard integer programming problem, subject to three constraints. The problem formulation includes a very large auxiliary vector - its length is equal to twice the number of labels, multiplied by the number of unlabeled instances. In order to solve it in practice, the constraints are relaxed to transform the problem into a continuous optimization problem. The authors proposed to use the Quasi-Newton method to resolve the continuous optimization problem. The HoAL method was tested on four multi-label datasets for image and text classification.

The HoAL method is difficult to apply to large-scale datasets, since the computation of the auxiliary vector used to determine the batch of unlabeled instances is computationally expensive. On the other hand, the defined objective function is costly to evaluate, since the current classifier is updated in each iteration of the Quasi-Newton method.

Table 1 shows a summary of the state-of-the-art multi-label AL strategies. A check symbol (✓) indicates that the AL strategy in the rows implements the feature presented in the columns. Generally speaking, many state-of-the-art multi-label AL strategies employ the BR approach [Tsoumakas et al. 2010] to break down a multi-label problem into several binary classification problems. Bibliographic revision reveals that most multi-label AL strategies have been tested with BR-SVM as a base classifier, i.e. the BR method using binary Support Vector Machine classifiers [Li et al. 2004; Brinker 2006; Yang et al. 2009; Li and Guo 2013]. Previous works have often been tested on the MLC task. Most AL strategies use informativeness-based criteria to select the most useful unlabeled instances. Some strategies simply extend the binary uncertainty concept for multi-label data by aggregating the value associated with each label, e.g. taking the minimum [Brinker 2006] or the average value over all labels [Li et al. 2004; Yang et al. 2009; Singh et al. 2009; Wu et al. 2014]. Most multi-label AL strategies have been designed to select one unlabeled instance at a time, whereas the batch-mode multi-label AL problem has been much less studied [Chakraborty et al. 2011; Zhang et al. 2014a]. The few existing works based on batch-mode multi-label AL formulate the problem of selecting the best batch of unlabeled instances as an integer programming problem, which is usually difficult to apply to large-scale datasets. To date, to the best of our knowledge, a multi-label AL strategy that combines the three criteria (informativeness, representativeness and diversity) has not been proposed.

Table 1. Summary of state-of-the-art multi-label AL strategies. The AL strategies are ordered by year of publication. **I**: informativeness; **R**: representativeness; **D**: diversity; **All**: all labels assignment; **IL**: instance-label pairs assignment.

Multi-label AL strategy	Source	Type	Selection criterion			Labels assignment	
			I	R	D	All	IL
ML	[Li et al. 2004]	Myopic	✓			✓	
MML	[Li et al. 2004]	Myopic	✓			✓	
BinMin	[Brinker 2006]	Myopic	✓			✓	
2DAL	[Qi et al. 2009]	Myopic	✓				✓
Multi-view-2DAL	[Zhang et al. 2009]	Myopic	✓				✓
MMC	[Yang et al. 2009]	Myopic	✓			✓	
CMN	[Esuli and Sebastiani 2009]	Myopic	✓			✓	
MAAL	[Singh et al. 2009]	Myopic	✓			✓	
BMAL	[Chakraborty et al. 2011]	Batch-mode	✓		✓	✓	
HLR	[Hung and Lin 2011]	Myopic	✓			✓	
SHLR	[Hung and Lin 2011]	Myopic	✓			✓	
MCU	[Wang et al. 2012]	Myopic	✓			✓	
SGAL	[Tang et al. 2012]	Myopic	✓			✓	
MMU	[Li and Guo 2013]	Myopic	✓			✓	
LCI	[Li and Guo 2013]	Myopic			✓	✓	
Adaptative	[Li and Guo 2013]	Myopic	✓		✓	✓	
AUDI	[Huang and Zhou 2013]	Myopic	✓		✓		✓
HoAL	[Zhang et al. 2014a]	Batch-mode	✓		✓		✓
EMAL	[Wu et al. 2014]	Myopic	✓			✓	
LMAL	[Wu et al. 2014]	Myopic	✓				✓
QUIRE	[Huang et al. 2014]	Myopic	✓	✓			✓
MIML	[Vasisht and Damianou 2014]	Myopic	✓		✓	✓	✓
AURO	[Huang et al. 2015]	Myopic	✓				✓
CosMAL	[C.Ye et al. 2015]	Myopic	✓				✓

3. BATCH-MODE MULTI-LABEL ACTIVE LEARNING

Before introducing our approach, we show by means of an example, the importance of selecting a batch of unlabeled instances taking into account informativeness, representativeness and diversity criteria. In simpler terms, Figure 1 illustrates this situation for a binary linear classifier, where squares represent unlabeled instances. The dark squares represent the most uncertain instances for the current classifier - they are near the classification boundary. Suppose that, we want to select a batch of four unlabeled instances. If we use an AL strategy that only uses an informativeness-based criterion, the resulting batch would be composed by the instances 1, 2, 3 and 6, since they are the nearest to the classification boundary. However, the overlapping of information between instances 2 and 3 is not considered.

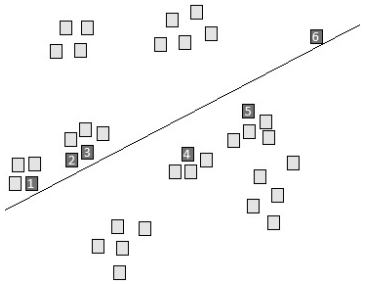


Fig. 1. Pool of unlabeled instances.

Suppose that we use an AL strategy that combines informativeness and diversity criteria. If the diversity-based criterion guarantees the diversity among the selected instances, then a possible batch would be composed by instances 1, 3, 5 and 6; in this case, instance 2 is not considered since it is similar to instance 3. The informativeness and diversity criteria do not guarantee that the selected instances are representative of the underlying distribution. In our example, instance 6 is not representative of other instances in the distribution, it could be an outlier, so knowing its label is unlikely to improve the accuracy of the model. Finally, if we use an AL strategy that combines the informativeness, diversity and representativeness criteria, then instance 6 is not considered, and a potential batch could be composed of instances 1, 3, 4 and 5.

To the best of our knowledge, the existing batch-mode multi-label AL strategies only consider informativeness and diversity criteria. We considered that a good balance between the three above-mentioned selection criteria would lead to superior performance in the resolution of the batch-mode AL problem.

3.1. Problem formulation

In multi-label learning, let us say that $\mathcal{L}$ is a label space with q labels. Multi-label instance i is represented as a tuple $\langle \mathbf{X_i}, \mathbf{Y_i} \rangle$, where $\mathbf{X_i}$ is the feature vector and $\mathbf{Y_i}$ the category vector of the instance i. Let us say $\mathbf{Y_i}$ is a binary vector of length q, where component $Y_{i\ell}$ represents whether the instance i belongs to the ℓ-th label or not. On the other hand, in pool-based AL scenarios, we have a small set of labeled data L_s and a large set of unlabeled data U_s.

Let us say Φ is a multi-label classifier which, for a given unseen instance, returns probabilities of belonging for each possible label $\ell \in \mathcal{L}$. Let us say $P_{\Phi}^{i,\ell}$ is the posterior probability that the instance i belongs to the ℓ-th label. So, the uncertainty of the

prediction of classifier Φ with respect to whether the given instance i belongs to the ℓ-th label can be computed as follows:

$$u_{\Phi}^{i,\ell}=|0.5 - P_{\Phi}^{i,\ell}| \tag{1}$$

where $|.|$ represents the absolute value function. This information-based criterion was proposed in [Lewis and Gale 1994; Lewis and Catlett 1994]. An instance i with a large $u_{\Phi}^{i,\ell}$ value means that the classifier Φ has little doubt in differentiating whether the instance belongs or does not belong to label ℓ. On the other hand, an instance i with small $u_{\Phi}^{i,\ell}$ value means that it is more ambiguous for the current classifier to predict whether the instance belongs or does not belong to label ℓ. So, we can compute the overall uncertainty of the classifier Φ by classifying the instance i as follows:

$$u_{\Phi}^{i}=\frac{\sum_{\ell \in \mathcal{L}} u_{\Phi}^{i,\ell}}{q} \tag{2}$$

This criterion for measuring the uncertainty of unlabeled instances was proposed in [Wu et al. 2014]. Let us say S is a batch of unlabeled instances of size k. We can compute the whole uncertainty of the batch of instances S as follows:

$$u(S)=\frac{\sum_{i \in \mathcal{S}} u_{\Phi}^{i}}{k} \tag{3}$$

The lower the value of $u(S)$, the greater the overall uncertainty of the elements that belong to the batch S. In order to measure the diversity among the unlabeled instances in S, we define the following diversity-based criterion:

$$d(S)=\frac{\sum_{i,j \in S} d_{\mathcal{F}}(i,j)}{k^2} \tag{4}$$

where $d_{\mathcal{F}}(i,j)$ is a distance function which measures the distance between two instances in the feature space. The higher the value of $d(S)$, the greater the diversity among the elements that belong to S. On the other hand, in order to measure the density around the unlabeled instances in the batch S, we define the following representativeness-based criterion:

$$r_i=\frac{\sum_{j \in U_s} d_{\mathcal{F}}(i,j)}{m} \tag{5}$$

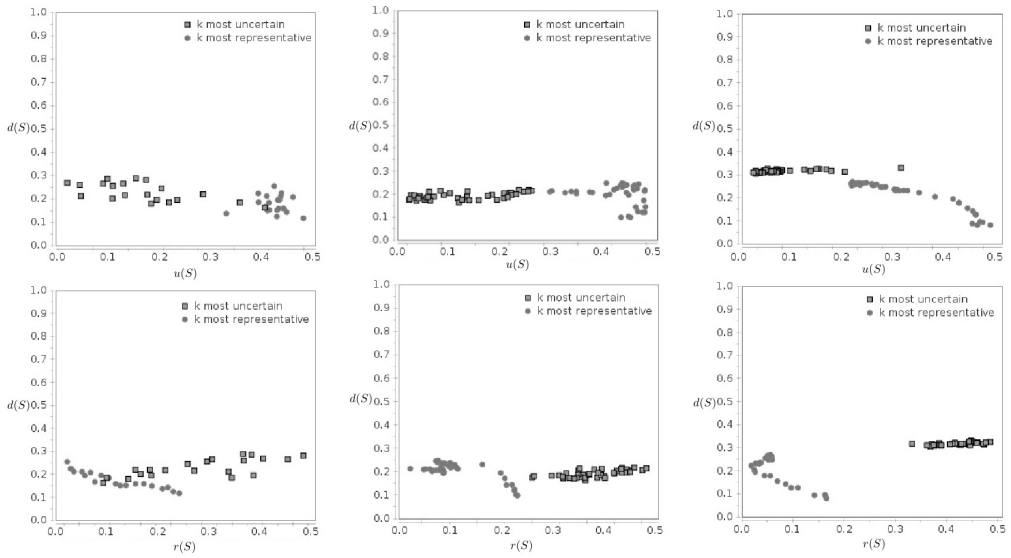
$$r(S)=\frac{\sum_{i \in S} r_i}{k} \tag{6}$$

where r_i is the density of the unlabeled instance i, and m is the number of instances that exist in the pool of unlabeled instances U_s.

The $r(S)$ criterion measures the density around the elements that belong to the batch S. It is based on the main idea of the information density framework described in [Settles and Craven 2008]. The lower the value of $r(S)$, the greater the density around the elements that belong to the batch S. In this work, we used the well-known Heterogeneous Euclidean Overlap Metric (HEOM) [Wilson and Martínez 1997] to compute the distance between two instances in the feature space. HEOM makes it possible to compute distance values on datasets where data are incomplete and have both continuous and nominal attributes [Deza and Deza 2009].

To illustrate the importance of querying batches of instances that have a good balance between the three criteria exposed above, we first perform an empirical study on various datasets. We designed two AL strategies, the first one selects the k most uncertain unlabeled instances in each iteration; the uncertainty value of an instance

is computed by means of the equation 2. The second strategy selects the k most representative unlabeled instances in each iteration; the density value of an instance is computed by means of the equation 5. For each batch of instances returned by the two AL strategies, the values of $u(S)$, $d(S)$ and $r(S)$ are computed, and these values are finally plotted for further analysis. BR-SVM is used as base classifier. Logistic regression models are fitted to the outputs of the SVMs to obtain proper probability estimates. We performed 50 iterations, in each iteration a batch of 15 instances was selected. The BR-SVM classifier was initially trained with a small number of labeled instances.



(a) Medical dataset. (b) Scene dataset. (c) Yeast dataset.

Fig. 2. Batches of instances selected.

Figure 2 shows the results of the experiments conducted on the Medical [Pestian et al. 2007], Scene [Boutell et al. 2004] and Yeast [Elisseeff and Weston 2001] datasets. The squares represent the batches of unlabeled instances that were selected by the first AL strategy, whereas the circles represent the batches of unlabeled instances selected by the second AL strategy. In the graphs that show the $u(S)$ vs. $d(S)$ relationship, we can observe that the batches of the most uncertain instances (square points) are those that have the smaller $u(S)$ values, as is expected. On the other hand, the batches of the most representative instances (circle points) usually have larger $u(S)$ values, indicating that they are less uncertain.

In the graphs that show the $r(S)$ vs $d(S)$ relationship, we can observe that the circle points have smaller $r(S)$ values, as is expected. On the other hand, the square points usually have larger $r(S)$ values, indicating that the unlabeled instances that compose the batches are less representative of the underlying distribution. In all graphs, we can also observe that both square and circle points have small $d(S)$ values, it seems that the batches of instances selected do not guarantee the diversity among their elements.

The experimental results shows us that the three criteria proposed seem to be in conflicting, i.e. achieving the optimal value for one criterion requires some compromise

on another. To achieve a good balance between the informativeness, representativeness and diversity criteria, we suggest to formulate the batch-mode multi-label AL as a multi-objective problem.

Definition 3.1. *Batch-mode multi-label AL problem*. Let us say Ω is the set of all possible subsets (batches) of k unlabeled instances. Select the batch of unlabeled instances that simultaneously optimizes multiple criteria.

$$\begin{array}{c} minimize\, O(S) = \ (O_1(S), O_2(S), \cdots, O_t(S)) \\ s.t.\ S \in \Omega \end{array} \tag{7}$$

where t is the number of objectives and O_i is the i-th objective.

In our case, we have three objectives $O_1(S)=u(S)$, $O_2(S)=r(S)$ and $O_3(S)=d(S)$. Without loss of generality, we assume $u(S)$, $r(S)$ and $d(S)$ are to be minimized. It is easy convert the $d(S)$ objective into a minimal objective. The problem is formulated as a multi-objective problem, where we are trying to find a batch(s) of unlabeled instances for which the three objectives reach better values. In multi-objective problems, we can find a set of optimal solutions. To compare batches of unlabeled instances, the concept of domination relation is defined as follows [Deb 2001].

Definition 3.2. *Domination*. Given two batches of unlabeled instances $S_1, S_2 \in \Omega$, S_1 dominates S_2 (denoted as $S_1 \preceq S_2$) if and only if:

$$\forall\, i \in \{1 \dots t\}\ O_i(S_1) \leq O_i(S_2)\ \wedge\ \exists\, i \in \{1 \dots t\}\ O_i(S_1) < O_i(S_2) \tag{8}$$

If $S_1 \not\preceq S_2$ and $S_2 \not\preceq S_1$, it is said that S_1 is non-dominated with regard to S_2. A batch S is said to be Pareto optimal if and only if S is not dominated by any other batch in Ω. The set of all Pareto optimal batches is called the Pareto optimal set or Pareto front [Deb 2001].

3.2. Genetic algorithm for solving the optimization problem

Pareto front cannot be computed efficiently in many cases. In pool-based AL scenarios, a large number of unlabeled instances are available. Consequently, even if it is possible to find all possible Pareto optimal batches, they are of exponential size. Therefore, we do not consider it practical to use computationally expensive methods to resolve the multi-objective problem formulated, since the oracle (e.g. a human annotator) would have to wait a considerable time before labeling a batch of unlabeled instances between each AL iteration.

This situation motivates us to apply evolutionary algorithms in order to find acceptable solutions in a reasonable time. The evolutionary algorithms have proven to be useful for resolving multi-objective problems in many domains [Deb 2001].

In this work, we propose to use the well-known Non-dominating Sorting Genetic Algorithm (NSGA-II) [Deb et al. 2002] to resolve the multi-objective problem formulated. NSGA-II is a multi-objective genetic optimization algorithm based on Pareto-optimal front with low computational requirements. NSGA-II reduces the high computational complexity of non-dominated sorting, the lack of elitisms and the need to specify the sharing parameter for insuring diversity in a population.

Note that, our proposal is not restricted to use NSGA-II as solver, indeed any existing evolutionary method for multi-objective optimization could be used, even a method that follow a non-evolutionary approach. Next, the main components of the genetic algorithm designed are briefly explained:

Encoding the chromosomes: Each chromosome represents a batch of unlabeled instances (see Figure 3). An integer representation is used for encoding the chromosomes.

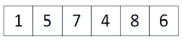


Fig. 3. The chromosome represents a batch of six unlabeled instances: i_1, i_5, i_7, i_4, i_8 and i_6.

Fitness function: The fitness function is a composite function by means of the informativeness, representativeness and diversity criteria. We try to simultaneously optimize three objective functions as represented in equation 7.

Generation of the initial population: Each individual chromosome is randomly coded. To code a chromosome, the first unlabeled instance is chosen and placed in the first position of the chromosome. The next instance is also chosen from among all except the first instance selected, and placed in the second position of the chromosome, and so on. This process is repeated as many times as there are individuals in the population. The probabilities of the unlabeled instances being selected are computed using their informativeness and representativeness values, so that the probability for the unlabeled instance i being chosen was proportional to $1/(1+e^{-1/(u_{\Phi}^{i}+r_i)})$. Unlabeled instances with better values of informativeness and representativeness are more likely to be selected.

Selection operator: We used the selection operator proposed in [Deb et al. 2002]. The selection process leads the algorithm toward a uniformly spread-out Pareto-optimal front. The NSGA-II algorithm assigns to each individual two attributes: non-domination front and crowding distance. Between two solutions with different non-domination fronts, the individual with the lower front is selected. Otherwise, if both individuals belong to the same front, then the individual that is located in a lesser crowded region is selected.

Crossover operator: We modified the classical Uniform Crossover in order to generate valid offspring. Given two parents p_1 and p_2, let us say S_1 and S_2 are their chromosomes viewed as sets of elements. To generate the first offspring, the difference S=$S_2 \setminus S_1$ is computed. The positions that have the elements of S in the chromosome of p_2 are marked in the chromosome of p_1. A copy of the chromosome of p_1 is made, and only the elements in the marked positions may be exchanged with the corresponding elements of the chromosome of p_2. To generate the second offspring, the same procedure is performed by swapping the parents.

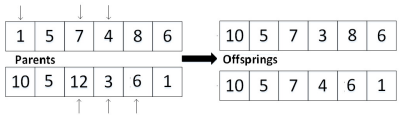


Fig. 4. Example of the crossover operator used. The small arrows indicate the marked positions.

Mutation operator: A simple one locus mutation operator is used. The operator randomly takes one gene and changes its value by a new unlabeled instance. The new unlabeled instance is selected with a proportional probability as used in the generation of the initial population. The operator also checks that the selected unlabeled instance does not exist in the chromosome.

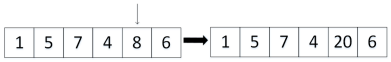


Fig. 5. One gene is randomly selected, and its value is changed by an unlabeled instance that does not exist in the chromosome.

Population update: NSGA-II algorithm follows an elitism criterion to prevent the loss of the fittest individuals. NSGA-II performs several steps to update the popula-

tion for next generation. Firstly, a population is created by the union of the current population and the intermediate population that results from the application of the selection, crossover and mutation operators. Secondly, the population of individuals is organized in fronts of non-dominated individuals. For each individual, the value of the front to which it belongs is assigned. Thirdly, an estimate of the density of solutions surrounding each individual is computed. The density of an individual is computed as the average distance between two points on either side of this individual along each of the objectives. Fourthly, to create the population for the next generation, individuals from the first front to the last front are added to a new population, while the size of the population is not completed. In the case where all individuals from certain front cannot be included in the new population, the individuals with less crowding distance are preferred. More information about the steps of the NSGA-II algorithm can be found in [Deb et al. 2002].

3.3. Active learning strategy

We defined an AL strategy named Evolutionary Strategy for Batch-Mode Multi-Label Active Learning (ESBMAL). ESBMAL aims to select a set of unlabeled instances from U_s, which are usually informative across all labels, diverse between each other, and representative of the underlying distribution. Algorithm 1 shows the main steps that the ESBMAL strategy follows to determine a batch on unlabeled instances in each AL iteration.

ALGORITHM 1: ESBMAL strategy.

Input : $GA \rightarrow$ genetic algorithm, $U_s \rightarrow$ pool of unlabeled instances, $\Phi \rightarrow$ multi-label classifier
Output: $S_{best} \rightarrow$ batch of unlabeled instances

begin
  // Create the initial population
  $GA.createInitialPopulation(U_s, \Phi)$;
  // Assign a random individual to S_{best}
  $S_{best} \leftarrow GA.selectRandomIndividual()$;
  for $iter \leftarrow 1$ **to** *GA.iterations* **do**
    // Select parents
    $GA.doSelection()$;
    // Cross parents
    $GA.doCrossover()$;
    // Mutate individuals
    $GA.doMutation()$;
    // Update the population for next generation
    $GA.doUpdate()$;
    // Obtain non-dominated individuals
    $P \leftarrow GA.getParetoFront()$;
    // Obtain individual with lower value of $u(S) + r(S) + d(S)$
    $S_{iter} \leftarrow getIndividual(P)$;
    if $S_{iter} \preceq S_{best}$ **then**
      $S_{best} \leftarrow S_{iter}$;
    end
    if $S_{iter} \not\preceq S_{best} \wedge S_{best} \not\preceq S_{iter}$ **then**
      // Obtain individual with lower value of $u(S) + r(S) + d(S)$
      $S_{best} \leftarrow min(S_{iter}, S_{best})$;
    end
  end
  return S_{best};
end

This approach must be used with base classifiers which can obtain proper probability estimates from their outputs. The new strategy is not restricted to base multi-label classifiers that use problem transformation methods, it can also be used with learning algorithms that directly handle the multi-label data (known as algorithm adaptation methods) [Tsoumakas et al. 2010; Gibaja and Ventura 2014].

Regarding the computational cost to determine a batch of unlabeled instances, let us say $f(\Phi)$ is the cost function of the classifier Φ to classify an instance. m is the number of unlabeled instances. N is the number of individuals contained within the population of the genetic algorithm, t the number of objectives, and g is the number of generations that the genetic algorithm performs. Since we have focused on a pool-base AL scenario, ESBMAL needs $O(m \cdot f(\Phi))$ steps to compute the informativeness of all unlabeled instances. On the other hand, the NSGA-II algorithm needs in one generation $O(t \cdot N^2)$ steps [Deb et al. 2002], in our case the number of objectives is equal to three. Therefore, the overall complexity of the genetic algorithm is $O(g \cdot N^2)$.

The computational complexity of ESBMAL is determined by the multi-label algorithm used as base classifier and the number of unlabeled instances, or the multi-objective algorithm used. In this work, we used NSGA-II as multi-objective algorithm, therefore ESBMAL strategy requires $O(max(m \cdot f(\Phi), g \cdot N^2))$ steps. On the other hand, one potential drawback of the ESBMAL strategy is that the number of required distance calculations grows with the number of unlabeled instances. However, the distance scores among each pair of unlabeled instances can be precomputed for an efficient lookup before executing the strategy for the first time.

4. EXPERIMENTAL STUDY AND DISCUSSION

In this section, the experiments that were carried out to evaluate the effectiveness of our proposal are described. In the experiments, 22 real multi-label datasets with different scale were used. Some statistics of the multi-label datasets are given on Table 2. In multi-label data, the label cardinality is the average number of labels per example, and the label density is the label cardinality divided by the total number of labels.

Table 2. Statistics of the benchmark datasets, number of instances (n), number of features (d), number of labels (q), different subsets of labels (d_s), label cardinality (l_c) and label density (l_d).

$Dataset$	$Domain$	$Source$	n	d	q	d_s	l_c	l_d
Arts	Text	[Ueda and Saito 2002]	7484	23146	26	599	1.654	0.064
Bibtex	Text	[Katakis et al. 2008]	7395	1836	159	2856	2.402	0.015
Birds	Audio	[Briggs and et. al. 2013]	645	260	9	133	1.014	0.053
Business	Text	[Ueda and Saito 2002]	11214	21924	30	233	1.599	0.053
Cal500	Music	[Turnbull et al. 2008]	502	68	174	502	26.044	0.150
Computers	Text	[Ueda and Saito 2002]	12444	34096	33	428	1.507	0.046
Corel16k	Images	[Barnard et al. 2003]	13811	500	161	4937	2.867	0.018
Education	Text	[Ueda and Saito 2002]	12030	27534	33	511	1.463	0.044
Entertainment	Text	[Ueda and Saito 2002]	12730	32001	21	337	1.414	0.067
Emotions	Music	[Trohidis et al. 2008]	593	72	6	27	1.869	0.311
Enron	Text	[Klimt and Yang 2004]	1702	1001	53	753	3.378	0.064
Genbase	Biology	[Diplarisa et al. 2005]	662	1186	27	32	1.252	0.046
Health	Text	[Ueda and Saito 2002]	9205	30605	32	335	1.644	0.051
Medical	Text	[Pestian et al. 2007]	978	1449	45	94	1.245	0.028
Recreation	Text	[Ueda and Saito 2002]	12828	30324	22	530	1.429	0.065
Reference	Text	[Ueda and Saito 2002]	8027	39679	33	275	1.174	0.035
Scene	Images	[Boutell et al. 2004]	2407	294	6	15	1.074	0.179
Science	Text	[Ueda and Saito 2002]	6428	37187	40	457	1.450	0.036
Social	Text	[Ueda and Saito 2002]	12111	52350	39	361	1.279	0.033
Society	Text	[Ueda and Saito 2002]	14512	31802	27	1054	1.670	0.062
TMC2007-500	Text	[Tsoumakas and Vlahavas 2007]	28596	500	22	1341	2.160	0.098
Yeast	Biology	[Elisseeff and Weston 2001]	2417	103	14	198	4.237	0.303

The performance of AL strategies was evaluated using several evaluation measures suggested in [Tsoumakas et al. 2010; Gibaja and Ventura 2014]. Due to lack of space, the formal definition of the evaluation measures used can be consulted in the electronic appendix. The label-based measures used in this work were the Micro-Average F_1-Measure (M_{iF_1}) and Macro-Average F_1-Measure (M_{aF_1}). The micro approach aggregates the true positive, true negative, false positive and false negative values of all labels, and then calculates the F_1-measure. The macro approach computes the F_1-measure for each label and then the values are averaged over all labels. The bipartition-based measures used were the Hamming Loss (H_L) and Example-based F_1-Measure (F_{1Ex}). H_L averages the symmetrical differences among the predicted and

actual label sets, while F_{1Ex} calculates the F_1-Measure over all examples in the test set. The ranking-based measures used were the Ranking Loss (R_L), Average Precision (A_P) and One Error (O_E). R_L averages the proportion of label pairs that are incorrectly ordered. A_P averages how many times a particular label is ranked above another label which is in the true label set. O_E averages how many times the top-ranked label is not in the set of true labels of the instance.

Generally speaking, the AL strategies are evaluated by means of constructing learning curves, plotting an evaluation measure as a function of the number of labeled instances. Through a visual inspection, a strategy is considered superior to the alternatives if it dominates them for most of the points along their learning curves [Settles 2012]. The visual inspection of the learning curves provides a qualitative intuition of which strategy performs better. However, visually comparing several learning curves can be very confusing, as several intersections among the learning curves could occur. In this work, the Area Under Learning Curve (AULC) was used to compare AL strategies in a quantitative manner. In the case where two AL strategies were compared, the Wilcoxon signed-ranks test [Wilcoxon 1945] was performed. In the case where more than two AL strategies were compared, the Friedman test [Friedman 1940] was performed to evaluate whether there was a significant difference in the results. If the Friedman test indicated that the results were significantly different, the Hommel procedure [Hommel 1988] was used to perform multiple comparisons with a control method as proposed in [García et al. 2010]. In the statistical analysis, the Adjusted p-values (APVs) [Wright 1992] were considered. APVs take into account the fact that multiple tests are conducted and can be compared directly with any significance level [García et al. 2010].

In all experiments, a 10-fold cross validation method was carried out 10 times with different seeds. The average of the test results was calculated. For each fold execution the iterative experimental protocol described in Algorithm 2 was adopted. This experimental protocol is similar to the experimental protocols previously used in [Yang et al. 2009; Esuli and Sebastiani 2009; Chakraborty et al. 2011; Huang and Zhou 2013; Zhang et al. 2014a].

ALGORITHM 2: Experimental protocol used in each fold execution.

Input : $T_r \rightarrow$ training set, $T_s \rightarrow$ test set, $\gamma \rightarrow$ multi-label AL strategy, $\theta \rightarrow$ oracle for labelling the unlabeled instances, $s \rightarrow$ number of sampling instances, $k \rightarrow$ number of instances to select in each iteration, $\beta \rightarrow$ maximum number of iterations
Output: $\Phi \rightarrow$ multi-label classifier

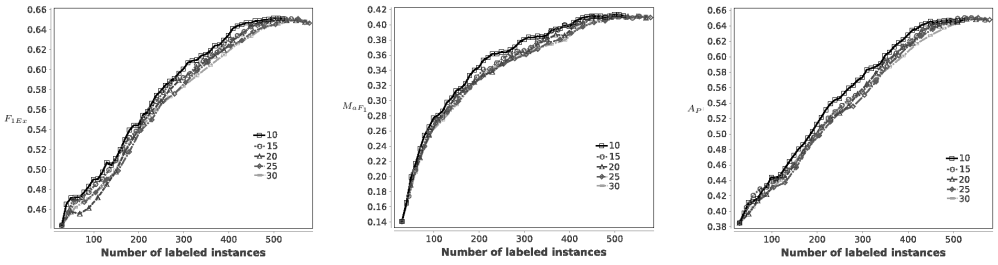
```
1  begin
       // Construct the labeled and unlabeled sets
2      L_s ← Resample (s, T_r);
3      U_s ← T_r \L_s;
4      for i ← 1 to β do
           // Train Φ with L_s
5          Φ ← Train (L_s,Φ);
           // Evaluate the effectiveness of Φ on T_s
6          Test (T_s, Φ);
           // Select the best subset of instances
7          S_i ← SelectSubsetInstances (γ, Φ, U_s, k);
           // Label the selected instances
8          Label (θ,S_i);
           // Update the labeled and unlabeled sets
9          L_s ← L_s ∪S_i ;
10         U_s ← U_s \S_i ;
11     end
12     return Φ;
13 end
```

The 5% of the training set T_r was randomly selected to construct the labeled set L_s. Therefore, the initial classifier was trained with few labeled instances. The non-selected instances of T_r form the unlabeled set U_s. For each unlabeled instance, its label set was hidden. The maximum number of iterations (β) was set at 50. In each iteration, the multi-label classifier Φ determined the significance of the L_s set in classifying the test set T_s. The labelling process was done in a simulated manner, namely the oracle reveals the hidden label sets of the unlabeled instances and they are added to the L_s set.

Regarding the parameters for the genetic algorithm used by ESBMAL, the number of individuals in the population was set at 100, the crossover operator was used with a probability of 90%, the mutation operator was used with a probability of 10%, and the number of generations performed was 100. For the sake of fairness, in the experiments, BR-SVM was used as base classifier, since most state-of-the-art multi-label AL strategies have been tested with BR-SVM as their base classifier. A linear kernel and a penalty parameter equal to 1.0 were used, as proposed in [Yang et al. 2009]. Logistic regression models are fitted to the outputs of the SVMs to obtain proper probability estimates.

4.1. Influence of the batch size in the ESBMAL strategy

We carried out a study to analyze the influence of the batch size in the performance of the ESBMAL strategy. Different executions of the ESBMAL strategy with batch sizes equal to 10, 15, 20, 25 and 30, were performed. In Figures 6-8, we see some graphs that represent the learning curves for the ESBMAL strategy on the Birds, Emotions and Yeast datasets. Figure 6 shows that the ESBMAL strategy obtained the best results with a batch size equal to 10. However, the difference in performance with other batch sizes was not significantly large. Similar behavior is showed in Figures 7 and 8.

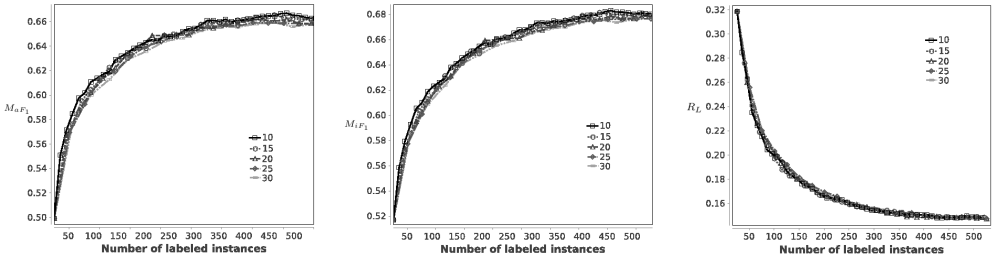


(a) Performance for F_{1Ex}. (b) Performance for M_{aF_1}. (c) Performance for A_P.

Fig. 6. Performance of the ESBMAL strategy on the Birds dataset with different batch sizes.

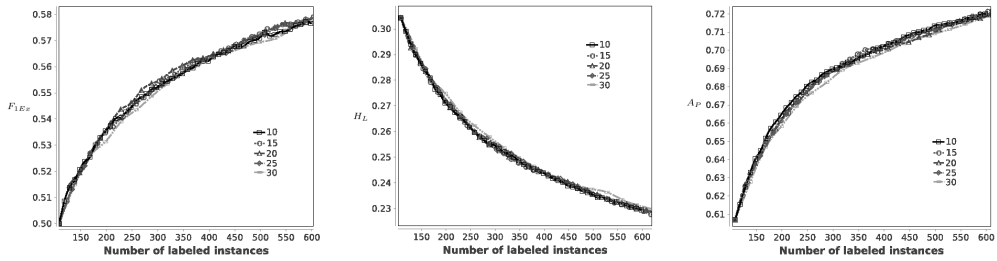
To compare the performance of the ESBMAL strategy with different batch sizes in a quantitative manner, the AULC values were calculated and then statistical tests were carried out. Tables with the AULC results for all evaluation measures considered in this work can be consulted on the electronic appendix. Table 3 summarizes the average rankings, the Friedman statistics and the p-values computed by the Friedman test for all evaluation measures considered. Note that, on average, the ESBMAL strategy obtained the best values for smaller batch sizes. However, in all cases, the Friedman test does not reject the null hypothesis at the significance level α=0.05. This means that, from a statistical point of view, the performance of the ESBMAL strategy was not significantly different when using different batch sizes.

The evidence shows, in some ways, the robustness of the ESBMAL strategy with respect to the number of unlabeled instances selected in each AL iteration. The search space is bigger as the batch size increases. In this cases, a possible way of increasing



(a) Performance for M_{aF_1}. (b) Performance for M_{iF_1}. (c) Performance for R_L.

Fig. 7. Performance of the ESBMAL strategy on the Emotions dataset with different batch sizes.



(a) Performance for F_{1Ex}. (b) Performance for H_L. (c) Performance for A_P.

Fig. 8. Performance of the ESBMAL strategy on the Yeast dataset with different batch sizes.

Table 3. Evaluation of ESBMAL strategy using different batch sizes. Summary of the Friedman test.

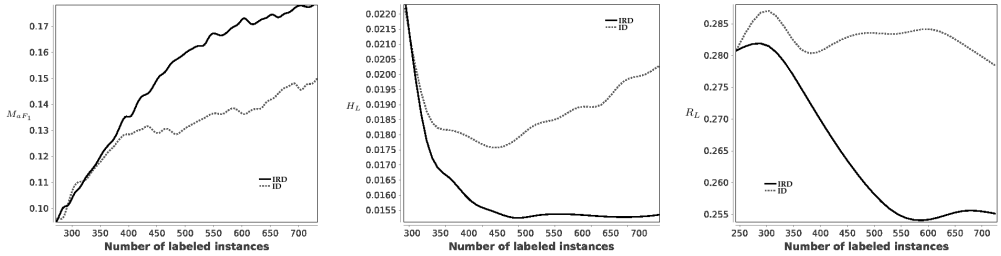
Batch	Measure						
	M_{iF_1}	M_{aF_1}	H_L	F_{1Ex}	A_P	O_E	R_L
10	2.523	2.364	2.727	2.705	3.068	2.773	2.386
15	2.591	2.954	2.364	2.477	2.750	2.682	3.114
20	3.477	2.682	3.046	3.136	2.568	2.841	3.046
25	3.227	3.477	3.432	3.046	3.023	3.364	2.773
30	3.182	3.523	3.432	3.636	3.591	3.341	3.682
Friedman statistic	6.227	8.882	7.518	6.918	5.309	3.754	7.990
p-value	0.183	0.060	0.111	0.140	0.257	0.440	0.092

the effectiveness of the ESBMAL strategy is by incrementing the number of generations of the genetic algorithm. However, by incrementing the number of generations, the ESBMAL strategy is more time consuming.

4.2. Benefit of combining the informativeness (I), diversity (D) and representativeness (R) criteria

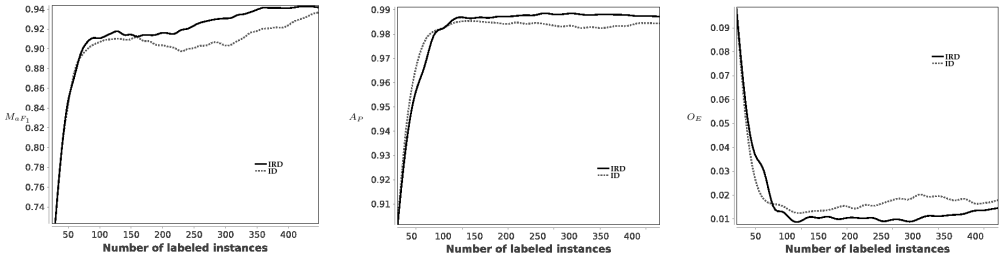
We carried out a study to analyze the benefit of combining the three criteria proposed in Section 3.1. We compared the ESBMAL strategy (in this section, we call it as IRD) with a version of ESBMAL that only uses the informativeness and diversity criteria (dubbed ID). The ID version is identical to the original ESBMAL proposed in Section 3.3, except that the representativeness (R) criterion is not taken into account. The ID version combines the two criteria that have been commonly used in the resolution of the batch-mode AL problem (I and D), and we want to analyze if with the inclusion of the R criterion a better performance is achieved.

Figures 9 and 10 represent the learning curves for the IRD and ID strategies on the Bibtex and Genbase datasets. Through visual inspection, we can observe that the IRD strategy obtained better results than the ID strategy.



(a) Performance for M_{aF_1}. (b) Performance for H_L. (c) Performance for R_L.

Fig. 9. Comparison between the IRD and ID versions of the ESBMAL strategy on the Bibtex dataset.



(a) Performance for M_{aF_1}. (b) Performance for A_P. (c) Performance for O_E.

Fig. 10. Comparison between the IRD and ID versions of the ESBMAL strategy on the Genbase dataset.

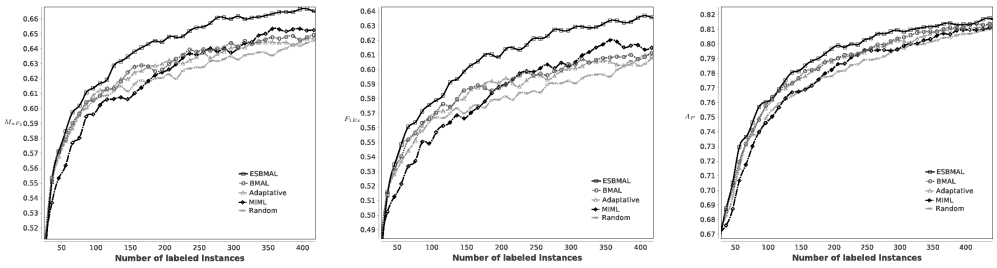
Due to lack of space, tables with the AULC values for all evaluation measures considered in this work can be consulted on the electronic appendix. Table 4 shows the p-values computed by the Wilcoxon signed-rank test. As we can see, in all cases, the Wilcoxon signed-rank test rejected the null hypothesis at the significance level α=0.05. This means that, from a statistical point of view, the performance combining I, R and D criteria was significantly better than that which only considered I and D criteria.

Table 4. Comparison between the IRD and ID versions of the ESBMAL strategy. Summary of the Wilcoxon signed-rank test.

	Measure						
	M_{iF_1}	M_{aF_1}	H_L	F_1Ex	A_P	O_E	R_L
p-value	0.000027	0.00003	0.000216	0.000029	0.000047	0.000055	0.00029

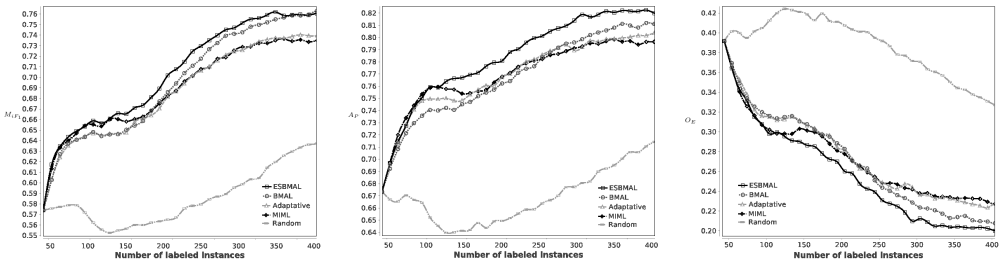
4.3. Comparison of ESBMAL with state-of-the-art multi-label AL strategies

For the sake of fairness, in this section we compared the ESBMAL strategy with state-of-the-art AL strategies that query all the labels assignments of the selected unlabeled instances, and combine two selection criteria (see Table 1 for more information). The BMAL [Chakraborty et al. 2011] strategy was included in the comparison since it is a batch-mode AL strategy and queries all the labels assignments. On the other hand, two myopic AL strategies, Adaptative [Li and Guo 2013] and MIML [Vasisht and Damianou 2014], were included in the comparison in order to compare the effectiveness of our proposal faced with this type of AL strategy. The Adaptative and MIML strategies query all the labels assignments, and combine informativeness and diversity selection criteria. In the comparison, a random sampling strategy (dubbed Random), that randomly chooses the instances from the available set of unlabeled instances, was also included.



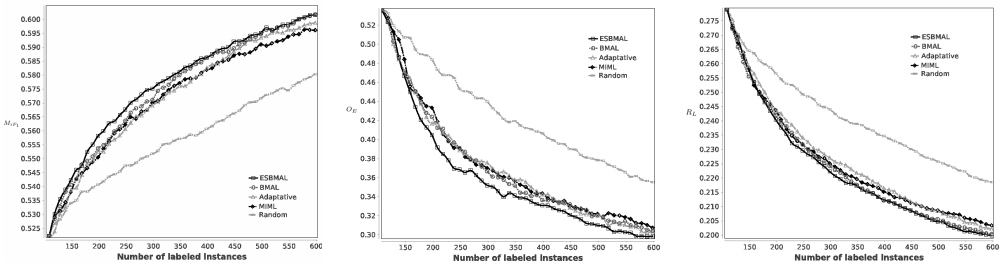
(a) Performance for M_{aF_1}. (b) Performance for F_{1Ex}. (c) Performance for A_P.

Fig. 11. Comparison between the ESBMAL strategy and state-of-the-art multi-label AL strategies on the Emotions dataset.



(a) Performance for M_{iF_1}. (b) Performance for A_P. (c) Performance for O_E.

Fig. 12. Comparison between the ESBMAL strategy and state-of-the-art multi-label AL strategies on the Medical dataset.



(a) Performance for M_{iF_1}. (b) Performance for O_E. (c) Performance for R_L.

Fig. 13. Comparison between the ESBMAL strategy and state-of-the-art multi-label AL strategies on the Yeast dataset.

In Figures 11-13, we see some graphs that represent the learning curves of the AL strategies on the Emotions, Medical and Yeast datasets. We can see that the ESBMAL strategy outperformed the other AL strategies. Also, all AL strategies considered in the comparison performed better than Random. Through a visual inspection taking into account the BMAL, Adaptative and MIML strategies, it is difficult to make a general conclusion which one performs better than other.

Due to lack of space, tables with the AULC results for all evaluation measures considered in this work can be consulted on the electronic appendix. Table 5a summarizes the average rankings, the Friedman statistics and the p-values computed by the Friedman test. Note that, on average, the ESBMAL strategy obtained the best results. In all cases, the Friedman test rejects the null hypothesis at the significance level α=0.05.

This means that, from a statistical point of view, there are significant differences in the performance of the multi-label AL strategies considered.

Table 5b shows the results of the Hommel procedure. We can reject the null hypothesis at the significance level α=0.05 in all cases, except in the cases of ESBMAL vs. BMAL for the A_P and O_E measures. The evidence showed that ESBMAL obtained better results than the myopic AL strategies considered. This means that ESBMAL selected batches of instances which were better than those selected by myopic AL strategies in a greedy manner. ESBMAL was also superior to the BMAL strategy in almost all evaluation measures considered in this work.

Table 5. Multiple comparison among the AL strategies

AL strategy	Measure						
	M_{iF_1}	M_{aF_1}	H_L	F_{1Ex}	A_P	O_E	R_L
ESBMAL	1.273	1.250	1.205	1.068	1.250	1.455	1.295
BMAL	2.250	2.341	2.432	2.023	1.841	2.205	2.454
Adaptative	3.136	3.136	3.091	3.364	3.227	3.227	3.159
MIML	3.341	3.273	3.318	3.636	3.682	3.205	3.204
Random	5.000	5.000	4.954	4.909	5.000	4.909	4.886
F. statistic	67.590	66.791	62.209	78.045	78.518	59.482	60.091
p-value	0.000	0.000	0.000	0.000	0.000	0.000	0.000

(a) Summary of the Friedman test.

ESBMAL vs.	Measure						
	M_{iF_1}	M_{aF_1}	H_L	F_{1Ex}	A_P	O_E	R_L
BMAL	0.040	0.022	0.010	0.045	0.215	0.116	0.015
Adaptative	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MIML	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Random	0.000	0.000	0.000	0.000	0.000	0.000	0.000

(b) The APVs computed by the Hommel procedure.

ESBMAL strategy performed well on multi-label datasets with diverse characteristics. However, the evidence suggested that ESBMAL had a smaller performance on datasets which have a high label cardinality and low label density, e.g. the Cal500, Corel16k, Scene, TMC2007-500 and Yeast datasets.

5. CONCLUSIONS

In this work, a new strategy to perform batch-mode AL on multi-label data was proposed, known as ESBMAL. Batch-mode AL was formulated as a multi-objective problem, where we aim to select batches of instances with a good balance of three criteria (informativeness, representativeness and diversity). A genetic algorithm was designed to resolve the multi-objective problem stated. The use of evolutionary algorithms allows us to find feasible solutions in a reasonable time. ESBMAL can be used with any multi-label classifier that can obtain proper probability estimates from their outputs.

The empirical study showed that the ESBMAL strategy performed well on multi-label datasets with diverse characteristics. Also, ESBMAL strategy obtained good results for the two tasks analyzed; Multi-Label Classification and Label Ranking. The evidence showed that a superior performance in the resolution of the batch-mode multi-label AL problem can be attained by means of combining the three selection criteria considered in this work. The evidence also suggested that the multi-objective problem stated, and the use of the evolutionary algorithms to resolve it, is a good approach to resolve the batch-mode multi-label AL problem.

Under the experimental settings considered in this work, the NSGA-II algorithm was effective in the solution of the multi-objective problem formulated. However, in future research, we are going to test the effectiveness of the AL strategy proposed with distinct multi-objective methods. It would be important to test the ESBMAL strategy with other multi-label learning algorithms as base classifiers. It would also be interesting to extend our proposal to query instance-labels pairs instead of querying all the labels assignments of the selected unlabeled instances. The selection of instance-labels pairs taking into account label correlations can lead to a considerable reduction of the labeling cost on multi-label data.

REFERENCES

K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. 2003. Matching Words and Pictures. *J. Mach. Learn. Res.* 3 (2003), 1107–1135.

M. Boutell, J. Luo, X. Shen, and C. Brown. 2004. Learning multi-label scene classification. *Pattern Recognit.* 37, 9 (2004), 1757–1771.

F. Briggs and et. al. 2013. The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *Proceedings of the International Workshop on Machine Learning for Signal Processing (MLSP'13)*. IEEE.

K. Brinker. 2006. *From Data and Information Analysis to Knowledge Engineering*. Springer-Verlag, Chapter On Active Learning in Multi-label Classification, 206–213.

S. Chakraborty, V. Balasubramanian, and S. Panchanathan. 2011. Optimal Batch Selection for Active Learning in Multi-label Classification. In *Proceedings of the 19th ACM international conference on Multimedia (MM'11)*. ACM, Scottsdale, Arizona, United States of America, 1413–1416.

C.Ye, J. Wu, V. Sheng, P. Zhao, and Z. Cui. 2015. Multi-label active learning with label correlation for image classification. In *IEEE International Conference on Image Processing (ICIP'15)*. IEEE, 3437–3441.

K. Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 2 (2002), 182–197.

M. M. Deza and E. Deza. 2009. *Encyclopedia of distances*. Springer.

S. Diplarisa, G. Tsoumakas, P. Mitkas, and I. Vlahavas. 2005. Protein Classification with Multiple Algorithms. In *Proceedings of the 10th Panhellenic Conference on Informatics (PCI'05)*. 448–456.

A. Elisseeff and J. Weston. 2001. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, T.G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Vol. 14. MIT Press, 681–687.

A. Esuli and F. Sebastiani. 2009. Active Learning Strategies for Multi-Label Text Classification. In *Advances in Information Retrieval*. Springer, 102–113.

M. Friedman. 1940. A Comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* 11 (1940), 86–92.

Y. Fu, X. Zhu, and A. K. Elmagarmid. 2013. Active Learning With Optimal Instance Subset Selection. *IEEE Trans. Cybern.* 43, 2 (2013).

S. García, A. Fernández, J. Luengo, and F. Herrera. 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* 180 (2010), 2044–2064.

E. Gibaja and S. Ventura. 2014. Multi-label learning: a review of the state of the art and ongoing research. *WIREs Data Mining Knowl. Discov.* 4 (2014), 411–444.

G. Hommel. 1988. A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika* 75, 2 (1988), 383–386.

S. Huang, S. Chen, and Z. Zhou. 2015. Multi-Label Active Learning: Query Type Matters. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAI Press, 946–952.

S. Huang, R. Jin, and Z. Zhou. 2014. Active Learning by Querying Informative and Representative Examples. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 10 (2014), 1936–1949.

S.J. Huang and Z.H. Zhou. 2013. Active query driven by uncertainty and diversity for incremental multi-label learning. In *Proceedings of 13th International Conference on Data Mining*. IEEE, 1079–1084.

C. W. Hung and H. T. Lin. 2011. Multi-label Active Learning with Auxiliary Learner. In *Proceedings of the Asian Conference on Machine Learning*. JMLR, 315–330.

I. Katakis, G. Tsoumakas, and I. Vlahavas. 2008. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD Discovery Challenge*.

B. Klimt and Y. Yang. 2004. The Enron corpus: a new dataset for email classification research. In *Proceedings of the 15th European Conference on Machine Learning (ECML'04)*. Springer, 217–226.

D. Lewis and J. Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh International Conference on Machine Learning*. 148–156.

D. Lewis and W. Gale. 1994. A sequential algorithm for training text classifier. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag, 3–12.

T. Li and M. Ogihara. 2003. Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval*. Washington DC, United States of America, 239–240.

X. Li and Y. Guo. 2013. Active Learning with Multi-Label SVM Classification. In *Proceedings of the Twenty-Third International joint Conference on Artificial Intelligence*. AAAI Press, 1479–1485.

X. Li, L. Wang, and E. Sung. 2004. Multi-label SVM active learning for image classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 4. IEEE, 2207–2210.

J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing (BioNLP'07). Association for Computational Linguistics*. Stroudsburg, PA, United States of America, 97–104.

G.J. Qi, X.S. Hua, Y. Rui, J. Tang, and H.J. Zhang. 2008. Two-dimensional active learning for image classification. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 1–25.

G.J. Qi, X.S. Hua, Y. Rui, J. Tang, and H.J. Zhang. 2009. Two-dimensional multi-label active learning with an efficient online adaptation model for image classification. *IEEE Trans. Pattern Anal. Mach. Intell* 99, 1 (2009).

B. Settles. 2012. *Active Learning* (1 ed.). Morgan & Claypool.

B. Settles and M. Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*. ACL Press, 1069–1078.

M. Singh, E. Curran, and P. Cunningham. 2009. Active Learning for Multi-Label Image Annotation. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*. 173–182.

Jinhui Tang, Zheng-Jun Zha, Dacheng Tao, and Tat-Seng Chua. 2012. Semantic-gap-oriented active learning for multilabel image annotation. *IEEE Trans. Image Process.* 21, 4 (2012), 2354–2360.

K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. 2008. Multilabel classification of music into emotions. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR'08)*. 325–330.

G. Tsoumakas, I. Katakis, and I. Vlahavas. 2010. *Data Mining and Knowledge Discovery Handbook* (2 ed.). Springer-Verlag, New York, United States of America, Chapter Mining Multi-label Data, 667–686.

G. Tsoumakas and I. Vlahavas. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML'07)*. Warsaw, Poland, 406–417.

D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. 2008. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio Speech Lang. Process.* 16, 2 (2008), 467–476.

N. Ueda and K. Saito. 2002. Parametric mixture models for multi-labeled text. In *Proceedings on Neural Information Processing Systems (NIPS'15)*. MIT Press, 737–744.

D. Vasisht and A. Damianou. 2014. Active Learning for Sparse Bayesian Multilabel Classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 472–481.

P. Wang, P. Zhang, and L. Guo. 2012. Mining Multi-label Data Streams Using Ensemble-based Active Learning. In *Proceedings of the 12th SIAM International Conference on Data Mining*. 1131–1140.

F. Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics* 1, 6 (1945), 80–83.

D. Wilson and T.R. Martínez. 1997. Improved heterogeneous distance functions. *J. Artif. Int. Res.* 6 (1997), 1–34.

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

J. Wu, V. Sheng, J. Zhang, P. Zhao, and Z. Cui. 2014. Multi-label Active Learning for Image Classification. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 5227–5231.

B. Yang, J.T. Sun, T. Wang, and Z. Chen. 2009. Effective Multi-Label Active Learning for Text Classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Paris, France, 917–926.

B. Zhang, Y. Wang, and F. Chen. 2014a. Multilabel Image Classification Via High-Order Label Correlation Driven Active Learning. *IEEE Trans. Ima. Process.* 23, 3 (2014), 1430–144.

J. Zhang, X. Wu, and V. S. Sheng. 2014b. Active Learning with Imbalanced Multiple Noisy Labeling. *IEEE Trans. Cybern.* 44, 3 (2014).

X. Zhang, J. Cheng, C. Xu, H. Lu, and S. Ma. 2009. Multi-view multi-label active learning for image classification. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'09)*. IEEE, 258–261.

Received ; revised ; accepted

TITLE:

JCLAL: A Java Framework for Active Learning

AUTHORS:

O. Reyes, E. Pérez, M. C. Rodríguez Hernández, H. M. Fardoun and S.Ventura

Journal of Machine Learning Research, *Volume 17 (95), pp.1-5, 2016*

RANKING:

Impact factor (JCR 2015): 2.450

Knowledge area:

Computer Science, Artificial Intelligence: 29/130

Journal of Machine Learning Research 17 (2016) 1-5 Submitted 7/15; Revised 5/16; Published 5/16

JCLAL: A Java Framework for Active Learning

Oscar Reyes OGREYESP@GMAIL.COM
Eduardo Pérez EPEREZP@FACINF.UHO.EDU.CU
Department of Computer Science
University of Holguín
Holguín, Cuba

María del Carmen Rodríguez-Hernández 692383@UNIZAR.ES
Department of Computer Science and Systems Engineering
University of Zaragoza
Zaragoza, Spain

Habib M. Fardoun HFARDOUN@KAU.EDU.SA
Department of Information Systems
King Abdulaziz University
Jeddah, Saudi Arabia

Sebastián Ventura SVENTURA@UCO.ES
Department of Computer Science and Numerical Analysis
University of Córdoba
Córdoba, Spain
Department of Information Systems
King Abdulaziz University
Jeddah, Saudi Arabia

Editor: Geoff Holmes

Abstract

Active Learning has become an important area of research owing to the increasing number of real-world problems which contain labelled and unlabelled examples at the same time. JCLAL is a Java Class Library for Active Learning which has an architecture that follows strong principles of object-oriented design. It is easy to use, and it allows the developers to adapt, modify and extend the framework according to their needs. The library offers a variety of active learning methods that have been proposed in the literature. The software is available under the GPL license.

Keywords: active learning, framework, java language, object-oriented design

1. Introduction

In the last decade, the study of problems which contain a small number of labelled examples and a large number of unlabelled examples at the same time have received special attention. Currently, there are two main areas that research the learning of models from labelled and unlabelled data, namely Semi-Supervised Learning and Active Learning (AL). AL is

concerned with learning accurate classifiers by choosing which instances will be labelled, reducing the labelling effort and the cost of training an accurate model (Settles, 2012).

Currently, there are several software tools which assist the experimentation process and development of new algorithms in the data mining and machine learning areas, such as Rapid Miner, WEKA, Scikit-learn, Orange and KEEL. However, these tools are focused to Supervised and Unsupervised Learning problems.

Some libraries and independent code that implement AL methods can be found on the Internet, such as Vowpal Wabbit, DUALIST, Active-Learning-Scala, TexNLP and LibAct. The Active-Learning-Scala and LibAct libraries are mainly focused to AL, they implement several AL strategies that have been proposed in the literature. On the other hand, Vowpal Wabbit, DUALIST and TexNLP have been designed for a different purpose, but they also include some AL methods.

To date, and in our opinion, there has been insufficient effort towards the creation of a computational tool mainly focused to AL. In our view, a good computational tool is not only a tool which includes the most relevant AL strategies, but also one that is extensible, user-friendly, interoperable, portable, etc.

The above situation motivated the development of the JCLAL framework. JCLAL is an open source software for researchers and end-users to develop AL methods. It includes the most relevant strategies that have been proposed in single-label and multi-label learning paradigms. It provides the necessary interfaces, classes and methods to develop any AL method.

This paper is arranged as follows: Section 2 provides a general description of the JCLAL framework. The Section 3 presents an example for using the software. Finally, the documentation and the requirements of this software are outlined in Section 4.

2. The JCLAL Framework

JCLAL is inspired by the architecture of JCLEC (Ventura et al., 2007; Cano et al., 2014) which is a framework for evolutionary computation. JCLAL provides a high-level software environment to perform any kind of AL method. It has an architecture that follows strong principles of object-oriented programming, where it is common and easy to reuse code. The main features of the library are the following:

- Generic. Through a flexible class structure, the library provides the possibility of including new AL methods, as well as the ability to adapt, modify or extend the framework according to developer's needs.

- User friendly. The library has several mechanisms that offer a user friendly programming interface. It allows users to execute an experiment through an XML configuration file.

- Portable. The library has been coded in the Java programming language. This ensures its portability between all platforms that implement a Java Virtual Machine.

- Elegant. The use of the XML file format provides a common ground for tools development and to integrate the framework with other systems.

- Open Source. The source code is free and available under the GNU General Public License (GPL). It is hosted at SourceForge, GitHub, OSSRH repository provided by Sonatype, and Maven Central Repository.

JCLAL aims to bring the benefits of machine learning open source software (Sönnenburg et al., 2007) to people working in the area of AL. The library offers several state-of-the-art AL strategies for single-label and multi-label learning paradigms. It uses the WEKA (Hall et al., 2009) and MULAN (Tsoumakas et al., 2011) libraries. WEKA is one of the most popular libraries which has several resources on supervised learning algorithms. On the other hand, MULAN is a Java library which includes several multi-label learning algorithms. For future versions, we hope to provide AL strategies related with multi-instance and multi-label-multi-instance learning paradigms.

Currently, the library provides the following single-label AL strategies: Entropy Sampling, Least Confident and Margin Sampling which belong to the Uncertainty Sampling category. Together with the Vote Entropy and Kullback Leibler Divergence strategies which belong to the Query By Committee category. In the Expected Error Reduction category, the Expected 0/1-loss and Expected Log-Loss strategies are included. One AL strategy belongs to the Variance Reduction family. The Information Density framework is also provided. More information about all of these single-label strategies can be found in (Settles, 2012). On the other hand, the following multi-label AL strategies are provided: Binary Minimum (Brinker, 2006), Max Loss (Li et al., 2004), Mean Max Loss (Li et al., 2004), Maximal Loss Reduction with Maximal Confidence (Yang et al., 2009), Confidence-Minimun-NonWeighted (Esuli and Sebastiani, 2009), Confidence-Average-NonWeighted (Esuli and Sebastiani, 2009), Max-Margin Prediction Uncertainty (Li and Guo, 2013) and Label Cardinality Inconsistency (Li and Guo, 2013).

The Stream-Based Selective Sampling and Pool-Based Sampling scenarios are supported. JCLAL provides the interfaces and abstract classes for implementing batch-mode AL methods and other types of oracle. Furthermore, the library has a simple manner of defining new stopping criteria which may change according to the problem. The library contains a structure which allows a set of listeners to simply define the events of an algorithm. The AL methods can be tested using the following evaluation methods: Hold-Out, k-fold cross validation, 5X2 cross validation and Leave-One Out. A method for actual deployment is also provided.

The library contains a set of utilities, e.g. algorithms for random number generation, sort algorithms, sampling methods and methods to compute, for example, AUC. A plug-in which permits the integration of the library with WEKA's explorer is also provided.

3. Using JCLAL

The library allows the users to execute an experiment through an XML configuration file as well as directly from Java code. A configuration file comprises a series of parameters required to run an algorithm. Below, an example of a configuration file is shown, which we call `MarginSampling.cfg`.

In this example, a 10-fold cross validation evaluation method is used on the data set `ecoli` located in the folder `datasets`. For each fold, 5% of the training set is selected to

construct the labelled set and the rest of the instances form the unlabelled set. A pool-based sampling scenario with the Margin Sampling strategy is used. The Naive Bayes algorithm is used as a base classifier.

```
<experiment>
 <process evaluation-method-type="net.sf.jclal.evaluation.method.kFoldCrossValidation">
  <rand-gen-factory seed="9871234" type="net.sf.jclal.util.random.RanecuFactory"/>
  <file-dataset>datasets/ecoli.arff</file-dataset>
  <stratify>true</stratify>
  <num-folds>10</num-folds>
  <sampling-method type="net.sf.jclal.sampling.unsupervised.Resample">
   <percentage-to-select>5.0</percentage-to-select>
  </sampling-method>
  <algorithm type="net.sf.jclal.activelearning.algorithm.ClassicalALAlgorithm">
   <stop-criterion type="net.sf.jclal.activelearning.stopcriteria.MaxIteration">
      <max-iteration>50</max-iteration>
   </stop-criterion>
   <stop-criterion type="net.sf.jclal.activelearning.stopcriteria.UnlabeledSetEmpty"/>
   <listener type="net.sf.jclal.listener.ClassicalReporterListener">
    <report-title>Margin-Sampling</report-title>
    <report-frequency>1</report-frequency>
    <report-directory>reports/ecoli</report-directory>
    <report-on-file>true</report-on-file>
   </listener>
   <scenario type="net.sf.jclal.activelearning.scenario.PoolBasedSamplingScenario">
    <batch-mode type="net.sf.jclal.activelearning.batchmode.QBestBatchMode">
     <batch-size>1</batch-size>
    </batch-mode>
    <oracle type="net.sf.jclal.activelearning.oracle.SimulatedOracle"/>
    <query-strategy type="net.sf.jclal.activelearning.singlelabel.querystrategy.
    MarginSamplingQueryStrategy">
     <wrapper-classifier type="net.sf.jclal.classifier.WekaClassifier">
       <classifier type="weka.classifiers.bayes.NaiveBayes"/>
     </wrapper-classifier>
    </query-strategy>
   </scenario>
  </algorithm>
 </process>
</experiment>
```

There are several ways to execute an experiment. One way is using the JAR file. For running the experiment just type:

```
java -jar jclal-1.0.jar -cfg "examples/MarginSampling.cfg"
```

After the experiment is run, a summary report which comprises information about the induced classifier and several performance measures is created.

4. Documentation, Requirements and Availability

The library is available under the GNU GPL license. A user manual and developer documentation which describes the software packages, examples, information to include new methods, API reference and running tests, is provided.

The software requires Java 1.7, Apache commons logging 1.1, Apache commons collections 3.2, Apache commons configuration 1.5, Apache commons lang 2.4, JFreeChart 1.0, WEKA 3.7, MULAN 1.4 and JUnit 4.10 (for running tests). There is also a mailing list and a discussion forum for requesting support on using or extending the framework.

Acknowledgments

This research was supported by the Spanish Ministry of Economy and Competitiveness, project TIN-2014-55252-P, and by FEDER funds.

References

K. Brinker. *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the Gesellschaft für Klassifikation e.V. University of Magdeburg*, chapter On Active Learning in Multi-label Classification, pages 206–213. Springer Berlin Heidelberg, 2006.

A. Cano, J. M. Luna, A. Zafra, and S. Ventura. A classification module for genetic programming algorithms in JCLEC. *Journal of Machine Learning Research*, 1:1–4, 2014.

A. Esuli and F. Sebastiani. *Advances in Information Retrieval: Proceedings of the 31th European Conference on IR Research (ECIR)*, chapter Active Learning Strategies for Multi-Label Text Classification, pages 102–113. Springer Berlin Heidelberg, 2009.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An Update. *SIGKDD explorations*, 11(1):10–18, 2009.

X. Li and Y. Guo. Active learning with multi-label SVM classification. In *Proceedings of the 23th International Joint Conference on Artificial Intelligence*, pages 1479–1485, 2013.

X. Li, L. Wang, and E. Sung. Multi-label SVM active learning for image classification. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 4, pages 2207–2210. IEEE, 2004.

B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 1st edition, 2012.

S. Sönnenburg, M. L. Braun, C. S. Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, K.-R. Müller, F. Pereira, C. E. Rasmussen, G. Rätsch, B. Schölkopf, A. Smola, P. Vincent, J. Weston, and R. C. Williamson. The need for open source software in machine learning. *Journal of Machine Learning Research*, 8:2443–2466, 2007.

G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. MULAN: a java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás. JCLEC: a java framework for evolutionary computation. *Soft Computing*, 12:381–392, 2007.

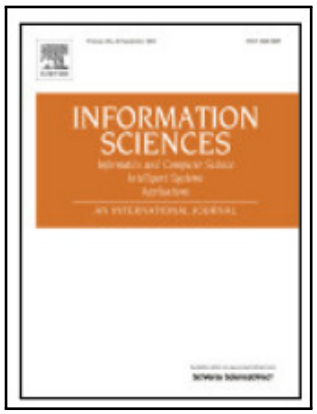
B. Yang, J. T. Sun, T. Wang, and Z. Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 917–926. ACM, 2009.

TITLE:

Statistical Comparisons of Active Learning Strategies over Multiple Datasets

AUTHORS:

O. Reyes, A. H. Altahi, and S. Ventura



Information Sciences, *submitted, 2016*

RANKING:

Impact factor (JCR 2015): 3.364

Knowledge area:

Computer Science, Information Systems: 8/143

Statistical Comparisons of Active Learning Strategies over Multiple Datasets

Oscar Reyes[a], Abdulrahman H. Altahi[b], Sebastián Ventura[a,b,*]

[a]*Department of Computer Science and Numerical Analysis, University of Córdoba, Spain*
[b]*Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

Abstract

Active learning has become an important area of research owing to the increasing number of real-world problems where a huge amount of unlabeled data is available. Active learning strategies are commonly compared by means of visually comparing learning curves. However, in cases where several active learning strategies are tested on multiple datasets, the visual comparison of learning curves may not be the best choice to decide whether a strategy is significantly better than another one. In this paper, two approaches are proposed, based on the use of non-parametric statistical tests, to statistically compare active learning strategies over multiple datasets. The application of the two approaches is illustrated by means of an experimental study, demonstrating the usefulness of the proposal for improving analysis of active learning performance.

Keywords: Active learning, Non-parametric statistical test, Area under learning curve, Rate of performance change, Active learning iterations

1. Introduction

Machine learning aims to construct computational algorithms able to determine general patterns from available data. In the learning process, not all data are useful, because noisy, redundant and incomplete data can affect in many ways the performance of learning algorithms. Consequently, the acquisition of a high-quality and compact dataset (a.k.a. training set) from which a learning algorithm can determine useful patterns is very important [1].

Sample selection is an important preprocessing step in data mining. Sample selection aims to select a representative subset from the original dataset, in such a manner that the performance of the learner generated from the selected subset will be the same (even higher) as if the original dataset is used [2]. The main advantages in applying sample selection methods are as follows [1–4]: reduce storage requirements by means of removing redundant information present in datasets, reduce computation effort in the classification of new patterns, increment the performance of learning algorithms by means of removing noisy points and outliers, enable learning algorithms to work effectively with large-scale datasets, and reduce the labeling cost.

Sample selection methods can be roughly classified into two categories [1]: instance selection and active learning. Instance selection aims to condense a dataset by filtering noisy and redundant data. Instance selection methods can be categorized into two groups [5]: wrapper methods where the selection criterion is based on the accuracy obtained by a learner, and filter methods where the selection criterion is not based on the results of a learner.

On the other side, active learning aims to process incomplete data, referring to data with missing labels, by means of selecting instances from unlabeled datasets, reducing the labeling effort and cost of training an accurate learner [6, 7]. Nowadays, we find many modern problems where a huge amount of unlabeled data is available. Sometimes, the labeling process may be subject to little or no cost. However, for many supervised learning tasks, data labeling is a time-consuming process that requires expert handling [6]. Successful applications of active learning include text

*Corresponding author. Tel:+34957212218; fax:+34957218630.
Email addresses: ogreyesp@gmail.com (Oscar Reyes), ahaltalhi@kau.edu.sa (Abdulrahman H. Altahi), sventura@uco.es (Sebastián Ventura)

categorization [8–10], image classification [11–13], protein structure prediction [14], natural language processing [15, 16], information retrieval [17, 18], information extraction [19, 20] and many more.

Active learning methods can be categorized according to the type of selection strategy (a.k.a. query strategy) used to iteratively select the unlabeled instances into [6]: uncertainty-based query [20–23], version space-based query [24–26], expected model change-based query [20, 27], expected error reduction-based query [28–30], variance reduction-based query [31–33], and density-weighted methods [20, 34, 35].

Active learning is similar to wrapper-based instance selection since they always have a learning algorithm involved in the process. However, in the active learning process an annotator (e.g. a human expert) is also required for labeling the selected unlabeled instances. Active learning is an iterative process, where in each iteration a selection strategy selects a set of unlabeled instances, the instances selected are labeled by the annotator, the instances are added to the training set, and the learning algorithm is trained with the new training set.

In this paper, we focus on active learning area and we aim to study the following problem: *Given n selection strategies that are tested on m datasets, determine whether the selection strategies differ significantly in performance*. In other words, we aim to study how to statistically compare active learning strategies over multiple datasets.

It is well known that it is not possible to find one algorithm which performs best for all problems [36, 37]. Consequently, the evaluation of experimental results is considered an essential part of any research, and over the last few years, statistical tests have been increasingly used by authors to validate results and draw conclusions when comparing algorithms. Since the publication of Demšar's work [38], non-parametric statistical tests have been widely used to validate empirical results obtained by algorithms in areas such as machine learning [39, 40], data mining [41] and computational intelligence [41–44]. Non-parametric tests are preferred over parametric tests, due to the absence of strong limitations (normality, independence and homoscedasticity) regarding the kind of data to be analyzed [38].

In spite of the call made by the machine learning scientific community for a correct statistical analysis of published results, there has not been a rigorous use of statistical tests to compare the performance of active learning methods. To date, selection strategies have been commonly evaluated by means of visually comparing learning curves [6]. The visual comparison of learning curves provides a qualitative way to determine whether a selection strategy outperforms another one. However, visually comparing several learning curves can often be very confusing, as the curves may overlap. The visual comparison of learning curves is further complicated when several selection strategies with similar performances are compared over a large number of datasets.

In this work, two approaches are proposed, based on the use of non-parametric statistical tests, to compare active learning methods. The first approach is based on the analysis of the area under learning curve and the rate of performance change. The second approach considers the intermediate results derived from the active learning iterations.

An experimental study was conducted to illustrate the application of the two approaches proposed in this work, where four selection strategies were tested on 26 datasets, showing the usefulness of our proposal for a better comparison of selection strategies. To the best of our knowledge, this work is the first attempt at examining how to do statistical comparisons of active learning strategies over multiple datasets.

This paper is arranged as follows: Section 2 shows some basic definitions and the state-of-the-art in the evaluation of active learning performance. Section 3 presents the two approaches proposed in this work to statistically compare active learning strategies. Section 4 describes the experimental study carried out in this work. Some guidelines are given in Section 5. Finally, Section 6 concludes the paper.

2. Preliminaries

In this section, the basic definitions used in this work are exposed. On the other hand, a review of the state-of-the-art techniques used for the evaluation of active learning performance is carried out.

2.1. Basic definitions

Let us say Φ is the base classifier used in an active learning process. L and U represent the labeled set and unlabeled set, respectively. θ is a selection strategy that selects a set of unlabeled instances from U using some selection criterion. Active learning is an iterative process, where the following steps are commonly performed in each iteration:

1. Φ is trained with the labeled set L.

2. The performance of Φ is tested.
3. θ selects a subset of unlabeled instances from U.
4. The selected unlabeled instances are labeled by an annotator (e.g. a human expert).
5. The selected instances are added to L and removed from U.

These steps are repeated iteratively. In active learning literature, several stopping conditions have been used. Commonly, the active learning process is repeated k times (number of iterations). However, we can use as stopping criterion when the performance of the base classifier attained a certain level. The way of testing the performance of the base classifier depends of the problem studied. Commonly, the performance of the base classifier is tested by means of using a test set and analyzing an evaluation measure.

Let us say L_i and U_i are the labeled set and unlabeled set, respectively, in the i-th iteration of the active learning process. Φ_i is the classifier constructed in the i-th iteration using L_i as training set.

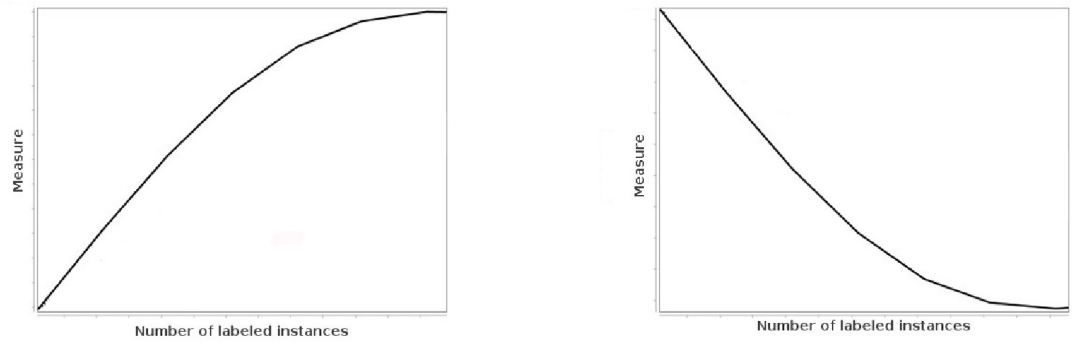
Definition 2.1. *Superiority of a classifier.* A classifier Φ_1 is considered superior to a classifier Φ_2, denoted as $\Phi_1 \succ \Phi_2$, if Φ_1 has a better performance than the classifier Φ_2, where both classifiers were tested under the same conditions.

Definition 2.2. *Ideal selection strategy.* A selection strategy is considered ideal if it is able to select in every iteration a set of unlabeled instances implying the construction of a classifier that it is superior than all classifiers generated in previous iterations.

The following axiom is directly derived:

Axiom 2.1. An active learning process that uses an ideal selection strategy generates a sequence of classifiers $\Phi_1, \Phi_2 \ldots, \Phi_k$ satisfying $\Phi_k \succ \Phi_{k-1} \succ \ldots \succ \Phi_1$.

An active learning process can be represented as a learning curve which plots the performance attained by the base classifier in every iteration. Figure 1 represents an active learning process using an ideal selection strategy. Figure 1a shows a case where the performance of the base classifier was tested in each iteration by means of analyzing a maximal evaluation measure. Figure 1b shows a case for a minimal evaluation measure. When a maximal measure is analyzed, the higher the value, the better the performance, whereas for a minimal measure the opposite occurs. Without loss of generality, we analyze in this work the case where the base classifier is tested by means of using a maximal evaluation measure.



(a) Maximal evaluation measure.

(b) Minimal evaluation measure.

Figure 1: Learning curves of an ideal selection strategy.

It is always desired that a selection strategy behaves as an ideal selection strategy. However, this condition is difficult to attain, the performance of a selection strategy can be biased in several ways by the base classifier used, the characteristics of the unlabeled data, etc. In particular iterations, a selection strategy can select a set of unlabeled instances that possibly could imply the construction of a classifier that is not superior in comparison to previous ones.

Let us say y_i is a numeric value that represents the performance of the classifier Φ_i, for instance y_i could be the value of an evaluation measure.

Definition 2.3. *Performance gain.* A performance gain is obtained if the classifiers constructed in the iterations i and j, where $j > i$, satisfy $\Phi_j > \Phi_i$. In performance gain conditions $y_j > y_i$.

If we analyze the results of two successive active learning iterations and a performance gain is evidenced, it indicates that the selection strategy was able to select a useful set of unlabeled instances that allowed to construct a superior classifier.

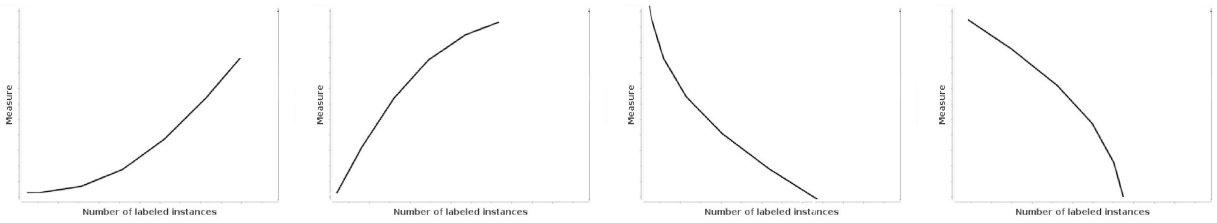
The definition of *performance loss* can be directly deduced from definition 2.3. In performance loss conditions $y_j \leq y_i$, where $j > i$. If we analyze the results of two successive active learning iterations and a performance loss is evidenced, it indicates that the selection strategy selected a set of unlabeled instances that did not allow to construct a superior classifier. In this case, the selection strategy could have selected redundant or noisy data.

Definition 2.4. *Rate of performance change*. The rate of performance change between two active learning iterations i and j, where $j > i$, is computed as follows:

$$m_{i,j} = \frac{y_j - y_i}{|L_j| - |L_i|} \tag{1}$$

where $|L_i|$ represents the number of instances in the labeled set of the i-th iteration.

Figure 2 shows four learning curves with different behaviors. Figure 2a shows an increasing learning curve, where $m_{i,i+1} > 0, \forall i = 1, 2, \ldots, k-1$. In this case, the curve is concave up, it means that the rates of performance change are increasing as the active learning process goes on. Figure 2b shows an increasing learning curve, however, the curve is concave down, the rates of performance change are decreasing. Figures 2c and 2d show two curves that are decreasing, and they are concave up and concave down, respectively.



(a) Increasing-concave up. (b) Increasing-concave down. (c) Decreasing-concave up. (d) Decreasing-concave down.

Figure 2: Learning curves with different behaviors.

Definition 2.5. *Turning point*. A turning point of a learning curve is when the performance changes from increasing to decreasing, or vice-versa.

2.2. State-of-the-art in the evaluation of active learning performance

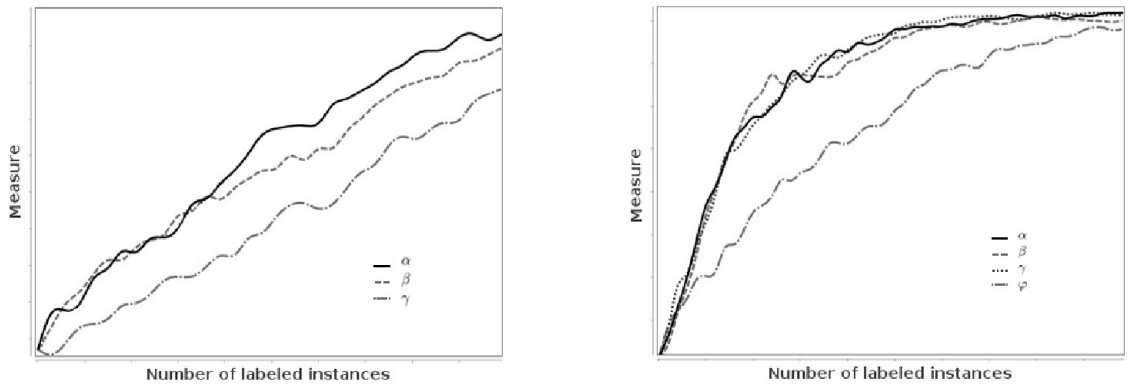
Generally speaking, selection strategies are commonly evaluated by means of visually comparing learning curves. Through visual inspection, a selection strategy is deemed superior to other strategies if it dominates them in most points along their learning curves [6].

Figure 3a shows the learning curves of three selection strategies. In this case, the visual comparison of the learning curves is an easy task to realize, since the difference in performance is visually obvious. Figure 3a shows that strategy α obtains better results than strategy β, and strategy β also obtains better results than strategy γ.

The visual comparison of learning curves is effective when a small number of selection strategies are compared, and their performances differ sufficiently so that the learning curves do not overlap greatly. If these conditions are satisfied, then visually comparing learning curves is an easy and intuitive way to compare selection strategies.

Conversely, the visual comparison of several selection strategies can be very confusing, as their learning curves may intersect at many points. If several selection strategies are tested over multiple datasets, and their performances are similar in many cases, the resulting learning curves may be very difficult to interpret, and the visual comparison of selection strategies may be a very difficult task to accomplish.

Figure 3b shows a case where four selection strategies are compared. In this case, we can conclude through a visual comparison that strategies α, β and γ outperform strategy φ. However, it is difficult to conclude whether there are differences in performance between strategies α, β and γ, as their learning curves overlap greatly.



(a) An easy visual comparison of the learning curves.

(b) A difficult visual comparison of the learning curves.

Figure 3: Visual comparison of learning curves.

The visual comparison of learning curves appears to be the most common way for comparing active learning strategies. We reviewed the papers from the proceedings of the six most recent editions (2011-2016) of the International Conference on Machine Learning (ICML), and the Workshop on Active Learning and Experimental Design at the International Conference on Artificial Intelligence and Statistics in year 2011 (AISTATS). Table 1 summarizes the number of papers focused on active learning that were published at these conferences.

	AISTATS	ICML					
	2011	2011	2012	2013	2014	2015	2016
Total of active learning papers	11	5	7	8	9	4	6
Papers that compare active learning strategies	8	3	5	4	5	3	4
Visually comparing learning curves	8	3	5	4	5	3	4
Statistical analysis	3	0	0	0	1	0	0

Table 1: A summary of the papers focused on active learning that were published at the International Conference on Machine Learning in years 2011-2016, and the Workshop on Active Learning and Experimental Design in year 2011.

We observed that many excellent and innovative works drew conclusions based on a visual comparison of learning curves (see Table 1). On the other hand, very few works drew conclusions by means of using statistical analysis. To date, to the best of our knowledge, a consensus on how to evaluate active learning performance quantitatively remains an open issue. In next section, we propose two different approaches to statistically compare active learning strategies over multiple datasets.

3. Evaluating active learning performance

In this section, the two approaches proposed in this work are described.

3.1. Analysing the area under learning curve and rate of performance change

In order to analyze the active learning performance quantitatively, we propose to analyze the area under the learning curve as a value that represents the performance of a selection strategy. The benefit of this type of evaluation, as opposed to visually comparing learning curves, is especially remarkable when several selection strategies are compared over a large number of datasets.

Definition 3.1. *Area Under Learning Curve* (AUC). The area under learning curve represents how much total performance of a classifier Φ has been accumulated as a result of increasing the labeled set L. The labeled set L is

iteratively increased by means of adding the unlabeled instances selected by a selection strategy. Thereby, an AUC value represents the overall performance of a selection strategy.

There are several methods to estimate the area under curve. In this work, we use one of the most simple, the trapezoidal rule.

Definition 3.2. AUC *of a selection strategy*. The AUC of the selection strategy θ, denoted as AUC_θ, is calculated as follows:

$$\text{AUC}_\theta = \frac{1}{2} \sum_{i=1}^{k-1} (y_{i+1} + y_i)(|L_{i+1}| - |L_i|) \tag{2}$$

where k is the number of active learning iterations, y_i is the performance of the classifier Φ_i, and L_i is the labeled set in the i-th iteration.

In case that a constant number of unlabeled instances, let us say b, are selected in all iterations of the active learning process, the AUC of the selection strategy θ is equal to $\frac{b}{2} \sum_{i=1}^{k-1} (y_{i+1} + y_i)$.

Analyzing AUC scores is a more robust and powerful way than visually comparing learning curves. The previous works [45, 46] used the AUC scores to measure the active learning performance. However, the main drawback of this evaluation form is that considering the AUC of a selection strategy, the cases where occur a performance gain (see definition 2.3) or a performance loss are not taken into consideration. The areas of trapezoids associated to a performance loss are linearly aggregated to areas of trapezoids associated to a performance gain. This situation provokes that two selection strategies which have curves with different behaviors could have equal AUC scores.

Figure 4 shows an example where two selection strategies have equal AUC. However, the strategy β shows a better performance than strategy α. The strategy β shows a performance gain in every pair of consecutive iterations, this is not the case of the strategy α. Note that, the strategy β corresponds to an ideal selection strategy (see the definition 2.2 and axiom 2.1).

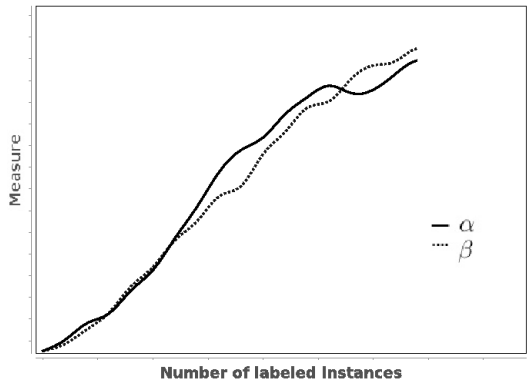


Figure 4: Example where two selection strategies have equal AUC.

The turning points (see definition 2.5) of a learning curve are of interest on the analysis of the performance of a selection strategy. It is desired that a learning curve is increasing in the most of active learning iterations, the number of turning points are minimum, and the rates of performance change are increasing.

Definition 3.3. *Positive Area Under Learning Curve* (PAUC). The positive AUC is defined as the summation of areas of trapezoids associated to a performance gain. The PAUC of the selection strategy θ, denoted as PAUC_θ, is calculated as follows:

$$\text{PAUC}_\theta = \frac{1}{2} \sum_{i=1}^{k-1} (y_{i+1} + y_i)(|L_{i+1}| - |L_i|) \quad : y_{i+1} > y_i \tag{3}$$

Definition 3.4. *Negative Area Under Learning Curve* (NAUC). The negative AUC is defined as the summation of areas of trapezoids associated to a performance loss. The NAUC of the selection strategy θ, denoted as NAUC_θ, is calculated as follows:

$$\text{NAUC}_\theta = \frac{1}{2}\sum_{i=1}^{k-1}(y_{i+1}+y_i)(|L_{i+1}|-|L_i|) \quad : y_{i+1} \leq y_i \tag{4}$$

Given a selection strategy θ, it is expected that $\text{PAUC}_\theta > \text{NAUC}_\theta$. If $\text{PAUC}_\theta \approx \text{NAUC}_\theta$, it could mean that the selection strategy θ has a similar performance to a strategy that randomly selects unlabeled instances.

On the other hand, the rates of performance change (see definition 2.4) give useful information about the increasing or decreasing in the performance of a selection strategy.

Definition 3.5. *Total of Positive Rates* (TPR). The total of positive rates of the selection strategy θ, denoted as TPR_θ, is computed as follows:

$$\text{TPR}_\theta = \sum_{i=1}^{k-1} m_{i,i+1} \quad : m_{i,i+1} > 0 \tag{5}$$

where $m_{i,j}$ is the rate of performance change between the iterations i and j.

Definition 3.6. *Total of Negative Rates* (TNR). The total of negative rates of the selection strategy θ, denoted as TNR_θ, is computed as follows:

$$\text{TNR}_\theta = \sum_{i=1}^{k-1} m_{i,i+1} \quad : m_{i,i+1} \leq 0 \tag{6}$$

Given a selection strategy θ, it is expected that $\text{TPR}_\theta > \text{TNR}_\theta$. Combining the measures PAUC, NAUC, TPR and TNR, the true performance of a selection strategy is computed as follows:

Definition 3.7. *True performance of a selection strategy* (TP). The true performance of the selection strategy θ, denoted as TP_θ, is defined as the difference between the positive AUC times the total of positive rates and the negative AUC times the total of negative rates.

$$\text{TP}_\theta = \text{PAUC}_\theta \cdot \text{TPR}_\theta - \text{NAUC}_\theta \cdot \text{TNR}_\theta \tag{7}$$

Given a selection strategy θ, the larger the TP_θ value, the better the performance of the strategy θ. A TP score can be interpreted as a general view of all active learning process. TP represents the overall performance of a selection strategy that was tested on a dataset.

Recall the problem stated in the introduction of this work, we have n active learning strategies that are tested on m datasets, and the problem is to analyze whether the active learning strategies differ significantly in performance. After computing the TP scores of the selection strategies on each dataset, a statistical analysis can be carried out, and final considerations may be given with a statistical support. Next, we briefly discussed the non-parametric tests that may be used to analyze the TP scores.

3.1.1. Summary of non-parametric statistical tests

Due to the absence of strong limitations, such as normality, independence and homoscedasticity in the collection of TP scores to be analyzed, we propose to conduct the statistical analysis by means of non-parametric statistical tests as recommended in study [38]. The selection of the non-parametric tests for an experimental study depends on the number of datasets and strategies, the type of comparison to be done, and if you want to detect less significant differences. Next, we briefly summarize the possible non-parametric tests that may be used in the statistical analysis of a collection of TP scores. The formal definitions of these tests, however, are not elaborated as it is beyond the scope of this work. For nice tutorials, you may consult the works [38, 39, 41].

Non-parametric tests can be classified by their capabilities to perform pairwise comparisons and multiple comparisons. The Sign test [47] and the Wilcoxon Signed-Ranks test [48] can be used to perform comparisons between

two selection strategies. However, the pairwise tests do not control the error propagation when more than two selection strategies are compared. The Friedman test [49], and its alternatives the Friedman Aligned-Ranks [50], Iman-Davenport correction [51] and Quade [52] tests may be used when comparing more than two selection strategies.

The Friedman test [49, 53] is the most well-known non-parametric test when there are more than two related samples. The Friedman test considers that the null hypothesis being tested is that all selection strategies obtain similar performances with non significant differences.

The Friedman Aligned-Ranks test [50] modifies the Friedman test, so that a global ranking is calculated instead of having a rank of methods for each dataset. In study [52], the Quade test was proposed, which scales the ranking of methods on each dataset depending on the differences observed in the algorithm's performance. These two proposals differ from the Friedman test in the ranking computation, and they may be used under the same circumstances as the Friedman test. However, in [41] it was shown that the use of the Friedman Aligned-Ranks and Quade tests is also suitable when the number of algorithms is low (not more than 4 or 5 algorithms). In addition, the authors concluded that the Quade test is very sensitive to the choice of datasets and it can report an excessive amount of significant differences, therefore its use has to be under justified circumstances and with special caution. On the other hand, in study [51], Iman and Davenport showed that the Friedman statistic is quite conservative and proposed a better measurement which is distributed according to the F distribution.

Statistical tests such as the Friedman, Friedman Aligned-Ranks, Quade and Iman-Davenport test are able to detect if there are significant differences in performance among a set of selection strategies. However, it is not possible to analyze if there are significant differences between a particular pair of selection strategies by means of these main tests. In this case, we can proceed with a post-hoc test to detect significant pairwise differences among selection strategies. A typical case would be to compare one selection strategy (commonly the strategy proposed by the author) with a set of state-of-the-art selection strategies. This type of comparison involves a control method where a family of hypotheses, all of which are related to the control method, is tested by conducting n-1 comparisons, where n is the number of strategies considered. On the other hand, under some circumstances, e.g. in review papers, it is interesting to perform all pairwise comparisons, therefore $n(n\text{-}1)/2$ comparisons are conducted.

The post-hoc tests can be categorized with respect to the way the value of the significance level is adjusted to compensate for multiple comparisons in: single-step, step-down and step-up [41]. Another important issue when using post-hoc procedures is the calculation of adjusted p-values [54]. Adjusted p-values take into account that multiple tests are conducted and they can be compared directly with any chosen significance level. Table 2 summarizes the post-hoc procedures that have been analyzed in studies [38–43].

Multiple comparisons	**Single-step**	**Step-down**	**Step-up**
Considering a control method	Bonferroni-Dunn procedure [55] Li procedure [56]	Holm procedure [57] Holland procedure [58] Finner procedure [59]	Hochberg procedure [60] Hommel procedure [61] Rom procedure [62]
All pairwise comparisons	Nemenyi procedure [63]	Holm procedure [57] Shaffer procedure [64] Bergman-Hommel procedure [65]	

Table 2: Summary of post hoc procedures analyzed in [38–43].

When performing multiple comparisons with a selection strategy as the control method, it is important to consider the following [38, 41]: The Bonferroni-Dunn test is a very conservative test and many differences may not be detected. On the other hand, the Holm, Hochberg, Hommel, Holland and Rom tests are more powerful than the Bonferroni-Dunn test. Although the Hommel and Rom test are the two most powerful, they are also the most difficult to apply and understand. The Finner test is a good alternative to the Hommel and Rom test, because it is powerful and easy to comprehend. The Li test is simpler than the Finner test, but it should be used with care.

Regarding to all pairwise comparisons among several selection strategies, it is important to consider the following [39]: The Nemenyi test is very conservative and many of the obvious differences may not be detected. The Holm, Shaffer and Bergamnn-Hommel tests are more powerful than the Nemenyi test. The Bergmann-Hommel test is the most powerful one, but it requires intensive computation when numerous algorithms are compared.

3.2. Analysing intermediate results

The previous section discussed the fact that several selection strategies can be statistically compared by means of using TP scores. However, with TP scores, important information derived from intermediate results (active learn-

ing iterations) of the active learning process is not completely considered. The analysis of intermediate results can reveal very significant information when selection strategies are compared, especially in cases where TP scores are statistically similar.

Definition 3.8. *Cut-point*. A cut-point represents a treatment among two selection strategies in a certain active learning iteration.

Let us say $y^{\theta}_{i,j}$ is the performance attained in the i-th iteration of the active learning process carried out over the j-th dataset, using the selection strategy θ.

Definition 3.9. *Cut-point score*. Given two selection strategies, α and β, and the cut-point associated to the i-th iteration of the active learning process carried out over the j-th dataset, the score of the cut-point is computed as follows:

$$c^{\alpha,\beta}_{i,j} = y^{\alpha}_{i,j} - y^{\beta}_{i,j} \tag{8}$$

Definition 3.10. *Ranking of cut-points for a dataset*. Given two selection strategies, α and β, the ranking of cut-points for the j-th dataset is an ordering of the cut-points taking into account their scores. The best cut-point is the one whose score is equal to $\max\{c^{\alpha,\beta}_{i,j} : 1 \leq i \leq k\}$, whereas the worst cut-point is the one whose score is equal to $\min\{c^{\alpha,\beta}_{i,j} : 1 \leq i \leq k\}$. The best cut-point has a rank value equal to one, the second best cut-point has a rank value equal to two, and so the worst cut-point has a rank value equal to k. If there is a tie among cut-points, the average rank values can be assigned.

Note that, in this work, we considered the number of cut-points equals to the number of active learning iterations (k). The number of cut-points can be less than the number of iterations, i.e. only a portion of the active learning iterations could be considered as cut-points. However, considering each active learning iteration as a cut-point results in a deeper analysis of active learning performance.

Definition 3.11. *Ideal ranking of cut-points*. Given two ideal selection strategies (see definition 2.2), α and β, where strategy α outperforms strategy β in each active learning iteration, and differences in performance among the two strategies increase as the active learning process goes on, the ideal ranking of cut-points is defined as follows:

$$Y = (k, k\text{-}1, \ldots, 1) \tag{9}$$

The smaller differences in performance (worst cut-points) will be at the beginning of the active learning process, whereas the bigger differences in performance (best cut-points) will be seen in the last iterations. We denoted Y_i as the ideal rank value of the i-th cut-point.

Figure 5 represents the learning curves of two ideal selection strategies. In this case an ideal ranking of cut-points would be generated.

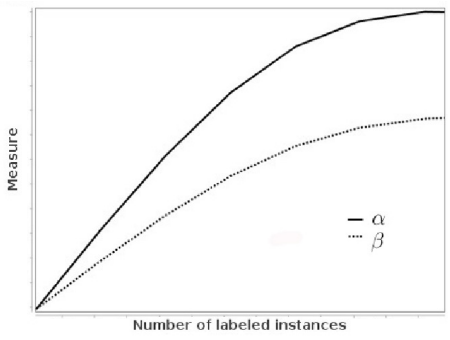


Figure 5: Example where an ideal ranking of cut-points would be generated.

In order to analyze intermediate results for the statistical comparison of selection strategies, we studied the following problem: *Given two selection strategies, α and β, which are tested on m datasets, determine whether strategy α significantly outperforms strategy β, using the results collected in active learning iterations.*

In this work, we resolve the problem before stated by means of the following two steps:

I Given the selection strategies α and β, compute a ranking of cut-points for each dataset, as shown in Table 3. The rank value of the i-th cut-point for the j-th dataset is denoted as r_{ji}.

Datasets	**Cut-points**			
	1	2	...	k
1	r_{11}	r_{12}	...	r_{1k}
2	r_{21}	r_{22}	...	r_{2k}
⋮	⋮	⋮	⋱	⋮
m	r_{m1}	r_{m2}	...	r_{mk}

Table 3: Rankings of cut-points.

II Determine if there is a significant amount of agreement between the ideal ranking of cut-points Y and the m observed rankings of cut-points.

Let us say R_i represents the mean rank for the i-th cut-point, and it is computed as follows:

$$R_i = \frac{\sum_{j=1}^{m} r_{ji}}{m}$$

If the null hypothesis is expressed as the equality between the mean rank of the k cut-points analyzed:

$$H_0 : R_1 = R_2 = \ldots = R_k$$

then we can do a test of the null hypothesis against the following ordered alternative:

$$H_1 : R_1 > R_2 > \ldots > R_k$$

The rejection of the null hypothesis would mean that the ideal ranking of cut-points is observed in the data, and therefore we can conclude that strategy α significantly outperforms strategy β.

So far, we have explained how to determine whether strategy α significantly outperforms strategy β, using the results collected in active learning iterations. Next, we explain a possible solution to carry out the statistical hypothesis testing formulated in the second step.

In this work, we propose to use a statistical test (dubbed as the Page trend test) that was proposed in study [66]. The Page trend test is a non-parametric statistical test, which can be classified under the family of tests for association in multiple classifications, similar to the Friedman test [44]. The Page trend test examines the hypothesis that treatments are ordered in a specific predictable sequence. In our case, the treatments are the cut-points, and the predicted sequence is represented by the ideal ranking of cut-points Y.

In study [66], Page defined the L statistic, which is calculated from a ranking matrix (see Table 3) as follows:

$$L = \sum_{i=1}^{k} (Y_i \sum_{j=1}^{m} r_{ji}) \tag{10}$$

The greater the L statistic, the lower the p-value for which the null hypothesis can be rejected. Appendix A shows demonstrations of the minimal, mean and maximal values of L statistic for the problem studied.

L critical values can be directly computed for small values of k (see, for example, table Q in [67] for values up to k=8 and m=12). In case that larger values are required, the normal approximation for the L statistic can be used to compute the asymptotic p-values with a continuity correction and the appropriate rejection region being the right tail. The normal approximation for the L statistic is computed as follows [67]:

$$z = \frac{12(L - 0.5) - 3mk(k+1)^2}{k(k+1)\sqrt{m(k-1)}} \tag{11}$$

The specific number of datasets and cut-points to consider will depend on the characteristics of each specific analysis and the available data, although in [67] it is exposed that the number of treatments (cut-points) should be approximately, at least twice the number of samples (datasets).

4. Experimental study

This section describes the experimental study carried out to illustrate the usefulness of our proposal for analyzing active learning performance.

4.1. Experimental setting

A large number of active learning methods have been proposed (see survey [6]). Due to the large number of existing active learning methods, in this work, we have focused on the simplest and most commonly used active learning framework, uncertainty sampling [21]. In our experimental study, three uncertainty sampling methods were compared: Margin Sampling [22], Least Confident [23] and Entropy Sampling [20]. In addition, a Random Sampling strategy was included as a base line in the experimentation. Random Sampling randomly chooses instances from the available unlabeled set.

All selection strategies used Support Vector Machines (SVM) as a base classifier. The SMO algorithm for SVMs, as presented in [68], was used. A linear kernel and a penalty parameter equal to 1.0 were used. Logistic regression models were fitted to the outputs of SVM to obtain proper probability estimates. These probabilities were used by the selection strategies.

We used 26 datasets from the UCI repository [69] (see Table 4). The datasets vary in size: from 150 up to 67557 instances, from 4 up to 45102 features, and from 3 up to 26 classes. All datasets are valid for the multi-class classification task.

Dataset	Instances	Features	Classes	Dataset	Instances	Features	Classes
Anneal	898	39	5	Multi-features	2000	650	10
Arrhythmia	452	280	13	Nursery	12960	9	5
Audiology	226	69	24	Optdigits	5620	65	10
Balance	625	5	3	Page-blocks	5473	11	5
BurkittLymphoma	220	22284	3	Pendigits	10992	17	10
Car	1728	7	4	Segment	2310	20	7
Connect	67557	43	3	Soybean	683	36	19
Dermatology	366	34	6	Vowel	990	14	11
Ecoly	336	8	8	Waveform	5000	41	3
Glass	214	10	7	Wine	178	14	3
Hypothyroid	3772	30	4	Yeast	1484	9	10
Iris	150	4	3				
Letter	20000	17	26				
Mfeat	2000	241	10				
Mousetype	214	45102	7				

Table 4: Summary description of datasets.

A 10-fold cross-validation was used and the selection strategies were executed ten times on each dataset, giving a total of 100 executions for each pair of strategy-dataset. Finally, the average of the results was calculated. For each fold execution, 5% of the training set was randomly selected to construct the labeled set. Therefore, the initial classifier was trained with few labeled instances. The non-selected instances of the training set formed the unlabeled set. For each unlabeled instance its class was hidden.

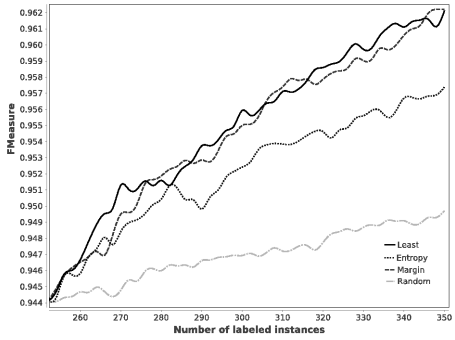
The maximum number of active learning iterations was set to 100. In each iteration, the selection strategies selected one unlabeled instance, and the performance of the base classifier was tested using a test set. The F-Measure was used to analyze the performance of the base classifier.

The labeling process was done in a simulated environment, i.e. the oracle reveals the hidden class of an unlabeled instance and the instance is added to the labeled set. A pool-based scenario to select the most informative unlabeled instance was used. In a pool-based scenario the entire unlabeled set is examined before selecting the most informative unlabeled instance [6].

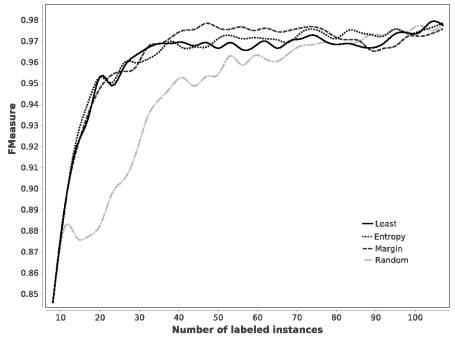
4.2. Visual comparison of learning curves

In this section, the visual comparison of learning curves is briefly illustrated by means of the experimental study carried out. Figures 6 and 7 represent the learning curves of the selection strategies on Optdigits, Wine, MFeat and Multi-features datasets.

Through a visual comparison, Figure 6a shows that Least Confident and Margin Sampling strategies obtained the best results, followed by Entropy Sampling strategy, on the Optdigits dataset. However, it is difficult to conclude whether the Least Confident strategy outperformed Margin Sampling strategy. Figure 6b shows that the three uncertainty sampling strategies outperformed Random sampling on the Wine dataset. However, it is difficult to conclude what is the uncertainty sampling strategy that obtained the best performance.



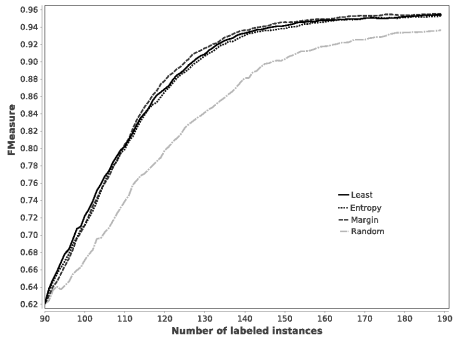
(a) Performance of the selection strategies on the Optdigits dataset.



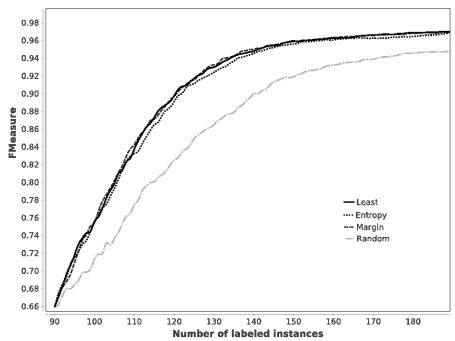
(b) Performance of the selection strategies on the Wine dataset.

Figure 6: Performance on Optdigits and Wine datasets.

Figures 7a and 7b show that Margin Sampling, Least Confident and Entropy Sampling strategies obtained better results than Random Sampling strategy on the MFeat and Multi-features datasets. However, a visual comparison among Margin, Least and Entropy strategies is difficult to carry out, owing to the fact that they had a similar performance on these datasets.



(a) Performance of the selection strategies on the MFeat dataset.



(b) Performance of the selection strategies on the Multi-features dataset.

Figure 7: Performance on MFeat and Multi-features datasets.

A total of 26 graphs of learning curves (one graph for each dataset) must be analyzed if the visual comparison of learning curves is used as the technique to analyze active learning performance in our experimental study. We only show the graphs of learning curves for four datasets, and we see that the visual comparison of learning curves produces a weak analysis. The results for many other datasets are similar, i.e. the Least Confident, Margin Sampling and Entropy Sampling strategies have similar performances. Consequently, conclusions from questions such as, "Which is the uncertainty sampling strategy that achieves the best performance?", are not possible, or very difficult to draw.

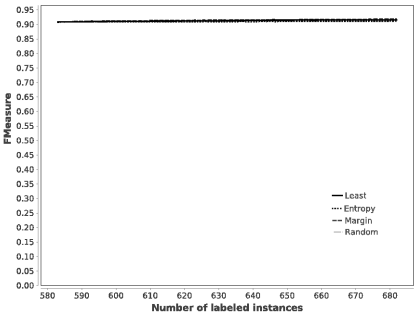
4.3. Analysing the TP scores

In this section, we proceeded to collect the TP scores of the selection strategies on each dataset. We used the Friedman test to conduct a multiple comparison and analyze if there were significant differences in the results[1]. Table 5 shows the TP scores of the selection strategies considered in the experimental study.

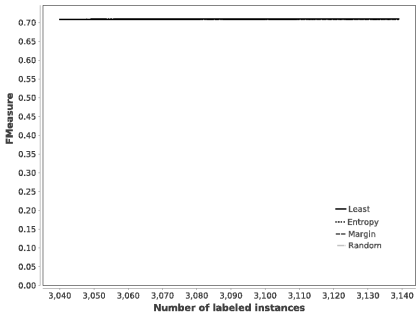
Dataset	**Selection strategy**			
	Entropy	**Least**	**Margin**	**Random**
Anneal	0.082 (2)	**0.087** (1)	0.080 (3)	0.060 (4)
Arrhythmia	**0.087** (1)	0.084 (2)	0.081 (3)	0.057 (4)
Audiology	0.218 (3)	0.223 (2)	**0.233** (1)	0.198 (4)
Balance	0.143 (2.5)	**0.144** (1)	0.143 (2.5)	0.108 (4)
BurkittLymphoma	**0.230** (1.5)	0.227 (3)	**0.230** (1.5)	0.179 (4)
Car	0.093 (2)	**0.094** (1)	0.089 (3)	0.049 (4)
Connect	**0.001** (1.5)	**0.001** (1.5)	0.000 (3.5)	0.000 (3.5)
Dermatology	**0.201** (1.5)	0.200 (3)	**0.201** (1.5)	0.178 (4)
Ecoli	0.115 (2.5)	**0.118** (1)	0.115 (2.5)	0.101 (4)
Glass	**0.134** (1)	0.123 (2)	0.118 (3)	0.084 (4)
Hypothyroid	0.034 (3)	**0.036** (1.5)	**0.036** (1.5)	0.020 (4)
Iris	**0.181** (1.5)	**0.181** (1.5)	0.180 (3)	0.176 (4)
Letter	0.003 (4)	0.006 (2.5)	**0.014**(1)	0.006 (2.5)
Mfeat	0.289 (3)	0.291 (2)	**0.293** (1)	0.264 (4)
MouseType	0.210 (2)	0.208 (3)	**0.219** (1)	0.187 (4)
Multi-Features	0.276 (3)	0.278 (2)	**0.279** (1)	0.247 (4)
Nursery	0.003 (3.5)	0.007 (2)	**0.010** (1)	0.003 (3.5)
Optdigits	0.013 (3)	0.017 (2)	**0.018** (1)	0.005 (4)
Page-blocks	0.011 (3)	0.015 (2)	**0.016** (1)	0.004 (4)
Pendigits	0.009 (3)	0.012 (2)	**0.021** (1)	0.006 (4)
Segment	0.046 (3)	0.051 (2)	**0.057** (1)	0.032 (4)
Soybean	0.234 (3)	**0.239** (1.5)	**0.239** (1.5)	0.179 (4)
Vowel	**0.086** (1)	0.083 (2)	0.066 (3)	0.040 (4)
Waveform	**0.030** (1)	0.024 (2)	0.014 (3)	0.012 (4)
Wine	0.124 (4)	0.126 (2)	**0.127** (1)	0.125 (3)
Yeast	0.029 (3)	**0.034** (1)	0.030 (2)	0.024 (4)
Average rank	2.404	1.865	1.865	3.865

Table 5: Comparison of the selection strategies. The rank values computed by Friedman test are showed between parentheses.

Note that, there are some cases where TP scores are close to 0, such as the TP scores obtained on the Nursery and Connect datasets. A TP score close to 0 could mean that the positive AUC and total of positive rates are approximately equal to the negative AUC and the total of negative rates. On the other hand, a TP score close to 0 could mean that the selection strategy selects unlabeled examples that do not significantly increase the performance of the base classifier, i.e. very small rates of performance change are obtained in all the iterations (see Figure 8).



(a) Performance of the selection strategies on the Nursery dataset.



(b) Performance of the selection strategies on the Connect dataset.

Figure 8: Performance on Nursery and Connect datasets.

The first step to calculate the Friedman statistic is to convert the original results into ranks. This test ranks the selection strategies for each dataset separately, where the best performing strategy has a rank value equal to one, the

[1]We have employed the software available at http://sci2s.ugr.es/sicidm to conduct all non-parametric tests used in the experimental study.

second best strategy has a rank value equal to two, and so on. In case of a tie, average ranks are computed. The procedure of the Friedman test is illustrated in Table 5. In the table, the last row shows the average rank computed by the Friedman test.

Average ranks by themselves provide useful information to compare selection strategies. In our experimental study, on average, the Margin Sampling and Least Sampling strategies ranked first with a rank value equal to 1.865, the Entropy Sampling ranked second with a rank value equal to 2.404, and as was expected, the last strategy was Random Sampling with a rank value equal to 3.865. We can say that on average the three uncertainty sampling strategies performed better than Random Sampling.

With four selection strategies, the Friedman statistic is distributed according to χ^2 distribution with three degrees of freedom. In our case, the Friedman statistic is equal to 41.792. The critical value using the χ^2 at α=0.05 with three degrees of freedom is 7.81. Due to the fact that the critical value is lower than Friedman statistic, we can reject the null hypothesis.

In machine learning papers where a hypothesis testing is carried out it is more common to report p-values to claim the rejection of the null hypothesis. The p-values indicate how significant the result is, the smaller the p-value, the stronger the evidence against the null hypothesis [39, 41]. In our case, the Friedman test rejected the null hypothesis with a p-value equal to 4.475×10^{-9}, so the null hypothesis is rejected with a high confidence level.

So far, we can say that there are significant differences in performance among the selection strategies. However, we cannot conclude anything about where these significant differences are located. To detect particular significant differences, we conducted a post-hoc test. In our experimental study, we were not interested in highlighting one specific selection strategy out of the four strategies compared. Consequently, we performed all pairwise comparisons among all selection strategies.

Table 6 shows the adjusted p-values obtained with the Nemenyi, Holm, Shaffer and Bergmann-Hommel tests. All post-hoc tests detected significant differences between uncertainty sampling strategies and Random sampling at α=0.01. Consequently, we can conclude that Margin Sampling, Least Confident and Entropy Sampling strategies significantly outperformed Random Strategy. However, neither of the post-hoc tests detected significant differences among the three uncertainty sampling strategies.

Hypothesis	**unadjusted p-value**	$p_{Nemenyi}$	p_{Holm}	$p_{Shaffer}$	$p_{Bergmann-Hommel}$
Least vs Random	0.0	0.0	0.0	0.0	0.0
Margin vs Random	0.0	0.0	0.0	0.0	0.0
Entropy vs Random	0.000045	0.000268	0.000179	0.000134	0.000089
Entropy vs Least	0.132622	0.795734	0.397867	0.397867	0.397867
Entropy vs Margin	0.132622	0.795734	0.397867	0.397867	0.397867
Least vs Margin	1	6	1	1	1

Table 6: Adjusted p-values computed by post-hoc tests.

4.4. Analyzing intermediate results

The previous section showed that the three uncertainty sampling strategies considered in the experimental study significantly outperformed Random Sampling. This conclusion was drawn with a statistical support by means of analyzing the TP scores. However, the tests were not able to detect differences in performance between the three uncertainty sampling strategies, as they showed similar overall performance when the TP scores were analyzed.

In this section, we conducted an analysis of intermediate results derived from active learning iterations as proposed in Section 3.2. The number of cut-points analyzed was equal to the number of active learning iterations (100).

Table 7 shows the rankings of cut-points per each dataset comparing the Margin Sampling with Random Sampling strategy. In the table, the last row represents the sum of rank values observed in the corresponding cut-points.

Rankings of cut-points by themselves provide useful information when comparing two selection strategies. Table 7 shows that smaller differences in performance among Margin Sampling and Random Sampling strategies are commonly concentrated at the beginning of the active learning process, i.e. in the first iterations[2]. We can see that, in some cases, larger differences in performance are concentrated at the end of the active learning process[3], e.g. in Arrhythmia,

[2]Larger rank values are assigned to cut-points with smaller scores.

[3]Smaller rank values are assigned to cut-points with larger scores.

Dataset	Cut-points												
	1	2	3	4	5	6	7	8	9	10	...	99	100
Anneal	100.0	99.0	98.0	91.0	87.5	87.5	97.0	83.0	83.0	83.0	...	76.5	76.5
Arrhythmia	63.0	58.5	60.5	65.5	65.5	62.0	71.0	72.5	76.0	76.0	...	1.0	4.0
Audiology	87.0	93.0	77.5	81.0	90.0	73.0	84.0	62.0	53.0	47.0	...	39.0	36.0
Balance	100.0	99.0	98.0	97.0	95.5	95.5	94.0	92.5	92.5	90.0	...	43.5	23.0
BurkittLymphoma	100.0	90.0	53.0	12.0	8.0	5.0	2.0	4.0	1.0	3.0	...	27.5	33.0
Car	99.5	99.5	98.0	96.5	96.5	93.0	92.0	94.0	95.0	90.5	...	16.5	41.5
Connect	50.5	100.0	50.5	1.0	50.5	50.5	50.5	50.5	50.5	50.5	...	50.5	50.5
Dermatology	100.0	83.0	62.0	48.5	28.5	21.0	19.0	15.0	8.0	6.0	...	98.5	98.5
Ecoli	100.0	99.0	97.0	44.0	67.5	74.0	87.5	62.5	44.0	27.0	...	81.5	87.5
Glass	97.0	96.0	98.0	99.5	99.5	93.0	91.0	89.5	93.0	95.0	...	19.5	32.5
Hypothyroid	100.0	99.0	97.0	93.5	97.0	97.0	95.0	93.5	91.5	89.5	...	10.0	10.0
Iris	23.0	3.0	19.0	29.5	60.0	63.0	61.5	65.0	18.0	23.0	...	10.5	16.5
Letter	99.5	92.0	92.0	92.0	99.5	92.0	92.0	92.0	77.5	77.5	...	10.5	10.5
Mfeat	100.0	98.0	99.0	97.0	96.0	82.0	73.0	70.0	55.0	52.5	...	94.0	94.0
MouseType	65.0	57.0	64.0	60.0	55.5	63.0	51.0	46.5	42.0	37.5	...	15.0	19.0
Multi-Features	100.0	97.0	98.0	99.0	96.0	79.0	62.0	54.0	46.5	44.0	...	88.0	88.0
Nursery	98.0	98.0	98.0	98.0	98.0	93.5	93.5	93.5	85.5	85.5	...	2.0	10.0
Optdigits	100.0	97.5	97.5	97.5	90.5	90.5	90.5	97.5	90.5	90.5	...	8.5	3.0
Page-blocks	100.0	98.0	98.0	98.0	95.5	95.5	93.5	93.5	90.5	90.5	...	8.5	8.5
Pendigits	100.0	98.5	98.5	95.5	95.5	91.5	95.5	95.5	91.5	91.5	...	3.5	3.5
Segment	100.0	97.5	97.5	97.5	97.5	95.0	94.0	93.0	90.5	90.5	...	4.5	24.5
Soybean	100.0	99.0	98.0	97.0	96.0	90.0	84.0	71.5	67.0	59.0	...	77.5	75.0
Vowel	94.5	96.0	99.0	97.0	100.0	94.5	98.0	91.5	84.0	91.5	...	2.0	2.0
Waveform	100.0	96.0	96.0	87.0	96.0	87.0	87.0	96.0	96.0	96.0	...	87.0	61.0
Wine	86.0	62.0	22.0	58.0	31.5	19.0	15.0	8.0	7.0	6.0	...	96.0	81.5
Yeast	99.0	93.5	65.5	81.0	74.0	87.0	97.0	87.0	90.5	97.0	...	24.0	11.0
Sum of ranks	2362	2299	2131.5	2013.5	2067.5	1974	1970.5	1873.5	1719.5	1690	...	995.5	1001

Table 7: Rankings of cut-points by means of comparing the Margin sampling with Random Sampling strategy.

Hypothyroid, Iris, Letter, MouseType, Nursery, Optdigits, Page-blocks, Pendigits and Vowel datasets. On the other hand, in datasets such as Anneal, Dermatology, Ecoli, MFeat, Multi-Features, Soybean and Wine, we can see that in the last iterations the Margin Sampling strategy had a similar or worse performance than Random Sampling. This behavior may indicate that, in the last iterations the Margin Sampling strategy selected unlabeled instances that did not yield an improvement on the performance of the base classifier.

Illustrating the calculation of the L statistic to compare the Margin Sampling with Random Sampling strategy:

$$L = \sum_{i=1}^{k}(Y_i \sum_{j=1}^{m} r_{ji}) = Y_1 \times 2362 + Y_2 \times 2299 + \ldots + Y_k \times 1001 = 100 \times 2362 + 99 \times 2299 + \ldots + 1 \times 1001 = 7322213.5$$

where the values $2362, 2299, \ldots, 1001$ are taken from the last row of Table 7. The associated p-value is equal to 0.000, therefore the null hypothesis can be rejected with a high confidence level. This means that there is a significant agreement between the ideal ranking of cut-points (Y) and the observed rankings of cut-points.

Table 8 shows the complete results for all pairwise comparisons among the four active strategies considered in the experimental study. The results show that Least Confident, Margin Sampling and Entropy Sampling strategies significantly outperformed Random Sampling strategy, and the null hypotheses were rejected with a p-value equal to 0.000. On the other hand, the evidence shows that the Least Confident strategy performed better than Entropy Sampling strategy. Furthermore, the Margin Sampling strategy performed better than Least Confident and Entropy Sampling strategies.

Hypothesis	L statistic	p-value
Margin Sampling vs. Least Confident	6766877.5	0.000711
Margin Sampling vs. Entropy Sampling	7045099.5	0.000000
Margin Sampling vs. Random Sampling	7322213.5	0.000000
Least Confident vs. Entropy Sampling	7184601	0.000000
Least Confident vs. Random Sampling	7306153.5	0.000000
Entropy Sampling vs. Random Sampling	6971644	0.000000

Table 8: All pairwise comparisons.

Finally, we concluded that, under the experimental study carried out in this work, the Margin Sampling strategy

showed the best performance, whereas Random Sampling showed the worst performance. In spite of the similar performance obtained by the three uncertainty sampling strategies, the evidence shows that the Margin Sampling strategy outperformed the other strategies when the intermediate results were analyzed.

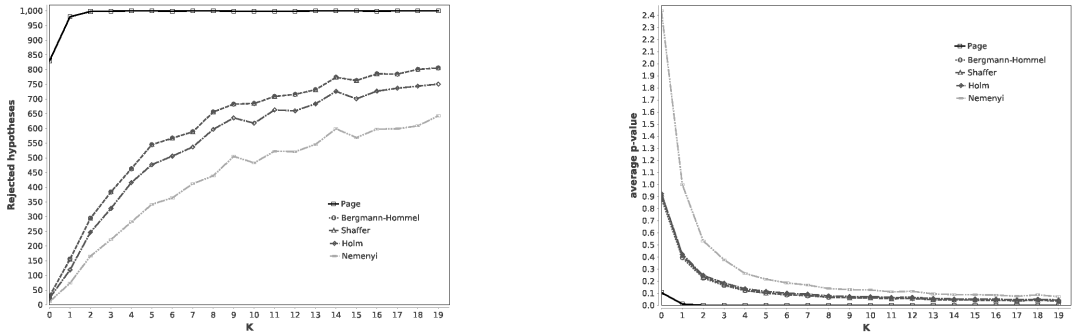
4.5. Power of statistical tests

In this section, the power of the non-parametric tests used in the experimental study is analyzed. The power of a statistical test is the probability that the test will correctly reject a false null hypothesis [38]. As power increases, the chance of committing a Type II error decreases.

We followed a similar method to the one used in studies [38, 39, 41] to determine the power of the non-parametric tests. We compared a pair of selection strategies 1000 times on 10 randomly chosen datasets, collecting the number of null hypotheses rejected (at a level of significance α=0.05) and the p-values.

As in work [38], the probability for the j-th dataset being chosen was proportional to $1/(1+e^{-Kd_j})$, where d_j is the (positive or negative) difference in performance among the two selection strategies on the j-th dataset, and K is the bias through which the differences between the selection strategies are regulated. With K=0, the selection is purely random, and for larger values of K the selected datasets are favorable to a particular selection strategy. For each K (k=0, 1, ..., 19), 10 datasets were selected 1000 times, and for each selection a non-parametric test was conducted. The difference in performance among two selection strategies on the j-th dataset (d_j) was calculated as the difference between their TP scores.

Figure 9 shows the results of this study considering the comparison between Margin Sampling and Entropy Sampling. Figure 9a shows the number of times that non-parametric tests rejected the equivalence between Margin Sampling and Entropy Sampling. Figure 9b shows the average p-values. As we can see, the test that analyzed intermediate results of active learning process (the Page trend test) is the most powerful one, as it rejected a larger number of hypotheses and reported the lower average p-values. This evidence shows the usefulness and power of considering intermediate results in the statistical comparison between two selection strategies.



(a) Number of rejected hypotheses.

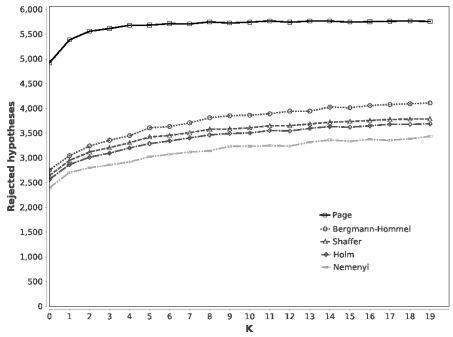
(b) Average p-values.

Figure 9: Margin Sampling vs. Entropy Sampling.

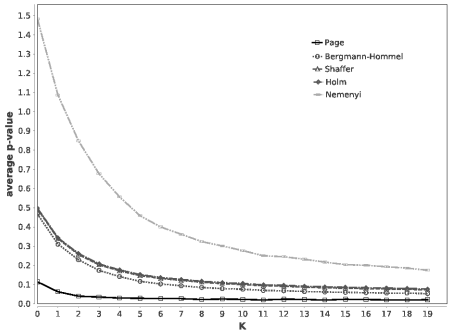
Figure 10 shows the results considering all pairwise comparisons between the four selection strategies compared. We can notice that the power of analyzing intermediate results is remarkable regarding to only analyze the TP scores. The large number of hypotheses rejected by means of analyzing intermediate results (Page trend test) shows us that this type of analysis can detect significant differences in many comparisons, even in comparisons where significant differences are not detected if we only analyze the TP scores (e.g. in comparisons between the Margin Sampling, Least Confident and Entropy Sampling strategies.)

On the other hand, between the four statistical tests which only consider the final results (TP scores) of the selection strategies, the Bergmann-Hommel test is the most powerful one, followed by the Shaffer test. The Nemenyi procedure is too conservative in comparison with the remaining procedures. This result matches the study carried out in [39], where the authors analyzed the power of several post-hoc tests for performing all pairwise comparisons.

Finally, we studied the impact of considering different numbers of cut-points in the statistical comparison using intermediate results of active learning process. Similar to the study of power conducted before, we compared a pair of selection strategies 1000 times on 10 randomly chosen datasets, using 10, 20, ..., 100 cut-points.



(a) Number of rejected hypotheses in all pairwise comparisons.

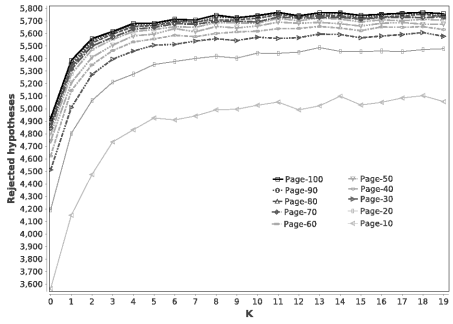


(b) Average p-values in all pairwise comparisons.

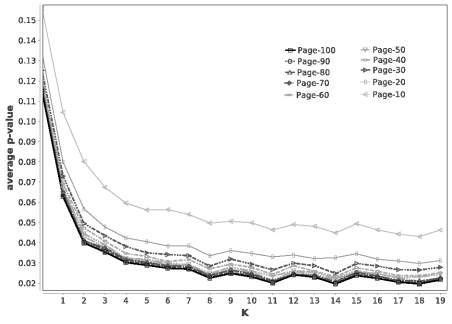
Figure 10: All pairwise comparisons.

Recall that, in the experimental study carried out in Section 4.4, the selection strategies performed 100 iterations on each dataset, and each active learning iteration was considered as a cut-point. In this study of power, we divided the 100 cut-points in ten bins with an equal number of cut-points; i.e. the first bin is from the first cut-point up to the tenth cut-point, the second bin is from the eleventh cut-point up to the twentieth cut-point, and so on. Suppose that, we want to perform a statistical comparison only considering 20 cut-points. In this case, we randomly select two cut-points (without replacement) from each bin of cut-points, therefore an equal number of cut-points per interval are selected.

Figure 11 shows the results of performing the statistical comparisons using different numbers of cut-points. We have denoted the Page trend test which uses 10 cut-points as Page-10, the Page trend test which uses 20 cut-points as Page-20, and so on. As was expected, the larger the number of cut-points considered, the greater the number of null hypotheses rejected. The evidence also shows that the Page trend test had a good stability, independently of the random selection of cut-points.



(a) Number of rejected hypotheses in all pairwise comparisons.



(b) Average p-values in all pairwise comparisons.

Figure 11: All pairwise comparisons considering different numbers of cut-points.

5. Summary and suggestions

Through the study conducted in this work, we present a summary and suggestions to compare active learning strategies:

- We encourage the use of non-parametric tests to analyze the effectiveness of active learning strategies. In many cases, it is not easy to do a visual comparison of the learning curves of selection strategies. When several selection strategies are compared over multiple datasets, the resulting graphs could be very difficult to interpret.

- Using the TP scores to analyze the active learning performance is a more powerful and robust procedure than visually comparing learning curves. However, we recommend the analysis of TP scores when the differences in the overall performance of the selection strategies are not very similar, since the statistical tests may not be able to detect significant differences between the selection strategies. In Section 3.1.1, we summarized several non-parametric tests that can be used to statistically compare a collection of TP scores.
- Analyzing intermediate results derived from active learning iterations can reveal very significant information when two selection strategies are compared, especially in cases where the TP scores are statistically similar. In this case, the Page trend test is a powerful non-parametric test which allows us to statistically compare two selection strategies by means of analyzing intermediate results.
- A significant number of active learning iterations should be considered as cut-points when two selection strategies are compared by means of analyzing intermediate results. The larger the number of cut-points selected, the better the analysis conducted and more reliable the conclusions drawn.
- Choosing cut-points knowing their success in favor of a certain selection strategy would be considered cheating. For the sake of fairness, for instance, a large number of active learning iterations can be randomly selected and they are considered as cut-points, or to simplify, all active learning iterations can be considered as cut-points.

6. Conclusions

In this paper, two approaches were proposed to statically compare several active learning strategies over multiple datasets. The first approach is based on the analysis of the area under learning curve and the rate of performance change. The second approach considers intermediate results derived from active learning iterations. The second approach is the most powerful, and its power is remarkable in those cases where the final results (TP scores) are statistically similar.

The two approaches allow us to draw meaningful conclusions with a statistical support. The use of statistical tests enhances the conclusions drawn about active learning performance, determining whether there is enough evidence to reject null hypotheses based on experimental results. The statistical tests studied in this work are not based on the assumptions of normality, independence or homoscedasticity of data to be analyzed, therefore, they can be widely used in a large number of application domains.

We recommend that when an active learning strategy is proposed, it should be compared to the most significant state-of-the-art strategies in a large number of datasets. The comparison of selection strategies should not only include a visual comparison of the learning curves, but a statistical analysis should be carried out as well. We encourage that such a statistical analysis be done by means of the two approaches proposed in this work.

Acknowledgements

This research was supported by the Spanish Ministry of Economy and Competitiveness, project TIN-2014-55252-P, and by FEDER funds.

Appendix A. Some properties of the *L* statistic

The definition of "*Cut-point*" and "*Ideal ranking of cut-points*" can be consulted in Section 3.2. The ideal ranking of cut-points is Y=(k,k-1, . . . ,1), Y_i=k-i + 1, where k is the number of cut-points considered.

Lemma 1. *The minimal value of the L statistic is equal to:*

$$L_{min} = \frac{mk(k+1)(k+2)}{6}$$

Datasets	Cut-points			
	1	2	...	k
1	1	2	...	k
2	1	2	⋯	k
⋮	⋮	⋮	⋱	⋮
m	1	2	⋯	k

(a) α does not outperform β in each active learning iteration.

Datasets	Cut-points			
	1	2	...	k
1	k	k-1	...	1
2	k	k-1	⋯	1
⋮	⋮	⋮	⋱	⋮
m	k	k-1	⋯	1

(b) α outperforms β in each active learning iteration.

Table A.9: Ranking matrices (α vs. β).

Proof. Given two selection strategies, α and β, the minimal value of the L statistic is reached when α does not outperform β in each dataset, such that differences increase in favor of β as the active learning process goes on. The ranking matrix between α and β is as shown in Table A.9a.

In this case, each ranking of cut-points is the inverse of the ideal ranking of cut-points Y. Therefore, $r_{ji}{=}i, 1 \leq j \leq m$ and $1 \leq i \leq k$. From the original equation of the L statistic, we obtain:

$$L{=}\sum_{i=1}^{k}(Y_i\sum_{j=1}^{m}r_{ji}){=}\sum_{i=1}^{k}(Y_i\sum_{j=1}^{m}i){=}\sum_{i=1}^{k}(m\cdot i\cdot Y_i){=}m\sum_{i=1}^{k}(i\cdot Y_i)$$

$$L{=}m\sum_{i=1}^{k}[i(k-i+1)]{=}m\sum_{i=1}^{k}\left[(k+1)i-i^2\right]{=}m\left[(k+1)\sum_{i=1}^{k}i-\sum_{i=1}^{k}i^2\right]$$

$$L{=}m\left[\frac{k(k+1)^2}{2}-\frac{k(k+1)(2k+1)}{6}\right]{=}mk(k+1)\left[\frac{(k+1)}{2}-\frac{(2k+1)}{6}\right]{=}\frac{mk(k+1)(k+2)}{6}$$

Lemma 2. *The mean value of the L statistic is equal to:*

$$L_{mean}{=}\frac{mk(k+1)^2}{4}$$

Proof. Given two selection strategies, α and β, the mean value of the L statistic is reached when the number of times that α outperforms β is as likely as the number of times that β outperforms α, in each dataset. L_{mean} is the expected value where all possible rank values of cut-points are taken to be equally likely. Therefore, $r_{ji}{=}\frac{(1+2+...+k)}{k}{=}\frac{(k+1)}{2}, 1 \leq j \leq m$ and $1 \leq i \leq k$. From the original equation of the L statistic, we obtain:

$$L{=}\sum_{i=1}^{k}(Y_i\sum_{j=1}^{m}r_{ji}){=}\sum_{i=1}^{k}\left[Y_i\sum_{j=1}^{m}\frac{(k+1)}{2}\right]{=}\sum_{i=1}^{k}\left[\frac{m(k+1)}{2}Y_i\right]{=}\frac{m(k+1)}{2}\sum_{i=1}^{k}(k-i+1)$$

$$L{=}\frac{m(k+1)}{2}\left[k\sum_{i=1}^{k}1-\sum_{i=1}^{k}i+\sum_{i=1}^{k}1\right]{=}\frac{m(k+1)}{2}\left[k^2-\frac{k(k+1)}{2}+k\right]{=}\frac{mk(k+1)^2}{4}$$

Lemma 3. *The maximal value of the L statistic is equal to:*

$$L_{max}{=}\frac{mk(k+1)(2k+1)}{6}$$

Proof. Given two selection strategies, α and β, the maximal value of the L statistic is reached when α outperforms β in each dataset, such that the differences increase as the active learning process goes on. The ranking matrix between α and β is as shown in Table A.9b.

In this case, each ranking of cut-points is equal to the ideal ranking of cut-points Y. Therefore, $r_{ji}{=}Y_i, 1 \leq j \leq m$ and $1 \leq i \leq k$. From the original equation of the L statistic, we obtain:

$$L_{max}{=}\sum_{i=1}^{k}(Y_i\sum_{j=1}^{m}r_{ji}){=}\sum_{i=1}^{k}(Y_i\sum_{j=1}^{m}Y_i){=}\sum_{i=1}^{k}(mY_i^2){=}m\sum_{i=1}^{k}Y_i^2$$

$$L_{max}=m\sum_{i=1}^{k}(k-i+1)^2=m\sum_{i=1}^{k}(k^2-2ki+2k+i^2-2i+1)=m(k^2\sum_{i=1}^{k}1-2k\sum_{i=1}^{k}i+2k\sum_{i=1}^{k}1+\sum_{i=1}^{k}i^2-2\sum_{i=1}^{k}i+\sum_{i=1}^{k}1)$$

$$L_{max}=m\left[k^3-k^2(k+1)+2k^2+\frac{k(k+1)(2k+1)}{6}-k(k+1)+k\right]=\frac{mk(k+1)(2k+1)}{6}$$

Lemma 4. *$L_{min} < L_{mean} < L_{max}, \forall k > 1$ and $m > 0$*

Proof. Analyzing L_{min} and L_{mean}:

$$\frac{mk(k+1)^2}{4} > \frac{mk(k+1)(k+2)}{6}$$

$$\frac{(k+1)}{2} > \frac{(k+2)}{3}$$

$$3k+3 > 2k+4$$

$$k > 1$$

So, $L_{min} < L_{mean}, \forall k > 1$ and $m > 0$. Analyzing L_{mean} and L_{max}:

$$\frac{mk(k+1)(2k+1)}{6} > \frac{mk(k+1)^2}{4}$$

$$\frac{(2k+1)}{3} > \frac{(k+1)}{2}$$

$$4k+2 > 3k+3$$

$$k > 1$$

So, $L_{mean} < L_{max}, \forall k > 1$ and $m > 0$. Finally, $L_{min} < L_{mean} < L_{max}, \forall k > 1$ and $m > 0$.

References

[1] X. Z. Wang, L. C. Dong, J. H. Yan, Maximum Ambiguity-Based Sample Selection in Fuzzy Decision Tree Induction, IEEE Transactions on Knowledge and Data Engineering 24 (8) (2012) 1491–1505.

[2] J. Zhai, X. W., X. Pang, Voting-based instance selection from large data sets with mapreduce and random weight networks, Information Sciences 367-368 (2016) 1066–1077.

[3] C. S. Pereira, G. D. C. Cavalcanti, Instance Selection Algorithm based on a Ranking Procedure, in: Proceedings of the International Conference on Neural Networks, IEEE, San Jose, California, USA, 2011, pp. 2409–2416.

[4] E. Leyva, A. Gónzalez, R. Pérez, A set of complexity measures designed for applying meta-learning to instance selection, IEEE Transactions on Knowledge and Data Engineering 27 (2) (2015) 354–367.

[5] P. Hernández-Leal, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. A. Olvera-López, InstanceRank based on borders for instance selection, Pattern Recognition 46 (2013) 365–375.

[6] B. Settles, Active Learning, 1st Edition, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool, 2012.

[7] Y. Fu, X. Zhu, B. Li, A survey on instance selection for active learning, Knowledge and Information Systems 35 (2013) 249–283.

[8] A. K. McCallum, K. Nigam, Employing EM in pool-based active learning for text classification, in: Proceedings of the International Conference on Machine Learning (ICML'1998), 1998, pp. 359–367.

[9] S. C. H. Hoi, R. Jin, M. R. Lyu, Large-scale text categorization by batch model active learning, in: Proceedings of the International Conference on the World Wide Web (WWW'2006), ACM Press, New York, 2006, pp. 633–642.

[10] S. Sohn, D. Comeau, W. Kim, W. Wilbur, Term-centric active learning for naive bayes document classification, Open Information Systems Journal 3 (2009) 54–67.

[11] S. H. C. Hoi, R. Jin, J. Zhu, M. R. Lyu, Batch mode active learning and its application to medical image classification, in: Proceedings of the 23rd International Conference on Machine Learning (ICML'2006), Pittsburgh, 2006, pp. 417–424.
[12] L. Copa, T. Devis, V. Michele, K. Mikhail, Unbiased query-by-bagging active learning for VHR image classification, in: Proceedings of the Conference on Image and Signal Processing for Remote Sensing XVI (ISPRS'2010), Vol. 7830, Toulouse, 2010.
[13] S. Vijayanarasimhan, P. Jain, K. Grauman, Far-sighted active learning on a budget for image and video recognition, in: Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2010), IEEE, San Francisco, 2010, pp. 3035–3042.
[14] M. W. Chang, L. Ratinov, N. Rizzolo, D. Roth, Learning and inference with constraints, in: Proceedings of the 23rd International Conference on Artificial Intelligence (AAAI'2008), Chicago, 2008, pp. 1513–1518.
[15] G. Tur, D. Hakkani-Tur, R. E. Schapire, Combining active and semi-supervised learning for spoken language understanding, Speech Communication 45 (2) (2005) 171–186.
[16] F. Olsson, A literature survey of active learning machine learning in the context of natural language procession, Tech. Rep. T2009:06, Swedish Institute of Computer Science (2009).
[17] C. Zhang, T. Chen, An active learning framework for content based information retrieval, IEEE Transactions on Multimedia 4 (2) (2002) 260–268.
[18] M. Wang, X. Hua, Active learning in multimedia annotation and retrieval: a survey, ACM Transactions on Intelligent Systems and Technology 2 (2) (2011) 3–23.
[19] R. Jones, R. Ghani, T. Mitchell, E. Riloff, Active learning for information extraction with multiple view feature sets, in: Proceedings of Adaptive Text Extraction and Mining (EMCL/PKDD'2003), Cavtat-Dubrovnik, 2003, pp. 26–34.
[20] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'2008), ACL Press, 2008, pp. 1069–1078.
[21] D. Lewis, W. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, ACM/Springer, 1994, pp. 3–12.
[22] T. Scheffer, C. Decomain, S.Wrobel, Active hidden markov models for information extraction, in: Proceedings of the International Conference on Advances in Intelligent Data Analysis (CAIDA'2001), Springer-Verlag, 2001, pp. 309–318.
[23] A. Culotta, A. McCallum, Reducing labeling effort for stuctured prediction tasks, in: Proceedings of the National Conference on Artificial Intelligence, AAAI Press, 2005, pp. 746–751.
[24] H. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proceedings of the ACM Workshop on Computational Learning Theory, 1992, pp. 287–294.
[25] N. Abe, H. Mamitsuka, Query Learning Strategies Using Boosting and Bagging, in: Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 1–10.
[26] P. Melville, R. J. Mooney, Diverse ensembles for active learning, in: Proceedings of the 21th International Conference on Machine Learning, Vol. 69, 2004, pp. 74–74.
[27] B. Settles, M. Craven, S. Ray, Multiple-instance active learning, in: Advances in Neural Information Processing Systems (NIPS), Vol. 20, MIT Press, 2008, pp. 1289–1296.
[28] N. Roy, A. Mccallum, Toward Optimal Active Learning through Sampling Estimation of Error Reduction, in: Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann, 2001, pp. 441–448.
[29] Y. Baram, R. El-Yaniv, K. Luz, Online Choice of Active Learning Algorithms, Journal of Machine Learning Research 5 (2003) 255–291.
[30] R. Moskovitch, N. Nissim, D. Stopel, C. Feher, R. Englert, Y. Elovici, Improving the detection of unknown computer worms activity using active learning, in: Proceedings of the German Conference on AI, Springer, 2007, pp. 489–493.
[31] D. Cohn, Neural network exploration using optimal experiment design, in: Advances in Neural Information Processing Systems (NIPS), Vol. 6, Morgan Kaufmann, 1994, pp. 679–686.
[32] D. Cohn, Z. Ghahramani, M. Jordan, Active learning with statistical models, Journal of Artificial Intelligence Research 4 (1996) 129–145.
[33] A. I. Schein, L. H. Ungar, Active learning for logistic regression: An evaluation, Machine Learning 68 (3) (2007) 235–265.
[34] A. Fujii, T. Tokunaga, K. Inui, H. Tanaka, Selective sampling for example-based word sense disambiguation, Computational Linguistics 24 (4) (1998) 573–597.
[35] H. T. Nguyen, A. Smeulders, Active learning using pre-clustering, in: Proceedings of the International Conference on Machine Learning (ICML), ACM Press, 2004, pp. 79–86.
[36] D. H. Wolpert, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.
[37] D. H. Wolpert, The supervised learning no-free-lunch theorems, in: Proceedings of the Sixth Online World Conference on Soft Computing in Industrial Applications, 2001.
[38] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research 7 (2006) 1–30.
[39] S. García, F. Herrera, An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[40] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (2009) 959–977.
[41] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences 180 (2010) 2044–2064.
[42] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 Special Session on Real Parameter Optimization, Journal of Heuristics 15 (2009) 617–644.
[43] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (2011) 3–18.
[44] J. Derrac, S. García, S. Hui, P. N. Suganthan, F. Herrera, Analyzing convergence performance of evolutionary algorithms: A statistical approach, Information Sciences 289 (2014) 41–58.
[45] G. C. Cawley, Baseline Methods for Active Learning, in: Workshop on Active Learning and Experimental Design, Vol. 16, 2011, pp. 47–57.
[46] R. Hu, Active learning for text classification, Ph.D. thesis, Dublin Institute of Technology (2011).

[47] R. G. D. Steel, A multiple comparison sign test: treatments versus control, Journal of the American Statistical Association 54 (1959) 767–775.
[48] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–83.
[49] M. Friedman, A comparison of alternative tests of significance for the problem of *m* rankings, Annals of Mathematical Statistics 11 (1940) 86–92.
[50] J. L. Hodges, E. L. Lehmann, Ranks methods for combination of independent experiments in analysis of variance, Annals of Mathematical Statistics 33 (1962) 482–497.
[51] R. L. Iman, J. M. Davenport, Approximations of the critical region of the Friedman statistic, Communications in Statistics (1980) 571–595.
[52] D. Quade, Using weigthed rankings in the analysis of complete blocks with additive block effects, Journal of the American Statistical Association 74 (1979) 680–683.
[53] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (1937) 674–701.
[54] S. Wright, Adjusted p-values for simultaneous inference, Biometrics 48 (1992) 1005–1013.
[55] O. J. Dunn, Multiple comparisons among means, Journal of the American Statistical Association 56 (1961) 52–64.
[56] J. Li, A two-step rejection procedure for testing multiple hypotheses, Journal of Statistical Planning and Inference 138 (2008) 1521–1527.
[57] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistic 6 (1979) 65–70.
[58] B. S. Holland, M. D. Copenhaver, An improved sequentially rejective Bonferroni test procedure, Biometrics 43 (1987) 417–423.
[59] H. Finner, On a monotonicity problem in step-down multiple test procedures, Journal of the American Statistical Association 88 (1993) 920–923.
[60] Y. Hochberg, A sharper Bonferroni procedure for multiple tests of significance, Biometrika 75 (1988) 800–803.
[61] G. Hommel, A stagewise rejective multiple test procedure based on a modified Bonferroni test, Biometrika 75 (1988) 383–386.
[62] D. M. Rom, A sequentially rejective test procedure based on a modified bonferroni inequality, Biometrika 77 (1990) 663–665.
[63] P. B. Nemenyi, Distribution-free multiple comparisons, Ph.D. thesis, Princeton University (1963).
[64] J. P. Shaffer, Modified sequentially rejective multiple test procedures, Journal of the American Statistical Association 395 (1986) 826–831.
[65] G. Bergmann, G. Hommel, Improvements of general multiple test procedures for redundant systems of hypotheses, in: P. Bauer, G. Hommel, E. Sonnemann (Eds.), Multiple hypothesis testing, Springer, 1988, pp. 100–115.
[66] E. Page, Ordered hypotheses for multiple treatments: a significance test for linear ranks, Journal of the American Statistical Association 58 (1963) 216230.
[67] J. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, 5th Edition, Chapman & Hall, 2010.
[68] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.
[69] M. Lichman, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2013).
URL `http://archive.ics.uci.edu/ml`

Conference publications

- O. Reyes, C. Morell and S. Ventura. ***Learning Similarity Metric to improve the performance of Lazy Multi-label Ranking Algorithms***. In Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA'2012). IEEE, pp. 246-251, 2012.

- O. Reyes, C. Morell and S. Ventura. ***ReliefF-ML: an extension of ReliefF algorithm to multi-label learning***. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. LNCS. Springer, vol. 8259, pp. 528-535, 2013.

- O. Reyes, C. Morell, and S. Ventura. ***Feature weighting on multi-label data through quadratic loss minimization***. In Congreso Internacional de Matemática y Computación, COMPUMAT-2013, Habana, Cuba, 2013.

- E. Pérez, O. Reyes and S. Ventura. ***Application of active learning in medical diagnosis***. In IV Encontro Regional de Computacão e Sistemas de Informacão, ENCOSIS-2015, Manaus/Amazonas, Brasil, 2015.

- O. Reyes and S. Ventura. ***Estrategia efectiva para el aprendizaje activo multi-etiqueta***. In XVII Conferencia de la Asociación Española para la Inteligencia Artificial, pp. 835-844, Salamanca, Spain, 2016.