UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Label operation for multi-label learning

Adriano Rivolli da Silva

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Adriano Rivolli da Silva

Label operation for multi-label learning

Thesis submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

USP – São Carlos
January 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

S586l

Silva, Adriano Rivolli da
Label operation for multi-label learning / Adriano Rivolli da Silva; orientador André Carlos Ponce de Leon Ferreira de Carvalho. -- São Carlos, 2020.
207 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2020.

1. Inteligência artificial. 2. Aprendizado de máquina. 3. Classificação multirrótulo. 4. Meta-aprendizado. I. Carvalho, André Carlos Ponce de Leon Ferreira de , orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Adriano Rivolli da Silva

Operação de rótulo para o aprendizado multirrótulo

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

USP – São Carlos
Janeiro de 2020

ACKNOWLEDGEMENTS

I waited so much for this moment! Thank you to my parents, Edinelson and Ilaice that one day showed me the importance of studying. I still remember the times I needed to show you my homework. Now, I can say: It worked! All the love and example I received from you led me until here. This thesis is for you.

To Karina Ap. da Silva Martin who was by my side for many years. You can also be proud of this achievement. If I arrived at this point, it is also because of you: your loyalty, partnership, faith, altruism and integrity. There are no words to say thank you enough, but you know how I am thankful for everything; each second, each breath, each smile, each dream, each rhyme, even each tear. For you Ká, my special thank you.

I also would like to say thank you to my supervisor André, a great professor and mainly person. You always guided me with wisdom, motivation and respect. In hard times, you advised me with the proper words. I hope I can follow your steps and get closer to what you taught me.

Thank you to my 'co-supervisor', Carlos Soares. It was a great honor to work with, such a brilliant researcher. With the same intensity, thank you professor Bernhard Pfahringer who supervised me in New Zealand. Both of you were fundamental for the development of this thesis.

A big thank you to all my family and friends for collaborating and understanding this time in which I was a Ph.D. student. I would especially like to cite some names: Andinei Borba, Carlos N. Silla Jr., Davi Pereira dos Santos, Douglas D. de Castilho Braz, Ekaterina Rodionova, Flávia Rivolli da Silva, Geovana Lourenço de Carvalho, Henrique Yoshikazu Shishido, Kemilly Dearo Garcia, Leonardo Rivolli Pereira, Luís Paulo Faina Garcia, Marina Stuchi, Nourah A. S. Alkhattaf, Thaís Kauana M. Sobral and Victor Hugo Barela. There are many other names I considered to write here, so much so that I am not being completely fair in not doing it. However, I know that my thanks are for these people too.

Last but not least important, my thanks to my friends from the Biocom; to the ICMC community; to my coworkers from UTFPR, that supported me while I was away from work; to my many English-learner partners all over the world that helped me practise English; and, my new friends from Londrina, São Carlos, Porto and New Zealand. Thank you to all of you!

“Pode cantar o que for bonito
Estranho ou esquisito
Pois o que vale aqui é o que vem do coração
Pode cantar pra fugir da lida
Pra esquecer da vida
Quem é que vai dizer o que se deve ou não nessa canção?”
(Flávio Rivolli)

RESUMO

DA SILVA, A. R. **Operação de rótulo para o aprendizado multirrótulo**. 2020. 207 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Tarefas de classificação nas quais instâncias são associadas com múltiplos conceitos são conhecidas como classificação multirrótulo e devido ao alto número de aplicações e dados multirrótulos disponíveis atualmente, é grande o interesse deste assunto pela comunidade de aprendizado de máquina. Consequentemente, têm sido propostas muitas estratégias explorando diferentes particularidades desse tipo de tarefa como o desbalanceamento dos rótulos, redução de dimensionalidade e a dependência dos rótulos. No entanto, alguns aspectos que podem afetar tais estratégias são negligenciados, como as que transformam os dados multirótulos em dados monorótulos e utilizam um algoritmo base para resolver as subtarefas geradas. O impacto de se escolher um algoritmo específico em detrimento de outro é desconhecido e normalmente ignorado, assim como foi observado que muitos rótulos nunca são corretamente preditos, independentemente da estratégia utilizada. Estas questões não têm recebido a devida atenção, mesmo podendo produzir resultados enganosos, portanto, esta pesquisa tem por objetivo investigar as estratégias multirrótulos explorando essas particularidades. Para tanto, um extensivo estudo comparativo foi realizado, cujo foco é analisar a influência do algoritmo base nos resultados. Além disso, a operação de rótulo é proposta como uma estratégia de otimização capaz de reduzir o número de rótulos incorretamente preditos. Foi constatada, por meio de uma metodologia empírica, que as operações de expansão e redução dos rótulos melhoraram diferentes medidas de avaliação e reduziram o problema dos rótulos não preditos, embora não completamente. O meta-aprendizado foi também investigado como forma de reduzir a complexidade das operações e prover algum entendimento sobre as questões estudadas. Com isso, as medidas de caracterização para meta-aprendizado foram sistematicamente investigadas, resultando em uma nova taxonomia para organizá-las. Desse modo, as desc obertas e contribuições apresentadas aqui são relevantes, principalmente, para a área de pesquisa em aprendizado multirrótulo e meta-aprendizado, assim como levantam novas questões relacionadas a aspectos despercebidos de tais áreas. A presente tese também tem potencial impacto na metodologia experimental desse tipo de pesquisa.

Palavras-chave: multirrótulo, operação com rótulos, transformação de problema, meta-aprendizado, meta-características.

ABSTRACT

DA SILVA, A. R. **Label operation for multi-label learning**. 2020. 207 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Classification tasks in which instances are associated with multiple concepts are known as multi-label classification. They have attracted growing attention in the machine-learning community, given the high number of applications and multi-labeled data available nowadays. Consequently, many strategies have been proposed exploring different particularities, such as label imbalance, dimensionality reduction and labels dependence. Despite that, some aspects that may affect strategies as a whole have been overlooked. For instance, some strategies transform the original multi-labeled data into single-labeled data upon which a base algorithm can be applied. However, the impact of choosing a specific base algorithm against another is unknown and usually ignored. Moreover, it was observed that many labels are never correctly predicted regardless of the strategies used. So far, very little attention has been paid to theses issues, which may produce misleading results. Therefore, this thesis aims to investigate the multi-label strategies covering these particularities. For such, an extensive comparative study is performed focusing on the influence of the base algorithms on the results. Moreover, label operation is proposed as an optimization procedure able to reduce the number of labels never predicted. Through an empirical methodology, label expansion and reduction enhanced different evaluation measures, mitigating the label prediction problem, although it was not completely removed. Additionally, meta-learning is used to reduce the complexity of the operations and to provide some understanding concerning the studied issue. Considering this, characterization measures for meta-learning were systematically investigated, which resulted in a new taxonomy to organize them. In summary, the findings and contributions presented here are relevant to the multi-label and meta-learning research fields. They potentially have an impact on the methodology, and raise open new questions concerning unnoticed aspects of these areas.

Keywords: multi-label, label operation, problem transformation, meta-learning, meta-features.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AIC	Akaike Information Criterion
AUC	Area Under ROC Curve
Auto-ML	Automatic Machine Learning
BIC	Bayesian Information Criterion
BR	Binary Relevance
BR+	BR plus
C4.5	Decision Tree Induction Algorithm version 4.5
C5	Decision Tree Induction Algorithm version 5.0
C5.0	Decision Tree Induction Algorithm version 5.0
CART	Classification And Regression Tree
CC	Classifier Chains
CLP	Constant Label Prediction
CRAN	Comprehensive R Archive Network
DBR	Dependent Binary Relevance
DCT	Data Characterization Tool
DT	Decision Tree
EBR	Ensemble of Binary Relevance
ECC	Ensemble of Classifier Chains
F1	F_1 measure
FN	False Negative
FP	False Positive
HL	Hamming-loss
IG	Information Gain
IID	Inner Imbalance Degree
kNN	k-Nearest Neighbors
lCard	Label cardinality
LE	Label Expansion
LP	Label Powerset
LR	Label Ranking
LR	Label Reduction
LR	Logistic Regression

MBR	Meta-BR
MFE	Meta-Features Extractor
ML	Machine learning
MLC	Multi-label Classification
MLP	Missing Label Prediction
MtL	Meta-Learning
NB	Naive Bayes
NFL	No Free Lunch
NS	Nested Stacking
OE	One-error
PCA	Principal Component Analysis
PruDent	Pruned and confiDent
PUL	Proportion of Unique Label sets
RAkEL	RAndom k-labEL sets
RDBR	Recursive Dependent Binary Relevance
REMEDIAL	REsampling MultilabEl datasets by Decoupling highly ImbAlanced Labels
RF	Random Forest
RL	Ranking-loss
SA	Subset-accuracy
SCUT	Score-Cut
SVM	Support Vector Machine
SVMt	Tuned Support Vector Machine
TN	True Negative
TP	True Positive
utiml	Utilities for Multi-label
WLP	Wrong Label Prediction
XB	Xie-Beni index
XGB	eXtreme Gradient Boosting

CONTENTS

CHAPTER

1

INTRODUCTION

Machine learning (ML), which is usually associated with artificial intelligence, can be characterized by systems that are able to learn from previous experience (MITCHELL, 1997). Using an inductive approach, learning algorithms induce predictive models from data, a collection of past facts (instances) described by a set of features. Usually, these models can address tasks that cannot be deterministically solved or are too complex (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012). This thesis is concerned with a particular sub-area of ML, data classification. Classification is a supervised learning task, in which the training instances are labeled with a target value, the "class". In the classification task, this thesis investigated Multi-label Classification (MLC).

MLC classification studies the classification tasks whose instances can have more than one label (CARVALHO; FREITAS, 2009; TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). In MLC, the class of an instance is the set of labels associated with the instance. Initially applied to the text categorization task (LEWIS *et al.*, 1996; JOACHIMS, 1998; SCHAPIRE; SINGER, 2000), MLC has been investigated widely by the ML community (GALINDO; VENTURA, 2014) with applications in distinct domains, such as:

- **Text** related to laws and legal documents (GONÇALVES; QUARESMA, 2003; MENCíA; FÜRNKRANZ, 2008), web pages (UEDA; SAITO, 2002), medical exams and diagnosis (PESTIAN *et al.*, 2007), electronic forums (CHARTE *et al.*, 2015a), news (LEWIS *et al.*, 2004), to mention some. Moreover, applications exploring tag suggestion (KATAKIS; TSOUMAKAS; VLAHAVAS, 2008), text organization (DAYRELL *et al.*, 2012) and sentiment analysis (HUANG *et al.*, 2013; LI *et al.*, 2015) have also been designed as MLC tasks.

- **Multimedia** related to the semantic annotation and object recognition of audio (TROHIDIS *et al.*, 2008; LO *et al.*, 2011; BRIGGS *et al.*, 2013), image (BOUTELL *et al.*, 2004; ANTENREITER; ORTNER; AUER, 2009; DUYGULU *et al.*, 2002) and video

(SNOEK *et al.*, 2006; QI *et al.*, 2007; MARKATOPOULOU; MEZARIS; KOMPATSIARIS, 2014).

- **Biological data** related to gene annotation (CLARE; KING, 2001; ELISSEEFF; WESTON, 2001; TANAKA *et al.*, 2015) and association of protein to functions (DIPLARIS *et al.*, 2005; ZOU, 2016; YANG; LU, 2006; XIAO; WU; CHOU, 2011; DUWAIRI; KASSAWNEH, 2008).

- **Miscellaneous** applications that do not fit in the previous categories. For instance, chemical analysis (KAWAI; TAKAHASHI, 2009), recommendation system (ZHENG; MOBASHER; BURKE, 2014), movement detection based on sensors (READ; ZLIOBAITE; HOLLMÉN, 2016), meta-learning (ZHANG; SONG, 2015; PINTO; SOARES; MENDES-MOREIRA, 2016a) and food truck recommendation (RIVOLLI; PARKER; CARVALHO, 2017), to cite some.

MLC learning strategies are typically organized into two groups (TSOUMAKAS; KATAKIS, 2007): (i) *problem transformation*, which relies on transforming the original data so that single-label algorithms can be applied; (ii) *algorithm adaptation*, which modifies the single-label algorithms to directly support MLC tasks. Thus, while the former modifies the data to fit it to the algorithm, the latter modifies the algorithm to fit it to the data (ZHANG; ZHOU, 2014).

Strategies that transform the problem are also called *algorithm independent*, since any algorithm can be applied to solve the MLC task (CARVALHO; FREITAS, 2009). A "base algorithm", as it is called in this context, is used to induce internal models that are able to predict a label or group of them. The MLC result is obtained by combining the model predictions following some strategy defined by the transformation solution.

The main efforts made by the MLC community comprise the development of new strategies. Hence, the choice of the base algorithm, as well as its actual impact in the MLC result, have been overlooked in the MLC literature. Thus, the investigation of the impact of the base algorithms to the transformation strategies is the starting point of this research.

By performing a broader study concerning the base algorithms, it was observed that some labels have never been correctly predicted. Somewhat surprisingly, this problem was not previously observed, despite being recurrently observed for all strategies and base algorithms investigated in this thesis. Consequently, this is the first attempt to address the matter in the MLC literature.

The proposed alternative to deal with this issue is called the label operation. The label operation modifies the instances related to some labels during the transformation process. Different operations are formalized and explored in order to enhance a given evaluation measure and mitigate the label prediction problem. Moreover, top of the shelf alternatives (BOUTELL *et al.*,

2004; MONTAÑÉS *et al.*, 2014; CHARTE *et al.*, 2015b; READ *et al.*, 2011; TSOUMAKAS; KATAKIS; VLAHAVAS, 2011; YANG, 2001) were considered and examined in comparison to the proposed alternative.

Finally, Meta-Learning (MtL) is also investigated. In a nutshell, MtL investigates the application of ML to enhance another ML task (BRAZDIL *et al.*, 2009; VILALTA; DRISSI, 2002b; SMITH-MILES, 2008). It is justified because, according to the No Free Lunch (NFL) theorem (WOLPERT; MACREADY, 1995), there is no learning algorithm that can outperform all other algorithms for all problems. As each learning algorithm has a particular inductive bias, which reflects in the performance of the predictive model generated, different algorithms can be suitable for different tasks. By combining the characterization of many tasks along with the respective performance of the algorithms, it is possible to create a recommendation system using ML that is able to suggest a likely good option for a new task (BRAZDIL *et al.*, 2009).

The development of MtL in this thesis is focused on organizing and understanding the characterization measures, commonly called *meta-features* (CASTIELLO; CASTELLANO; FANELLI, 2005). Using them to explain and enhance the label operation and the label prediction problem occurrence is the main reason for this exploration in this study.

In summary, the research questions that incrementally arose to guide the development of this work are:

1. Q1: What is the impact of the base algorithm on the transformation strategies?
2. Q2: How to overall improve the labels' predictions in order to mitigate the problem in which some of the labels are never correctly predicted?
3. Q3: How to find the right labels to be combined in the label operation procedure?

1.1 Hypothesis

Taking into account the current state of the art in MLC and MtL literature and the research questions previously presented, the hypotheses investigated in this thesis are:

1. ***The base algorithm has a stronger influence than the transformation strategy on the predictive performance of the MLC models***. If the choice of a base algorithm is more important regarding the quality of the results than a specific strategy, then it must be carefully selected. In empirical studies involving transformation strategies, multiple base algorithms should be considered, which is not currently observed in the MLC literature (READ *et al.*, 2011; MONTAÑÉS *et al.*, 2014; MADJAROV *et al.*, 2012).
2. ***The right combination of labels during the transformation process leads to an improvement in the MLC performance and this can mitigate the label prediction problem***. Con-

sidering that labels may have some relationship with other labels, combining them can take some advantage of it. On the other hand, during the inductive process, a label may seem such as noise for another close-related label, which is not well supported for many learning algorithms (GARCIA; LORENA; CARVALHO, 2012). Furthermore, by combining labels in different ways, it is possible to explore the dependency between them and obtain more balanced datasets in some cases.

3. ***MtL can reduce the complexity of label operation and improve their predictive performance***. Different alternatives can be explored concerning the use of MtL to complement the label operation: from generic scenarios like the suggestion of the operation type; to more specific scenarios such as finding the right pair of labels to be operated. Additionally, some predictive models can be interpreted. This can bring awareness concerning the behavior of the operations and the occurrence of the label prediction problem.

1.2 Objectives

The main aim of this study is to investigate the MLC strategies focusing on the predictive performance of individual labels. This includes the understanding of the behavior of the base algorithms and transformation strategies. A second goal is concerning a more in-depth investigation of meta-features for MtL, given that they play an important role in MtL applications (BENSUSAN; KALOUSIS, 2001; BILALLI; ABELLÓ; ALUJA-BANET, 2017). Finally, the support provided by MLC strategy recommendation systems based on MtL are explored.

1.3 Outline

This thesis is organized as a collection of the main papers written by the candidate as part of his Ph.D. project. Each chapter is a self-contained paper submitted to a relevant scientific journal in the ML area. Although they can be read in any order, they are presented in a logical sequence, representing the progress of the research. For the sake of convenience, the reference and appendix sections of each chapter were mixed in unique sections in the thesis. Moreover, the reader may notice that some sections from different chapters (mainly relative to the literature review) are very similar to each other because the content of the paper is fully transcribed in the thesis. It is important to observe that the formalism and notation between chapters may also present some differences, given that the papers address distinct fields and audiences of ML (MLC and MtL).

The remainder of this thesis is organized as follows:

Chapter 2 - *An empirical analysis of binary transformation strategies for multi-label learning* investigates the impact of the base algorithms over the binary transformation strategies. It comprises one of the largest empirical studies in the MLC literature, evaluating 6 distinct base

algorithms and 10 transformation strategies. Among the experimental results, the wrong label prediction problem is formalized and identified as a recurrent problem in MLC. This chapter is directly related to hypothesis 1.

Chapter 3 - *Characterizing classification datasets: a study of meta-features for meta-learning* is a survey of meta-features for MtL. A new taxonomy to organize the characterization measures is proposed and an extensive list of them are properly organized in the respective taxonomy. Some open issues concerning the meta-features are identified and discussed in detail. A new tool to characterize datasets is also proposed.

Chapter 4 - *Label operations for multi-label optimization* introduces the label operation as an optimization procedure that is able to enhance an evaluation performance measure and mitigate the previously mentioned label prediction problem. Two operations, expansion and reduction, are compared to the use of threshold optimization. This chapter is directly related to hypothesis 2.

Chapter 5 - *Recommending label operations for multi-label classification* combines MLC and MtL, in which the latter is used to reduce the complexity and enhance the label operation procedure. Besides, MtL is used to explain the label prediction problem and label operation combinations. This chapter is directly related to hypothesis 3.

Finally, Chapter 6 summarizes the contributions to the field, points out some limitations of the research and suggests directions for future work.

CHAPTER

2

AN EMPIRICAL ANALYSIS OF BR-BASED STRATEGIES AND BASE ALGORITHMS FOR MULTI-LABEL LEARNING

Collaborating authors

Jesse Read
Laboratoire d'Informatique (LIX) - École Polytechnique, Palaiseau, France

Carlos Soares
Fraunhofer AICOS and LIAAD-INESC TEC, University of Porto, Porto, Portugal

Bernhard Pfahringer
University of Waikato, Hamilton, New Zealand

André C. P. L. F. de Carvalho
University of São Paulo, São Carlos, Brazil

Abstract

Investigating strategies that are able to efficiently deal with multi-label classification tasks is a current research topic in machine learning. Many methods have been proposed, making the selection of the most suitable strategy a challenging issue. From this premise, this paper presents an extensive empirical analysis of the binary transformation strategies and base algorithms for multi-label learning. This subset of strategies uses the one-versus-all approach to transform the original data, generating one binary data set per label, upon which any binary base algorithm can be applied. Considering that the influence of the base algorithm on the predictive performance obtained by the strategies has not been considered in depth by many

empirical studies, we investigated the influence of distinct base algorithms on the performance of several strategies. Thus, this study covers a family of multi-label strategies using a diversified range of base algorithms, exploring their relationship over different perspectives. This finding has significant implications concerning the methodology of evaluation adopted in multi-label experiments containing binary transformation strategies, given that multiple base algorithms should be considered. Despite these improvements in strategy and base algorithms, for many data sets, a large number of labels, mainly those less frequent, were either never predicted, or always misclassified. We conclude the experimental analysis by recommending strategies and base algorithms in accordance with different performance criteria.

2.1 Introduction

Multi-label learning has been investigated widely by the machine learning community in recent years (CARVALHO; FREITAS, 2009; TSOUMAKAS; KATAKIS; VLAHAVAS, 2010; GALINDO; VENTURA, 2014). It deals with classification tasks where an instance can be simultaneously classified into more than one class. Each class is represented by one label. Several domains, such as text (KLIMT; YANG, 2004; PESTIAN *et al.*, 2007), multimedia (DUYGULU *et al.*, 2002; ZHOU; ZHANG, 2006; BRIGGS *et al.*, 2013) and biology (ELISSEEFF; WESTON, 2001), are intrinsically multi-label.

A common approach to dealing with multi-label classification tasks is to transform the original data set into one or more single-label data sets. A conventional binary classification algorithm, called *base algorithm* here, is used to induce predictive models for each one of them. As such, a transformation strategy defines how to decompose the original task into a set of single-label tasks and to combine the results obtained from these tasks to solve the original task (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). Many strategies have been proposed to address the multi-label tasks and transform the data, exploring different aspects, such as label correlation (READ *et al.*, 2011; CHERMAN; METZ; MONARD, 2012; MONTAÑÉS *et al.*, 2014), dimensionality reduction (TSOUMAKAS; KATAKIS; VLAHAVAS, 2008; ZHANG; WU, 2015) and class imbalance (ZHANG; WU, 2015; TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

Although the base algorithm can be seen as a hyperparameter for transformation strategies, it is generally fixed for all strategies, so that only a single base algorithm is considered in the whole experiment (READ *et al.*, 2011; MONTAÑÉS *et al.*, 2014; MADJAROV *et al.*, 2012). Given that a comprehensive comparison of the binary transformation strategies, using different base algorithms, has not yet been performed, this study assesses the hypothesis that the base algorithms can have a stronger influence than the binary transformation strategies on the predictive performance of multi-label models. At a glance, it may seem trivial to be investigated, however, if the choice of a base algorithm is more important regarding the quality

of the results than the specific strategy, then several of them should be considered in empirical studies evaluating these strategies.

In the multi-label literature, the most similar comparative study was performed by Madjarov *et al.* (2012), where 12 strategies (including 3 binary transformation strategies) were evaluated under several measures, using the original train and test partition of 11 benchmark data sets. Even though a variety of different algorithms were considered, the transformation strategies were evaluated with a single base algorithm, Support Vector Machine (SVM). Another large empirical study covering multiple ensemble strategies (MOYANO *et al.*, 2018) used only the C4.5 decision tree as the base algorithm. Nevertheless, a few studies have considered using more than one base algorithm. These studies include Tsoumakas and Katakis (2007) and Cherman, Metz and Monard (2012), who did not compare strategies using different base algorithms; and Zufferey *et al.* (2015), who compared strategies with distinct base algorithms, but just in a single data set.

Methods using Automatic Machine Learning (Auto-ML) to address multi-label classification tasks also consider multiple base algorithms (SÁ; PAPPA; FREITAS, 2017; SÁ; FREITAS; PAPPA, 2018; WEVER; MOHR; HÜLLERMEIER, 2018; WEVER *et al.*, 2019). During the search for a solution, the Auto-ML method may find a suitable combination between strategies and base algorithms that optimizes a fitness function. In these cases, choosing the base algorithm is seen as part of the solution and the comparison of the strategies does not fix a base algorithm, as observed in other studies.

Since the most common strategies are based on binary transformations, this paper will focus on these strategies. Hence, 10 binary transformation strategies and 5 different base algorithms (plus one with its hyperparameters tuned) were evaluated using 5x2-fold cross-validation for 20 benchmark data sets. In contrast to previous studies, which used null hypothesis significance testing, we ran Bayesian statistic tests (BENAVOLI *et al.*, 2017) to assess the statistical significance of the differences in the predictive performance of the assessed strategies over different evaluation measures. To the best of our knowledge, this is the most extensive multi-label empirical study carried out so far.

The results reported reinforce the claim that the predictive performance obtained by transformation strategies is affected by the base algorithm used. Thus, experimental studies in multi-label learning must take into account experiments with several different base algorithms. In particular, many of the binary transformation strategies obtained very similar results, with differences mainly being due to the choice of the base algorithm used. Therefore, previous comparative studies (MADJAROV *et al.*, 2012; MOYANO *et al.*, 2018) might have reached different conclusions if other base algorithms had been employed. Additionally, for many data sets, the investigated strategies consistently predicted only a subset of the existing labels, never assigning the remaining labels to any instance. This problem was previously observed in the food truck data set (RIVOLLI; SOARES; CARVALHO, 2018a), however, as far as we know, it

has never been widely investigated.

The rest of the paper is organized as follows: Section 2.2 formally defines the main concepts relevant for multi-label learning. Section 2.3 details the investigated strategies. Section 2.4 describes the experimental design, including data sets, evaluation procedures, base classifiers, tools, and hyperparameter values adopted. Section 2.5 presents, analyzes and discusses the empirical results. In the last section, conclusions are drawn concerning relevant findings from the experimental study and future work directions.

2.2 Multi-label learning

In multi-label learning, an instance can be simultaneously associated with more than one label. The main tasks in this field are *Multi-Label Classification* and *Label Ranking*.

Multi-Label Classification (MLC), the most common task (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010), induces a predictive model $h : \mathscr{X} \rightarrow \mathscr{Y}$ from a set of training data $\mathscr{D}$, which later assigns labels to new examples. This task can be formally defined as follows. Let $\mathscr{D}$ be a set of labeled instances, such that $\mathscr{D} = \{(x_1, Y_1), ..., (x_n, Y_n)\}$. Every labeled instance is composed of $x_i = (x_{i1}, x_{i2}, ..., x_{id}) \in \mathbb{R}^d$, and $Y_i \subseteq \mathscr{L}$, such that $\mathscr{L} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ is the set of all q labels λ_i. For the sake of convenience, the labels associated with the i^{th} instance, also called label set, can be seen as a binary vector $y_i = (y_{i1}, y_{i2}, \cdots, y_{iq}) \in \{0,1\}^q$, where $y_{ij} = 1$ *iff* $\lambda_j \in Y_i$ and $y_{ij} = 0$ *iff* $\lambda_j \notin Y_i$. Finally, model h is used to predict, for a test instance $(x_i, ?)$, the set of relevant labels $\hat{Y}_i$ (or $\hat{y}_i$ as a binarized prediction).

In the *Label Ranking* (LR) task, a model outputs the ranked labels for each test instance. This ranking can easily be computed using any model that provides a score value indicating its probability of being relevant to a given instance. Thus, the higher the score value, the better its ranking position. In turn, MLC can be derived from the LR formulation (GIBAJA; VENTURA, 2015).

A multi-label model can be obtained by using two approaches (TSOUMAKAS; KATAKIS, 2007), *problem transformation* and *algorithm adaptation*. The former converts the original multi-labeled data into a set of binary or multi-class data sets, whereas for the latter, the multi-label support is embedded into the algorithm's structure. Thus, the transformation approach fits the data to the algorithms, and the adaptation approach fits the algorithms to the data (ZHANG; ZHOU, 2014).

A straightforward transformation is to build a binary classifier for each label individually. This is known as the *Binary* approach. On the other hand, a multi-class transformation can be considered, in which each label set (combination of labels) is mapped to one class. Both approaches are *algorithm independent* (CARVALHO; FREITAS, 2009), in the sense that any traditional classification algorithm that is capable of handling such problems can be used as the

base algorithm.

We want to emphasize that the binary transformation approach implies that algorithms are trained separately, but not necessarily independently; this will become apparent in the following section. In addition, many hybrid approaches exist, such as *Pairwise*, which models pairwise combinations (a one-vs-one approach), and subset approaches, which includes the well-known RAndom k-labEL sets (RAkEL) strategy (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

Binary transformation generates at least one data set per label. Each binary data set $\mathscr{D}'_j$ is related to the label λ_j. The instances associated with λ_j are labeled with a class value of "1", all others are labelled with a class value of "0".The number of binary data sets generated is defined by $|\mathscr{D}'| = mq$, where m is the number of data sets per label. Therefore, the complexity of this family of strategies is linear in the number of labels q. Negative aspects of this approach include the tendency to generate rather imbalanced data sets and the fact that some of these strategies ignore the relationships between labels (ZHOU; TAO; WU, 2012).

The binary transformation strategies are organized into three groups, *one-round*, *stacking*, and *ensemble*, according to the value of m. *One-round* strategies are the simplest strategies, with $m = 1$. A special case of one-round is *chaining*, which increases the input space by adding already predicted labels as features to predict the others, in a chain. In *stacking* strategies, two rounds of training and prediction steps are performed, thus $m = 2$. They augment the input space in the second round by using the values of the labels predicted in the first round as features. When all the labels are used, they are called *full-stacking*. When only a subset of the labels is used, they are called *pruned-stacking*. Finally, in the *ensemble* strategies more than two models for each label ($m > 2$) are used and usually, the value of m is a hyperparameter defined by the user. When the same instances and attributes are shared by all internal models, the ensemble is *homogeneous*. However, when each member and label use distinct data sets as training data, the ensemble is *heterogeneous*. The former can be seen as an ensemble of multi-labeled data, whereas the latter as multiple ensembles of single-label data (GIBAJA; VENTURA, 2015). These groups and their strategies are detailed in Section 2.3.

A base classification algorithm must always be chosen to induce predictive models for each transformed data set $\mathscr{D}'$. Later, these models are used to predict the relevance of each label for new instances. If the models predict a score instead of a class, the strategies support both tasks, MLC and LR (GIBAJA; VENTURA, 2015). Logically, if the base algorithms are responsible for predicting a score and the binary transformation strategies are independent from them, any transformation strategy can be used to solve them. Distinctions among them will not be considered in the rest of this paper.

As previously mentioned, this study is restricted to analyzing strategies based on binary transformation, which are relevant for a broad group of researchers and practitioners. Besides, for most of them, their individual models can be trained separately (thus, allowing for parallelism), they are simple to interpret, they have been successfully used in many state-of-the-art comparisons

in the literature, and they usually exhibit acceptable time complexity, almost linear with the number of labels. Using separate classifiers, each focused on only one label, allows for higher flexibility, choosing potentially different approaches on a per-label basis. Furthermore, new labels can usually be added to the problem without retraining the models built for existing labels. In general, as some of the strategies are conceptually quite similar to each other, their practical differences may be highlighted by comparing their performances using different base algorithms, an approach we put forward in this paper.

2.3 Strategies

In this section, the 10 binary transformation strategies considered are described. Table 1 presents the strategies organized into groups, defined by the number of binary models generated per label, and the subgroups according to their main characteristic.

Table 1 – Binary transformation strategies organized into groups/subgroups according to the number of binary models per label and their main characteristic.

Group	Subgroup	Strategy	Reference
One-round	-	BR	Boutell *et al.* (2004)
	Chaining	CC	Read *et al.* (2011)
		NS	Senge, Coz and Hüllermeier (2013)
Stacking	Full	BR+	Cherman, Metz and Monard (2012)
		DBR	Montañés *et al.* (2014)
		RDBR	Rauber *et al.* (2014)
	Pruned	MBR	Godbole and Sarawagi (2004)
		PruDent	Alali and Kubat (2015)
Ensemble	Homogeneous	EBR	Read *et al.* (2011)
		ECC	Read *et al.* (2011)

2.3.1 One-round

The *one-round* strategies are characterized by generating only a single binary data set for each label. Binary models are induced from these data sets and used for multi-label prediction. The strategies from this group differ mainly by how they transform the data sets.

Binary Relevance (BR) (BOUTELL *et al.*, 2004) is the simplest and most popular multi-label strategy (LUACES *et al.*, 2012; MONTAÑÉS *et al.*, 2014). For each label λ_j, an independent binary data set is generated according to

$$\mathscr{D}'_j = \left\{ (x_i, y_{ij}) \mid 1 \leq i \leq n \right\}, \tag{2.1}$$

and will be used to induce a binary model θ_j. The prediction is performed using the values of all binary models as follows:

$$h_{br} = \{ \lambda_j \mid \theta_j(x) = 1,\ 1 \leq j \leq q \}. \tag{2.2}$$

2.3.1.1 Chaining

The *Classifier Chains* (CC) strategy (READ *et al.*, 2009; READ *et al.*, 2011) organizes the labels in a chain and increases the original input space of the transformed data set for a given label with the values of all previous labels in the chain. Thus, the data set is transformed as follows:

$$\mathscr{D}'_j = \left\{ ([x_i, y_{i1}, y_{i2}, \cdots, y_{i(j-2)}, y_{i(j-1)}], y_{ij}) \mid 1 \leq i \leq n \right\}. \tag{2.3}$$

The model related to the first label in the chain is obtained exclusively from the original input data, without adding any predictive attributes, as shown in Equation 2.1. The other models increase their input space by adding $j-1$ new attributes, where j is the position of the respective label in the chain. During the prediction phase, as the labels are predicted, their values are used to increase the input space, as shown next

$$\begin{aligned} h_{cc} &= \{\lambda_j \mid \hat{y}_j = 1,\ 1 \leq j \leq q\}, \text{where} \\ \hat{y}_j &= \theta_j([x, \hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{(j-2)}, \hat{y}_{(j-1)}]). \end{aligned} \tag{2.4}$$

Nested Stacking (NS) (SENGE; COZ; HÜLLERMEIER, 2013) brings two modifications to CC. In the training phase, it uses the predicted labels instead of the real labels. Furthermore, in the prediction phase, it makes a subset correction, in order to predict only preexisting label sets.

The transformation step is similar to Equation 2.3. However, the original label values y are changed by the predicted values $\hat{y}$, such that

$$\mathscr{D}'_j = \left\{ ([x_i, \hat{y}_{i1}, \hat{y}_{i2}, \cdots, \hat{y}_{i(j-2)}, \hat{y}_{i(j-1)}], y_{ij}) \mid 1 \leq i \leq n \right\},$$

where $\hat{y}_{ij}$ is the prediction of the binary model θ_j for the instance x_i presented in the training data. The prediction is obtained similarly to Equation 2.4 followed by the subset correction. The $\hat{y}$ is replaced by $y^* \in Y$, which is the vector in Y that is most similar to $\hat{y}$, such that

$$\begin{aligned} h_{ns} &= \{\lambda_j \mid y^*_j = 1,\ 1 \leq j \leq q\}, \text{where} \\ y^* &= \underset{y \in Y}{\arg\min}\ dist(\hat{y}, y), \end{aligned}$$

and *dist* is the hamming distance, which corresponds to the number of differences between two binary vectors. When more than one minimum is found, the label set with the higher frequency in the training data is selected.

2.3.2 Stacking

The stacking strategies are characterized by using the stacked generalization learning paradigm (WOLPERT, 1992). In the multi-label context, they use two rounds of binary transformation, where in the second round, the input space is augmented by the information from

the labels obtained from the first round.[1] The main difference among the stacked strategies is how they choose the labels that would augment the input space. Some of them use all labels (full stacking), while others use only a subset of labels (pruned stacking).

2.3.2.1 Full stacking

BR plus (BR+) (CHERMAN; METZ; MONARD, 2012) and *Dependent Binary Relevance* (DBR) (MONTAÑÉS *et al.*, 2014) are very similar to each other. In the training phase, they perform exactly the same procedure. The first round is characterized by the induction of a BR model, according to Equations 2.1 and 2.2. In the second round, the transformation is performed by increasing the input space using the original labels. To illustrate how it works, let $\phi_j(y)$ be a function that removes the label λ_j from the vector y, such that

$$\begin{aligned} \mathscr{D}_j^{''} &= \left\{([x_i, \phi_j(y_i)], y_{ij}) \mid 1 \leq i \leq n\right\}, \text{ where} \\ \phi_j(y) &= (y_1, \cdots, y_{(j-1)}, y_{(j+1)}, \cdots, y_q). \end{aligned} \tag{2.5}$$

It should be noted though, that there is a subtle difference in the prediction phase, precisely, in the second round. DBR predicts the labels using the second round binary models that use the labels obtained from the first round binary models. Using the ϕ function presented in Equation 2.5, the prediction is obtained as follows:

$$h_{dbr} = \{\lambda_j \mid \theta_j^{''}([x, \phi_j(h_{br}(x))]) = 1,\ 1 \leq j \leq q\}.$$

Differently, BR+ updates the labels from the first round binary models while the second prediction is occurring. Given a chain of labels (for example, $\lambda_1 \prec \lambda_2 \prec \cdots \prec \lambda_q$), the prediction is obtained in the following way:

$$\begin{aligned} h_{br+} &= \{\lambda_j \mid \theta_j^{''}([x, \phi_j(\hat{y})]) = 1,\ 1 \leq j \leq q\}, \\ \text{for each } j, \quad \hat{y} &= (\hat{y}_1, \cdots, \hat{y}_{(j-1)}, \theta_j^{''}([x, \hat{y}])), \hat{y}_{(j+1)}, \cdots, \hat{y}_q). \end{aligned} \tag{2.6}$$

Recursive Dependent Binary Relevance (RDBR) (RAUBER *et al.*, 2014) induces two models as DBR does, but it uses the second model several times in a recursive way. The labels predicted for the second model are used to update the input space and the second round is executed again until either the result converges or a fixed number of iterations is reached. In practice, it is the same process as in Equation 2.6, but while BR+ does only one update, RDBR updates recursively several times until a stopping criterion is reached.

2.3.2.2 Pruned stacking

The *Meta-BR* (MBR) strategy[2] (GODBOLE; SARAWAGI, 2004; READ *et al.*, 2011) augments the input space using the values of the most correlated labels (TSOUMAKAS *et*

[1] Although CC and NS also augment the input space, they are not considered stacking, given that only one-round is performed.

[2] Also known as 2BR (TSOUMAKAS *et al.*, 2009), *Meta-Stacking* (READ *et al.*, 2009) and *Stacking* (MONTAÑÉS *et al.*, 2014).

al., 2009). The Pearson product moment correlation coefficient for categorical variables ρ is computed for each pair of labels and a threshold τ is used to define which labels should augment the space of attributes. The data set in the second round is obtained in the following way:

$$\mathscr{D}_j^{''} = \left\{([x_i, \phi_j(\hat{y}_i)], y_{ij}) \mid 1 \leq i \leq n\right\}, \text{ where}$$
$$\phi_j(\hat{y}) = \{\hat{y}_l \mid \rho(\lambda_j, \lambda_l) \geq \tau,\ 1 \leq l \leq q\},$$

and $\phi(\hat{y}_i)$ returns only the most related labels. Unlike the other stacked strategies, instead of using the original labels in the second transformation, it uses the predicted labels obtained in the first round.

The final prediction is the result of the binary models in the second step, such that:

$$h_{mbr} = \{\lambda_j \mid \theta_j^{''}([x, \phi_j(h_{br}(x))]) = 1,\ 1 \leq j \leq q\}.$$

The *Pruned and confiDent* (PruDent) strategy (ALALI; KUBAT, 2015) uses only the most relevant labels, as MBR does, and the original values to augment the second input space, as BR+ and DBR do. The Information Gain (IG) measure is used to prune the irrelevant labels based on a threshold τ. The PruDent transformation is the same as Equation 2.5, with the exception of the ϕ function:

$$\phi_j(y) = \{y_l \mid IG(\lambda_j, \lambda_l) \geq \tau,\ 1 \leq l \leq q,\ l \neq j\}.$$

Contrary to the others, PruDent assigns a label to an example if either one of the corresponding models, first or second round, predicts it. The predictions are done in the following way:

$$h_{prud} = \{\lambda_j \mid \theta_j(x) = 1 \ \vee\ \theta_j^{''}([x, \phi_j(h_{br}(x))]) = 1,\ 1 \leq j \leq q\}.$$

2.3.3 Ensemble

Ensemble of Binary Relevance (EBR) and *Ensemble of Classifier Chains* (ECC) (READ *et al.*, 2011) are simply ensembles of models induced by the BR strategy and by the CC strategy, respectively. Both BR and CC use bagging and choose different random subsets of the attributes for each bagging iteration. To illustrate how EBR computes predictions, let m be the number of models in the ensemble and ϕ_i a function for selecting a random subset of attributes:

$$h_{ebr} = \{\lambda_j \mid \left(\frac{1}{m}\sum_{l=1}^{m} \hat{y}_{lj}\right) > \tau,\ 1 \leq j \leq q\}, \text{ where}$$
$$\hat{y}_l = h_{br}^l(\phi_l(x)),$$

$\hat{y}_{lj}$ is the predicted value of the BR model l for the label λ_j and τ is a threshold value.[3] For the ECC strategy, internal models are built using h_{cc} with different chains, avoiding the influence that choosing an inappropriate chain could have on the results.

2.4 Experimental design

This section presents an experimental comparison across the binary transformation strategies and base algorithms. It describes the multi-label data sets, followed by a short overview of evaluation measures and procedures. Next, it explains the methodology adopted and the environmental setup.

2.4.1 Data sets

Table 2 lists the 20 multi-label data sets used for the experiments. They are from distinct domains (column *Domain*) and have a wide diversity in their characteristics. The columns *Inst*, *Attr* and *Lbl* are respectively the number of instances, attributes and labels. Label sets (*lSets*) is the amount of distinct label combination, Proportion of Unique Label sets (*PUL*) indicates the proportion of label sets related to a single instance, label cardinality (*lCard*) measures the average number of labels per instance, label density (*lDen*) describes the average frequency of labels, dependency (*Dep*) shows the average unconditional labels' dependency (LUACES *et al.*, 2012), Inner Imbalance Degree (*IID*) measures the average label imbalance in the binary data sets (MONTEJO-RÁEZ; LÓPEZ; STEINBERGER, 2004) and, finally, correlation (*Corr*) indicates the average correlation between the predictive attributes and the labels.

Letting ρ_{jk} be the Pearson correlation coefficient between the j^{th} attribute and the label λ_k, the correlation is computed as

$$Corr = \frac{1}{q}\sum_{k=1}^{q} max(|\rho_{1k}|, |\rho_{2k}|, ..., |\rho_{dk}|),$$

where d is the number of attributes. A high value for this measure means that there is at least one attribute which is strongly correlated to each label, while a low value indicates the opposite.

These data sets are frequently used as benchmarks for multi-label experiments. They come from different domains, organized here as text, image, audio, biology and other. The text-domain data sets are related to aviation safety reports (`tmc2007-500`, (SRIVASTAVA; ZANE-ULMAN, 2005)), medical documents (`medical`, (PESTIAN *et al.*, 2007)), emails (`enron`, (KLIMT; YANG, 2004)), newsgroups (`20ng`, (LANG, 1995)), scientific literature (`fapesp`, (CHERMAN *et al.*, 2015); `ohsumed`, (JOACHIMS, 1998)), web forums (`stackex_chess`, (CHARTE *et al.*, 2015a)), and web content (`langlog` and `slashdot`, (READ *et al.*, 2011)). Text

[3] It can either be a predefined value, such as 0.5 (READ *et al.*, 2011) or dynamically defined using the cardinality value of the training data set (READ *et al.*, 2009).

Table 2 – Characteristics of the multi-label data sets.

Data set	Domain	Inst	Attr	Lbl	lSets	PUL	lCard	lDen	Dep	IID	Corr
20ng	text	19300	1006	20	55	0.31	1.03	0.05	0.08	0.9	0.45
birds	audio	337	260	15	115	0.53	1.84	0.12	0.08	0.75	0.39
cal500	audio	502	68	141	502	1.00	25.54	0.18	0.14	0.67	0.15
corel5k	image	4995	499	218	2940	0.76	3.37	0.02	0.16	0.97	0.12
emotions	audio	593	72	6	27	0.15	1.87	0.31	0.28	0.38	0.41
enron	text	1702	1001	42	722	0.74	3.34	0.08	0.12	0.84	0.22
fapesp	text	251	7286	18	61	0.46	1.35	0.08	0.11	0.85	0.57
flags	other	194	19	7	54	0.44	3.39	0.48	0.15	0.35	0.40
image	image	2000	294	5	20	0.10	1.24	0.25	0.15	0.51	0.33
langlog	text	1197	916	38	223	0.53	1.31	0.03	0.06	0.93	0.29
mediamill	image	42177	120	101	6554	0.63	4.56	0.05	0.22	0.93	0.10
medical	text	949	1421	20	55	0.22	1.20	0.06	0.19	0.88	0.76
msd-195	audio	2901	180	38	267	0.09	2.47	0.07	0.24	0.87	0.13
ohsumed	text	13929	1002	23	1147	0.50	1.66	0.07	0.04	0.86	0.32
scene	image	2407	294	6	15	0.20	1.07	0.18	0.11	0.64	0.43
slashdot	text	3776	1079	18	149	0.35	1.18	0.07	0.05	0.87	0.34
stackex-chess	text	1612	585	78	725	0.72	2.07	0.03	0.10	0.95	0.37
tmc2007-500	text	28596	500	22	1172	0.35	2.22	0.10	0.11	0.81	0.38
yeast	biology	2417	103	14	198	0.39	4.24	0.30	0.25	0.54	0.18
yelp8	image	10784	668	8	117	0.06	2.26	0.28	0.11	0.48	0.23

data sets have a higher number of attributes than most of the data sets from the other domains and also contain the largest average value of correlation between attributes and labels.

The image-domain data sets are related to food (yelp), images extracted from videos (mediamill, (SNOEK *et al.*, 2006)), scene classification (image, (ZHOU; ZHANG, 2006); scene, (BOUTELL *et al.*, 2004)), and vector graphics (corel5k, (DUYGULU *et al.*, 2002)). They have the highest average number of labels and label sets of all domains. The data sets with the highest average dependency degree among the labels are from the audio domain. They are related to detecting emotions in songs (emotion, (TROHIDIS *et al.*, 2011)), the identification of music styles (msd-195, (BERNARDINI; BENITO; MEZA, 2014)), music effects classification (cal500, (TURNBULL *et al.*, 2008)) and sounds of birds (birds, (BRIGGS *et al.*, 2013)).

The last two data sets are yeast (ELISSEEFF; WESTON, 2001), a data set from the biology domain that associates gene expressions with biological functions, and flags (GONÇALVES; PLASTINO; FREITAS, 2013), a data set of the countries where the color of their respective flags are the labels.

The data sets come from the Cometa repository (CHARTE *et al.*, 2018), an exhaustive collection of MLC datasets, integrated with the tools used in this work. The exceptions are the data sets fapesp and msd-195 obtained from their respective authors, and yelp8 from the Kaggle website.[4] The data sets were preprocessed with three operations. First, the labels with less than 10 instances were removed to ensure a minimum number of instances with each label in the training and test folds. Next, instances with no labels were also removed. Finally, predictive

[4] see <https://www.kaggle.com/c/yelp-restaurant-photo-classification>.

attributes with constant values were removed.

Concerning the characteristics shown in Table 2, the density (LDen) and the inner imbalance degree (IID) are inversely correlated. As the density increases, the imbalance degree decreases, and vice-versa. We did not find high correlation among the other characteristics.

2.4.2 Evaluation measures

The evaluation of the predictive performance of multi-label strategies requires using different measures to assess different dimensions (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). They are organized here in example-based, label-based and ranking measures. The example-based measures summarize the predictive performance over all instances, whereas the label-based measures summarize the performance over all labels. The ranking measures are a specialization of the former, using the prediction scores instead of the crisp values. As many evaluation measures are highly correlated with each other (PEREIRA *et al.*, 2018), a subset was used.

2.4.2.1 Example-based measures

Hamming-loss (HL) is an error measure that evaluates the misclassification rate for each label of every instance (SCHAPIRE; SINGER, 1999). This measure does not distinguish between false positive and false negative errors, giving the same weight for both, as shown next

$$\begin{aligned} HL = & \frac{1}{n}\sum_{i=1}^{n}\frac{1}{q}|h(x_i)\ \Delta\ Y_i|, \text{ where} \\ & A\ \Delta\ B = (A-B)\cup(B-A). \end{aligned} \tag{2.7}$$

While *Hamming-loss* is the most relaxed measure, *Subset-accuracy* (SA) is the strictest (GIBAJA; VENTURA, 2015). It accounts only for correctly predicted label sets, ignoring the partial hits. A partially correct prediction is valued the same was as a completely incorrect one, such that the set of predicted or observed labels is treated as a class value in single-label classification (ZHANG; ZHOU, 2014). It is computed as

$$\begin{aligned} SA = & \frac{1}{n}\sum_{i=1}^{n} I(h(x_i) = Y_i), \text{ where} \\ & I(\cdot) = \begin{cases} 1 & \text{if the predicate is true,} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.8}$$

Let us call the labels associated with an instance of relevant labels. We can use them to define the following measures: Precision is the fraction of relevant labels among those predicted. A high precision indicates a high ability of a model to correctly predict the labels, although not necessarily all of them. Recall is the fraction of relevant labels that have been predicted out of

all relevant labels. A high recall indicates that a model predicts many labels correctly, but not necessarily only the relevant labels. Thus, the F_1 *measure* (F1) computes the harmonic mean between precision and recall. A model with a high value in this measure can predict the relevant labels accurately and only them. It does not take the true negatives into account, combining just the rate of relevant labels among the predicted ones and the rate of predicted relevant labels over all relevant labels. F1 is computed as

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|h(x_i)\cap Y_i|}{|h(x_i)|+|Y_i|}. \tag{2.9}$$

2.4.2.2 Label-based measures

Label-based measures usually come in two variants: micro-averaged and macro-averaged. The macro-averaged measures summarize the label distribution by giving the same weight to all labels (YANG, 1999). They assess the consistency across all labels. Thus, they are too sensitive to the performance on the least common labels, which is usually low (JACKSON; MOULINIER, 2002).

To illustrate how these measures work, let TP, FP, TN and FN be respectively the true positive, false positive, true negative and false negative values from a confusion matrix, such that

$$Precision_b = \frac{TP}{TP+FP}, \tag{2.10}$$

$$Recall_b = \frac{TP}{TP+FN}, \tag{2.11}$$

$$F1_b = \frac{2TP}{2TP+FP+FN}. \tag{2.12}$$

The macro label-based version computes the previous measures for each label and returns their average value, such that

$$\text{macro-}\beta = \frac{1}{q}\sum_{j=1}^{q}\beta(TP_j, FP_j, TN_j, FN_j),$$

where $\beta = \{Precision_b \mid Recall_b \mid F1_b\}$, from Equations 2.10, 2.11 and 2.12, respectively.

The label prediction problem measures, Missing Label Prediction (MLP) and Wrong Label Prediction (WLP), (RIVOLLI; SOARES; CARVALHO, 2018a) will be also considered. The MLP measure indicates the proportion of labels that are never predicted by a strategy. The WLP measure, which can be seen as a generalization or relaxation of MLP, represents the case where a label might be predicted for some instances, but these predictions are always

wrong. Equations 2.13 and 2.14 formalize these measures, respectively. In an ideal scenario, their expected value is zero.

$$MLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j + FP_j == 0) \tag{2.13}$$

$$WLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j == 0) \tag{2.14}$$

2.4.2.3 Ranking measures

Ranking measures consider the ranking of labels instead of the quality of bipartitions, which defines the labels predicted. *One-error* (OE) is an extreme measure that only assesses the error of the label predicted with most confidence. This measure is computed as follows:

$$OE = \frac{1}{n}\sum_{i=1}^{n} I(\arg\max_{\lambda_j \in \mathscr{L}} f(x_i, \lambda_j) \notin Y_i)$$

Ranking-loss (RL) computes the average rate of label pairs that are incorrectly sorted when using their predicted probabilities. It is calculated as follows:

$$RL = \frac{1}{n}\sum_{i=1}^{n} \frac{|\{(\lambda_j, \lambda_k) | f(x_i, \lambda_j) \leq f(x_i, \lambda_k), (\lambda_j, \lambda_k) \in Y_i \times \overline{Y}_i\}|}{|Y_i||\overline{Y}_i|},$$

where $\overline{Y_i} = \mathscr{L} \setminus Y_i$.

2.4.3 Multi-label baselines

Different baselines were adopted, optimizing different measures. With the exception of the baseline$_{RL}$, they were proposed by Metz *et al.* (2012). The baseline$_{F1}$ literally predicts the label set that maximizes the F1 measure (Equation 2.9) for the training data, such that

$$baseline_{F1} = \arg\max_{\hat{Y} \subseteq \mathscr{L}} F1(Y, \hat{Y}),$$

where $\hat{Y}$ is the label set predicted. This baseline is also used to compare the label based measures *macro-F1*, *macro-precision* and *macro-recall*.

The baseline$_{HL}$ predicts the labels present in more than 50% of the training instances, such that

$$baseline_{HL} = \{\lambda_j \mid freq(\lambda_j) > 0.5, 1 \leq j \leq q\},$$

where $freq(\lambda_j)$ is the frequency of the label λ_j in the training data. In turn, baseline$_{SA}$ predicts the most frequent labe lset in the training data, such that

$$baseline_{SA} = \arg\max_{\hat{Y} \subseteq \mathscr{L}} \sum_{i=1}^{n} I(Y_i = \hat{Y})$$

where I is the indicator function defined in Equation 2.8.

Finally, the baseline$_{RL}$ (RIVOLLI; SOARES; CARVALHO, 2018a), an adaptation of the *General*$_B$ baseline (METZ *et al.*, 2012), predicts a ranking of labels according to their frequency, such that

$$rank(\lambda_j) = |\mathscr{L}| - |\{\lambda_k \mid \lambda_k \in \mathscr{L}, freq(\lambda_j) > freq(\lambda_k)\}|,$$

and

$$baseline_{RL} = \{\lambda_j \mid rank(\lambda_j) \leq lcard,\ 1 \leq j \leq q\},$$

where *lcard* is the label cardinality of the training data. This baseline is used for the ranking measures: *one-error* and *ranking-loss*.

2.4.4 Base algorithms

The strategies described in Section 2.3 require using a base algorithm to induce binary models. Algorithms that are frequently used as the base algorithm in multi-label experiments are *Decision Tree Induction Algorithms* (CHERMAN; METZ; MONARD, 2012; ALALI; KUBAT, 2015; TSOUMAKAS *et al.*, 2009), *Logistic Regression* (LR) (MONTAÑÉS *et al.*, 2014; RAUBER *et al.*, 2014; SENGE; COZ; HÜLLERMEIER, 2013; TSOUMAKAS *et al.*, 2009) and *Support Vector Machines* (SVM) (READ *et al.*, 2011; CHERMAN; METZ; MONARD, 2012; LI; ZHANG, 2014; LUACES *et al.*, 2012; MADJAROV *et al.*, 2012; TSOUMAKAS *et al.*, 2009).

Two classification algorithms that have been very successful in classification tasks, but not commonly used for multi-label classification, *Random Forest* (RF) and *eXtreme Gradient Boosting* (XGB), complete the set of base algorithms used in our experiments.

The *k-Nearest Neighbors* (kNN) and *Naive Bayes* (NB) algorithms were initially considered. They were discarded because they did not show competitive results when compared with the others. Although other base algorithms, such as Multilayer Perceptron, could also be investigated, they were not considered because those selected were able to support the claims addressed in this paper.

2.4.5 Experimental setup

The experiments were carried out using the R environment. The data sets were handled using code from the `mldr` package (CHARTE; CHARTE, 2015b). The strategies used R code from the `utiml` package (RIVOLLI; CARVALHO, 2018). By default, `utiml` prevents empty predictions (LIU; CHEN, 2015), in which case the strategy outputs the label with the highest probability/score, preventing an example from being predicted without any labels.

Most strategies and base algorithms used in the experiments require the definition of hyperparameter values. Tables 3 shows, for each strategy used, the default values recommended by the packages for the main hyperparameters.

Table 3 – Hyperparameters values for the strategies used in the experiments.

Strategy	Parameters/Values
BR/DBR	-
CC/NS	`chain = random(`$\mathscr{L}$`)`
BR+	`strategy = "Dyn"`

EBR/ECC	`m=10`
	`subsample = 0.75`
	`attr.space = 0.5`

MBR/PruDent	`phi = 0.1`
RDBR	`max.iterations = 5` `batch.mode = FALSE`

The implementation of the base algorithms used in the experiments come from the packages `C50`, `stats`, `randomForest`, `e1071` and `xgboost` for C5.0, LR, RF, SVM and XGB, respectively. Table 4 shows the values used for the hyperparameters of each base algorithm, which were those recommended in the corresponding package. SVMt is a tuned version of SVM for the *macro-F1* measure, where the range of values used in a Grid Search procedure is reported. To validate the hyperparameter values, holdout with 70% for training and 30% for validation is adopted for all data sets. SVM was singled out for tuning, due to the high effect of hyperparameter values on its performance (MANTOVANI *et al.*, 2015).

Table 4 – Hyperparameter values of the base algorithms used in the experiments.

Base algorithm	Parameters/Values	Reference

C5.0	`trials = 1`
	`CF = 0.25`
	`minCases = 2`
		Quinlan (1993)

LR	-	Gelman and Hill (2007)
RF	`ntree = 500`	Breiman (2001)

SVM	`kernel = "radial"`
	`cost = 1`
	`gamma = 1 / d`
		Chang and Lin (2011)

SVMt	`kernel = "radial"` `cost =` $[2^{-5}, 2^{-3}, \cdots, 2^{13}, 2^{15}]$ `gamma =` $[2^{-15}, 2^{-13}, \cdots, 2^{1}, 2^{3}]$	Madjarov *et al.* (2012)

XGB	`nrounds = 100`
	`eval_metric = "error"`
	`early_stop_round = 2`
		Chen and Guestrin (2016)

All results were obtained using 5x2-fold cross-validation with paired folds across all combinations of strategies and base algorithms. An iterative algorithm for the stratification of

multi-labeled data (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011) was applied to ensure similar label distributions between training and test data.

Different from previous comparative studies in the multi-label domain, two Bayesian statistical tests were used (BENAVOLI *et al.*, 2017). The Bayesian hierarchical correlated t-test was used to compare two strategies over multiple data sets, whereas the Bayesian correlated t-test was used for a single data set. When comparing two strategies, the Bayesian statistical test outputs the probability of three situations: strategy 1 is the best (left); strategy 2 is the best (right); and there is a draw between them (rope), which is a region of practical equivalence that indicates an insignificant difference in performance between the strategies. Benavoli *et al.* (2017) suggest the interval $[-0.01, 0.01]$, which represents a difference of 1% for a measure whose range is $[0, 1]$. This interval was used for all evaluation measures, with the exception of *hamming-loss*, where the interval was modified $[-0.001, 0.001]$ due to its finer granularity when compared to the other measures. Otherwise, no statistical differences was observed, given that, for *hamming-loss*, the number of mistakes made by a strategy is divided by the number of test instances times the number of labels. Thus, the larger the data set, the smaller the differences between the strategies.

2.5 Experimental results

This section presents the experimental results and the main findings from this study. The complete set of experimental results is publicly available online at <https://rivolli.github.io/ml-binary-transformation/>.

Initially, this section compares the results with multi-label baselines followed by the comparison of the most similar strategies. Next, the strategies are compared using fixed base algorithms, which is the traditional approach used in the multi-label literature. Afterwards, the base algorithms are compared by fixing the strategies. In the last set of comparisons, both strategies and base algorithms are combined without distinction. Finally, the main findings are highlighted.

2.5.1 Comparison with the baselines

Despite their importance for evaluating predictive performance, baselines have not been frequently used in multi-label experiments (METZ *et al.*, 2012). As a result, there are no clear standards for selecting baselines for evaluation. Table 5 presents a comprehensive set of results for the different baselines (Section 2.4.3) used in the experiments.

The baseline$_{F1}$ obtained the highest results for all measures in data sets with high average labels' frequency and low imbalance degree. The baseline$_{HL}$, on the contrary, had its best results in data sets with low average label frequency and high imbalance degree. Regarding the baseline$_{RL}$, used to evaluate the ranking measures, the results obtained are inversely correlated

Table 5 – Baseline values obtained for each data set and measure.

Data set	Baseline$_{F1}$ ↑				Baseline$_{HL}$ ↓	Baseline$_{RL}$ ↓		Baseline$_{SA}$ ↑
	F1	F1$_m$	Prec$_m$	Rec$_m$	HL	OE	RL	SA
20NG	0.098	0.098	0.051	1.000	0.096	0.948	0.505	0.052
birds	0.288	0.096	0.059	0.267	0.149	0.694	0.316	0.087
cal500	0.478	0.156	0.112	0.282	0.165	0.116	0.212	0.000
corel5k	0.204	0.006	0.003	0.018	0.018	0.776	0.194	0.010
emotions	0.464	0.472	0.312	1.000	0.330	0.555	0.409	0.125
enron	0.463	0.057	0.042	0.095	0.078	0.464	0.141	0.088
fapesp	0.198	0.059	0.033	0.250	0.115	0.857	0.374	0.096
flags	0.699	0.528	0.427	0.714	0.328	0.211	0.220	0.097
image	0.389	0.395	0.247	1.000	0.331	0.710	0.458	0.189
langlog	0.145	0.015	0.008	0.079	0.053	0.857	0.271	0.094
mediamill	0.516	0.027	0.022	0.040	0.036	0.197	0.068	0.056
medical	0.249	0.044	0.027	0.145	0.082	0.720	0.252	0.174
msd-195	0.246	0.051	0.031	0.158	0.078	0.751	0.226	0.082
ohsumed	0.270	0.046	0.029	0.130	0.091	0.716	0.254	0.084
scene	0.302	0.303	0.179	1.000	0.272	0.779	0.473	0.168
slashdot	0.220	0.067	0.038	0.278	0.104	0.845	0.270	0.139
stackex	0.188	0.011	0.006	0.040	0.033	0.737	0.232	0.065
tmc2007	0.447	0.076	0.054	0.136	0.093	0.408	0.163	0.087
yeast	0.576	0.311	0.236	0.500	0.232	0.249	0.211	0.095
yelp8	0.494	0.284	0.203	0.500	0.260	0.411	0.296	0.080

with the label cardinality, i.e. the lowest ranking-loss values were observed in data sets with high *lCard*. Finally, as the number of labels and label sets increase, the results obtained for the baseline$_{SA}$ decrease.

Figure 1 summarizes the number of strategy/base-algorithm pairs that did not perform statistically significantly better than the baselines for each data set and evaluation measure. With the exception of *macro-recall*, that can be easily maximized by predicting all labels, and some other measures in the case of the cal500 data set, at least one combination strategy/base-algorithm was always able to outperform the baselines for all measures and data sets. However, the considerable number of non-zero entries in Figure 1 corroborates the claim of Metz *et al.* (2012) that any new strategy should be compared with others using appropriate multi-label baselines.

2.5.2 Similarity of strategies

How the base algorithms affect the behavior of the binary transformation strategies is one of the questions investigated in this paper. According to Table 1, it is reasonable to assume that strategies within a group/subgroup are more similar to each other than the rest. However, the transformation strategies work with a base algorithm, which is used to induce the learning models from the transformed data, and its effect over the strategies is unknown so far. Following this rationale, the similarity of strategies using different base algorithms is analyzed in two distinct ways. First, by comparing their predictions, which removes the bias of a specific evaluation

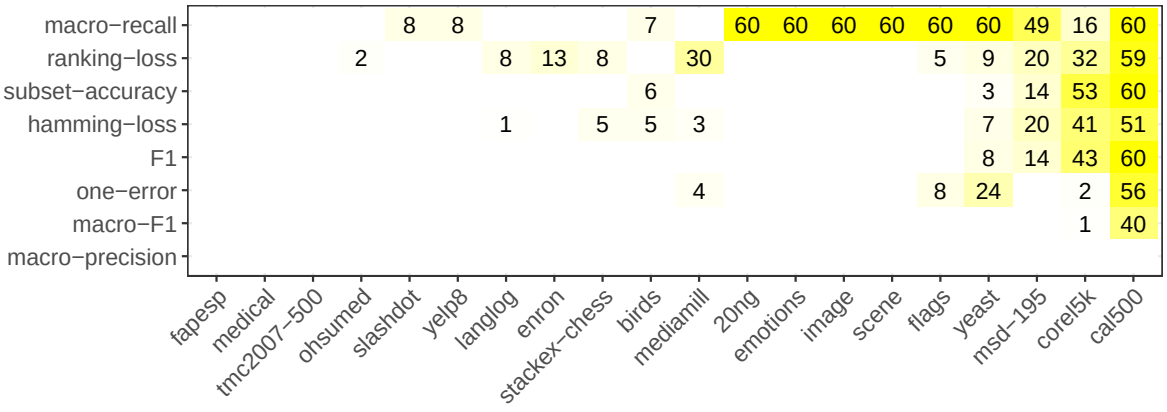


Figure 1 – Number of pairs strategy/base-algorithm that did not perform statistically significantly better than the baselines according to different evaluation measures.

measure. Second, by comparing their predictive performance statistically over distinct evaluation measures, which considers particularities of the learning process.

To compare the predictions obtained by the strategies, the Hamming distance (defined in Equation 2.7) is computed for each pair of strategies. The result indicates the difference between the predictions, and therefore, the average value over all data sets and repetitions can indicate how similar or distinct any two given strategies are.

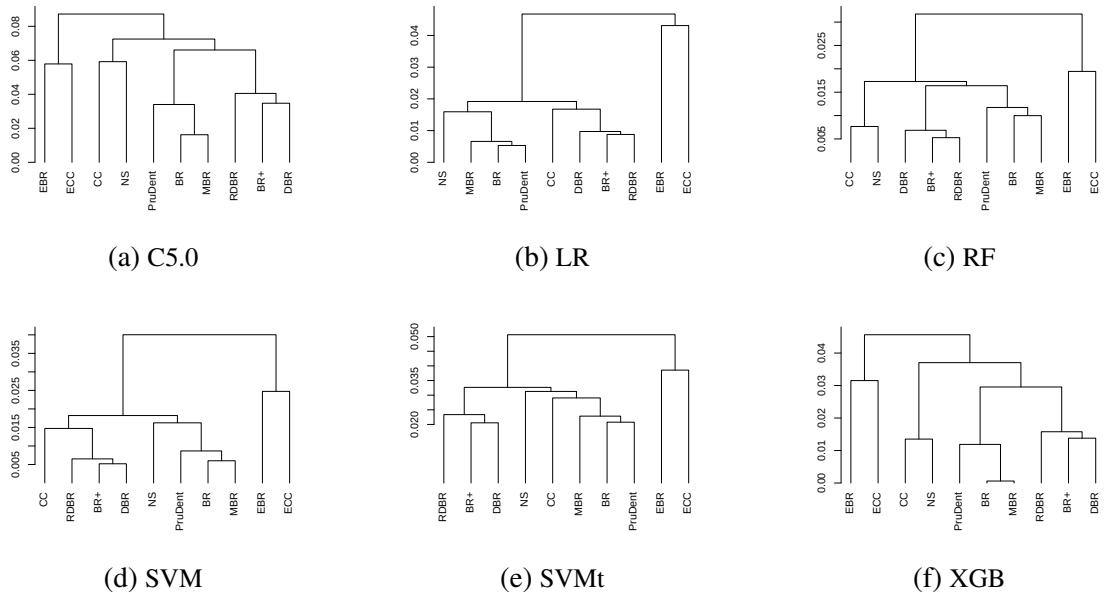
Initially, by fixing the base algorithm, the strategies were compared. For such, they were organized according to their similarity using the hierarchical clustering algorithm Averaged-Linkage (JAIN; DUBES, 1988). Figure 2 shows the hierarchy of strategies for each base algorithm. Similar results are observed regardless of the base algorithm, with some exceptions. In summary, the similarity of the predictions follows the intuition of the groups of strategies presented in Table 1.

For all base algorithms, the ensembles EBR and ECC presented the largest difference to all others. The full stacking BR+, DBR and RDBR were grouped together, following different paths, according to the base algorithm. These are the only consensus in the results. Other strategy pairs, such as the chaining CC and NS were the closest strategies only for the base algorithms C5.0, RF and XGB. Similarly, pruned stacking MBR and PruDent were not always in the same group.

Regarding the subgroups, the chaining strategies were more similar to the full stacking for some base algorithms, and to the pruned stacking for others. Pruned stacking was more related to BR than full stacking, which may indicate that the pruning approach impacted the results more than the use of stacking, for these strategies.

Looking at the base algorithms, the use of C5.0 leads to a larger difference among the results obtained by the strategies, and, on the other hand, RF leads to a higher similarity.

Next, when all the strategy/base-algorithm pairs were compared together (Figure 3), the



(a) C5.0 (b) LR (c) RF

(d) SVM (e) SVMt (f) XGB

Figure 2 – Similarity of strategies according to their bipartition predictions.

similarity between the base algorithms could also be compared. The base algorithms RF and XGB produced similar results, and likewise for SVM and LR. In the latter case, the similarity observed was still stronger than the former, since the same strategies using distinct base algorithms were clustered together. On the other hand, SVM and SVMt, despite being the same base algorithm using different hyperparameter values, were not so closely related as SVM and LR were.

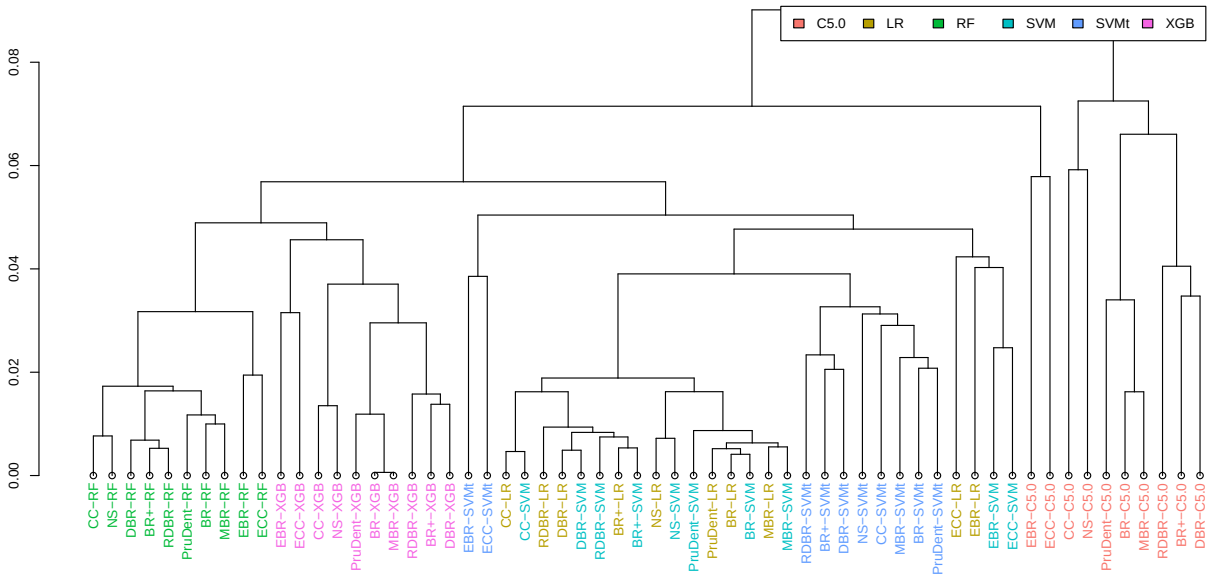


Figure 3 – Similarity of strategies and base algorithms according to their bipartition predictions.

With the exception of the ensembles and the SVM and LR base algorithms, all strategies are clustered according to the base algorithm, instead of the opposite, i.e. different variants of the same strategy grouped together. For instance, in this comparison, BR_{RF} is more similar to

DBR_{RF}, a full stacking approaching, than to BR_{XGB}. This shows that, for these strategies, their differences might not be strong enough to always be apparent, regardless of the choice of base algorithm.

To identify when small differences in prediction are significant, the pairs of strategies within a group/subgroup were statistically compared. The investigated hypothesis remains that the two distributions are equal such that a high probability means that the two strategies are similar and a low value that the two strategies are indeed dissimilar as one would be interested in. Figure 4 presents the rope probability of different pairs of strategies. The pairs are sorted according to their average values, from the most similar to the most distinct (from the bottom to the top). Likewise, the base algorithms are sorted from left to right.

Figure 4 – Rope probabilities from the Bayesian hierarchical test in the comparison of related strategies (y axis) for different base algorithms (x axis). The symbol '=' is used for probabilities greater than 0.95.

As previously observed, C5.0 was the base algorithm with the largest number of differences between strategies, whereas RF was the base algorithm with the lowest number of differences. Regardless of the evaluation measure, all pairs were considered similar to each other when RF was used. Additionally, the differences between the strategies were captured in different ways by the evaluation measures. For instance, no differences in *F1* results were observed; the ranking measures were more sensitive when comparing the pruned stacking strategies; and *hamming-loss* and *subset-accuracy* produced clear differences for the ensemble and full stacking strategies.

In summary, the results presented in this section showed that the base algorithms impact the strategies in different ways. Despite all the investigated strategies using the same paradigm (binary transformation), their small differences were captured by the evaluation measures for some of the base algorithms. By varying the base algorithm, a pair of close-related strategies can be seen as more similar, or more distinct, to each other, given a specific evaluation measure.

Therefore, it can be concluded that some base algorithms are more dominant than others.

2.5.3 Analysis of strategies

Following the procedure used in many multi-label studies, the strategies are compared with each other by fixing the base algorithm. As distinct base algorithms are considered, the differences between them can be contrasted. Using the Bayesian hierarchical statistical test, each pair of strategies with the same base algorithm is compared with each other. Figure 5 presents the results of the paired test, varying the base algorithms. For each base algorithm, the strategy whose probability to statistically outperform the other is higher than or equal to 95% is highlighted. Similar algorithms (rope $\geq 95\%$) are represented with an "=" character and an empty value indicates inconclusive results (probabilities $< 95\%$). The pairs of strategies with similar or inclusive results for all base algorithms were removed from the chart.

The main discrepancies in the results are observed in relation to the ensemble strategies and the base algorithm C5.0. For C5.0, EBR and ECC outperformed all other strategies for most evaluation measures, whereas for other base algorithms, ensembles were outperformed by different strategies. For the measures *F1*, *macro-F1* and *macro-recall* a more homogeneous result is observed across the base algorithms. In this case, the ensembles are clearly the best choice, probably due to the fact that they internally perform a thresholding calibration that allows them to obtain more balanced precision and recall results regardless of the base algorithm.

To detail the contradictions, Table 6 presents the cases where conflicting probabilities from the statistical test were found across distinct base algorithms. Probabilities indicating that the strategies are similar (rope $> 50\%$) and inconclusive results (all probabilities $< 50\%$) were omitted from the table, which led to the elimination of the columns relative to base algorithms RF and SVM. The bold markup highlights, for each base algorithm, the highest value and the cases where the probability is greater than or equal to 95% are underlined.

Table 6 – Divergent probabilities found across the base algorithms in the comparison of the strategies. Left and right are the probabilities obtained in the Bayesian hierarchical test.

Measure	Strategies	C5.0		LR		SVMt		XGB	
		left	right	left	right	left	right	left	right
HL	CC x DBR	**0.53**	0.00					0.35	**0.59**
Rec_m	BR x NS	**0.68**	0.01			0.04	**0.79**		
	NS x PruDent					**0.53**	0.03	0.08	**0.59**
OE	CC x MBR					**0.74**	0.09	0.3	**0.50**
RL	BR+ x MBR	0.24	**0.74**					**1.00**	0.00
	BR+ x PruDent	0.01	**0.81**	**1.00**	0.00				
	DBR x MBR	0.26	**0.72**					**1.00**	0.00
	DBR x PruDent	0.01	**0.86**	**1.00**	0.00				
	MBR x PruDent	0.05	**0.89**	**1.00**	0.00			0.00	**1.00**
	MBR x RDBR	**0.84**	0.15					0.01	**0.99**
	PruDent x RDBR	**0.97**	0.00	0.00	**1.00**				

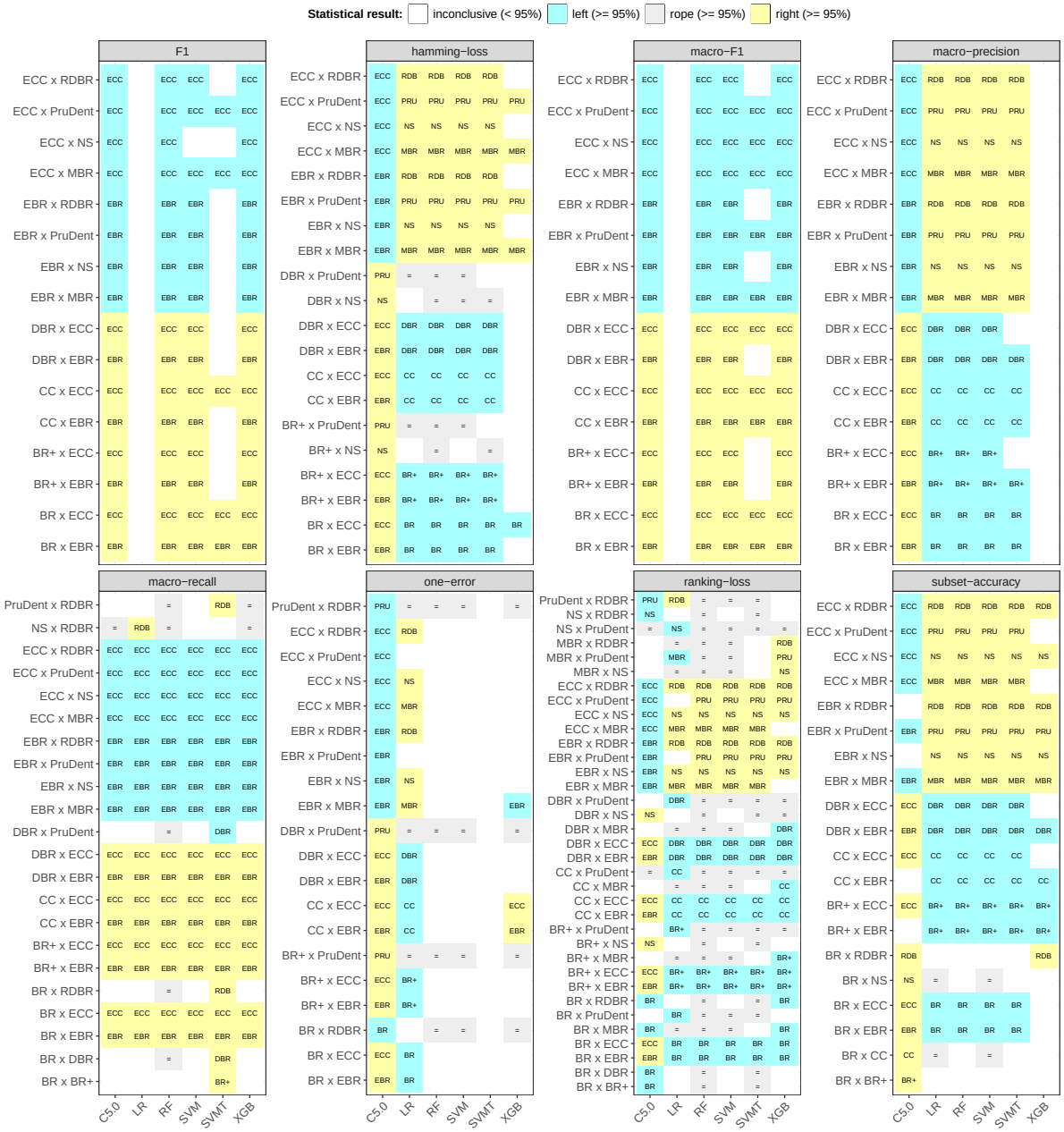


Figure 5 – Best strategy according to the results of the Bayesian hierarchical statistical test. The symbol '=' indicates they are similar with statistical significance.

Many observations showed low probabilities at least for one of the base algorithms. This indicates that the differences were not so evident according to the statistical test, even though they are still conflicting. In this sense, the most noticeable differences were observed in the ranking-loss measures, probably because the scores produced by the binary models are more sensitive to variation than the bipartitions.

Regarding the base algorithm, C5.0 shows many strongly significant differences, which reinforces the previous conclusions concerning C5.0 behaving very differently from the other base algorithms. Regarding the strategies, all observed differences are related to pairs of strategies where each comes from a different subgroup, e.g., a chaining strategy against a full stacking

strategy.

In conclusion, the comparison of the transformation strategies showed different results, for some measures, according to the base algorithm used. In this particular case, all strategies use a binary transformation, which makes them very similar to each other. Given that differences were still observed, it is reasonable to assume that when different transformation strategies are evaluated, it is important to investigate distinct base algorithms.

2.5.4 Analysis of base algorithms

Exploring a different perspective, the base algorithms are compared by fixing the strategies. The hypothesis investigated is that for each strategy some specific base algorithms perform better than the rest. Analogous to the previous section, Figure 6 presents the results of the paired test for base algorithms, in which all base algorithms were compared against each other for each one of the strategies. In this test, for each strategy, the algorithm whose probability to statistically outperform the other is higher than or equal to 95% is highlighted. Similar algorithms (rope $\geq$ 95%) are represented with an "=" character and an empty value indicates inconclusive results (probabilities $<$ 95%).

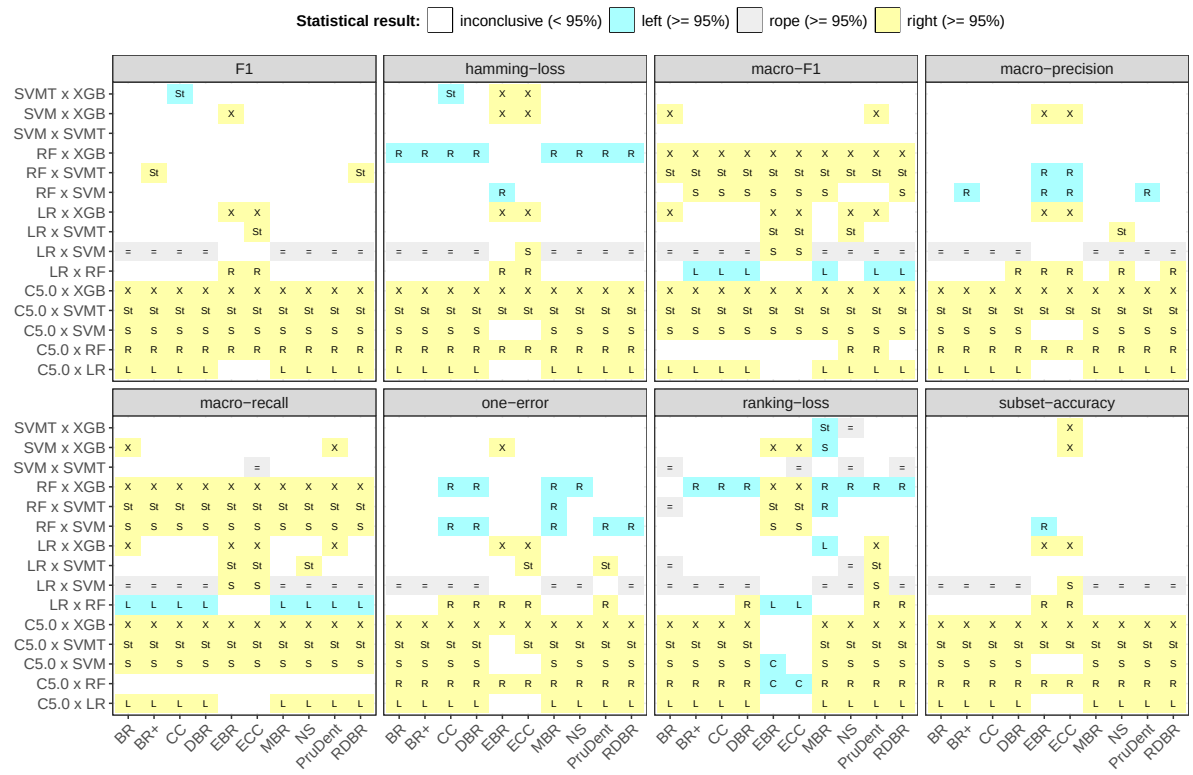


Figure 6 – Best base algorithm according to the results of the Bayesian hierarchical statistical test. The best option for each pair and strategy is indicated by the first letter of the base algorithm, such that C, L, R, S, St and X indicate C5.0, LR, RF, SVM, SVMt and XGB, respectively. The symbol '=' indicates they are similar with statistical significance.

At a glance, RF and XGB were the dominant base algorithms, regardless of the evaluation

measure used. However, they have not been used as the base algorithm in previous studies. In contrast, C5.0, followed by LR, obtained the worst results, despite their popularity in multi-label studies.

Probably due to the lack of diversity in the strategies considered, few variations concerning the best base algorithm were observed. Nevertheless, they are related to the ensembles, the most distinctive strategies among the ones investigated, as noticed in Section 2.5.2. An illustrative example that reinforces the investigated hypothesis is related to the *ranking-loss* measure. For many strategies, RF was the best base algorithm. However, for the ensembles, it was the worst. On the other hand, C5.0, which is not a good choice for many strategies, is a suitable alternative for the ensembles. This is very plausible, as ensemble-based base algorithms, similar to RF, perform better when their base learners are unstable – which is why decision tree induction algorithms (e.g., C5.0) are popular choices inside ensembles of machine learning algorithms. Since the predictions of ensemble-based base algorithms themselves reduce variance, they are not as suitable for ensembles strategies.

For some comparisons and evaluation measures, one of the base algorithms was statistically better than the other regardless of the strategy, mainly when C5.0 was involved, which typically is the worst of the two. In spite of this regularity, the results reinforce the conjecture that the performance of strategies depends on the base algorithm. In particular, the results of the ensemble strategies presented a greater variation, concerning the best base algorithms, compared to the other strategies. However, additional tests, including a more varied set of strategies, can increase support for this claim.

Some pairs of base algorithms, in particular LR/SVM and SVM/SVMt, presented similar results, with statistical significance, for different evaluation measures. Between LR and SVM, the latter was the best option only for the ensembles, but not for all measures. Comparing SVM and its optimized version, SVMt, despite the fact the latter performed apparently better than the former in terms of *F1*, *macro-F1* and *macro-recall*, the probabilities obtained in the Bayesian test were not greater than or equal to the 95%. Regarding C5.0 and LR, the latter shows clear advantages over the former. Finally, between RF and XGB, the most dominant base algorithms according to the experimental results, the choice between one of them depends on the evaluation measure. XGB was the best option for *macro-F1* and *macro-recall*, while RF was the best for *hamming-loss*, *one-error*, and *ranking-loss*.

In summary, the results presented in this section provide some support for the claim that the choice of base algorithm can strongly influence a strategy's performance. Furthermore, some base algorithms performed better on average than others, which again can influence and even distort comparisons of multi-label learning strategies.

2.5.5 Combining strategies and base algorithms

The previous analyses showed that the ranking of the best strategies varies according to the base algorithm used. To further investigate this issue, all strategy/base-algorithm pairs are evaluated against each other without distinctions. In order to summarize the 60 pairs (strategy/base-algorithm), Appendix A presents the ranking for each pair considering all data sets and the strategies' results using the best base algorithm. The statistical results comparing those strategies are presented in Appendix B.

Considering the BR strategy as a more robust baseline, its performance is analysed in relation to the other strategies. For the measures *F1*, *macro-F1* and *macro-recall* the ensembles outperform BR with statistical significance, regardless of the base algorithm. By contrast, BR outperforms them to the measures *hamming-loss*, *macro-precision*, *ranking-loss* and *subset-accuracy*. In relation to the other strategies, there is no case in which BR is completely outperformed by other strategy and vice-versa. Specifically for *one-error* measure, BR_{RF} achieved the best ranking over all combinations and outperformed the other strategies for 4 or 5 base algorithms.

To complement these results, Table 7 presents, for all the selected pairs, the number and percentage of other pairs that were statistically outperformed with a probability greater than or equal to 95%, according to the Bayesian statistical test. The strategies are sorted from top to bottom based on the number of pairs outperformed.

None of the strategies obtained a reasonable performance over all evaluation measures. The highest results are observed for the ensembles using XGB that outperformed more than 90% of the other strategies in terms of *F1*, *macro-F1* and *macro-recall*. Consequently, they are the best ranked pairs of strategy/base-algorithm according to the number of outperformed pairs. The lack of a dominant combination for the other measures shows that all the strategies obtained a good performance for some base algorithms.

Concerning the base algorithms, the best results were obtained mainly by either RF or XGB. Both algorithms are represented in the table by all strategies. In terms of strategies, despite being the simplest, BR presented a good performance for the *hamming-loss*, *one-error* and *ranking-loss*.

To sum up, when all strategies/base-algorithms pairs are compared, some strategies appear as dominant for some measures regardless of the choice of base algorithm, such as EBR and ECC for *macro-F1*. On the other hand, for some evaluation measures, the choice of the base algorithm dominates the results, regardless of the chosen strategies, such as RF for *ranking-loss*. Even though all strategies use binary transformation, and consequently are very similar to each other, statistical differences were observed between them. In conclusion, an empirical comparison of multiple transformation strategies together with multiple base algorithms should be considered for any future study proposing new transformations.

Table 7 – Selected pairs of strategy/base-algorithm and the percentage of other pairs that were statistically outperformed by them.

Strategy/base-algorithm	F1	$F1_m$	$Prec_m$	Rec_m	HL	OE	RL	SA
EBR_{XGB}	90%	92%	24%	92%	27%	27%	19%	20%
ECC_{XGB}	90%	85%	25%	92%	27%	22%	20%	27%
$PruDent_{RF}$	14%	2%	39%	0%	49%	58%	58%	32%
MBR_{LR}	14%	25%	24%	27%	32%	20%	37%	25%
$RDBR_{SVMt}$	32%	46%	25%	47%	29%	20%	37%	39%
DBR_{SVMt}	19%	36%	25%	44%	34%	24%	37%	36%
BR_{RF}	14%	2%	32%	0%	47%	66%	59%	32%
NS_{RF}	14%	12%	36%	0%	41%	53%	53%	39%
$BR+_{SVM}$	14%	27%	24%	27%	32%	20%	37%	31%
CC_{RF}	14%	7%	31%	0%	49%	53%	53%	37%
MBR_{XGB}	14%	46%	27%	36%	34%	19%	22%	32%
BR_{XGB}	14%	46%	27%	36%	32%	20%	39%	31%
$PruDent_{XGB}$	14%	47%	27%	39%	34%	20%	37%	31%
CC_{XGB}	14%	34%	27%	27%	27%	17%	37%	32%
NS_{XGB}	14%	37%	27%	27%	27%	17%	37%	34%
$BR+_{SVMt}$	17%	37%	25%	41%	34%	24%	37%	36%
MBR_{RF}	14%	3%	37%	0%	49%	53%	47%	32%
$RDBR_{RF}$	14%	0%	47%	0%	42%	29%	53%	37%
$BR+_{RF}$	14%	3%	42%	0%	42%	27%	51%	37%
DBR_{RF}	14%	3%	39%	0%	42%	49%	53%	36%
EBR_{SVM}	42%	64%	14%	92%	14%	14%	5%	7%
DBR_{XGB}	14%	42%	25%	34%	27%	19%	37%	32%
$RDBR_{XGB}$	14%	42%	27%	34%	31%	17%	37%	36%
$BR+_{XGB}$	14%	42%	27%	36%	29%	19%	37%	34%
EBR_{RF}	81%	27%	27%	29%	22%	20%	0%	20%
ECC_{RF}	88%	27%	27%	32%	20%	19%	0%	22%
NS_{SVMt}	14%	32%	32%	32%	31%	22%	37%	34%
CC_{SVMt}	14%	36%	31%	34%	36%	22%	37%	36%

2.5.6 Label prediction problems

It can be observed in Figure 7 that the values of *F1* are substantially higher than the values of *macro-F1* for many data sets. This occurs when the value of *F1* is very low for one or more labels. In practice, the least common labels are often behind these differences. As the previously defined label prediction problems MLP and WLP (Equations 2.13 and 2.14) provide a possible explanation, their average proportions over all strategy/base-algorithm pairs are presented in Table 8.

For the sake of clarity, the data sets without problems were removed from the table. For many data sets, the values obtained paint a clear picture, indicating that many labels were wrongly predicted or even never predicted at all. E.g., in the worst case, on average 73% of the labels from the `corel5k` ($\approx$ 159 labels) were wrongly predicted for all test instances, and 55% ($\approx$ 120 labels) were never predicted. The high values observed for many data sets indicate a problem generated by the binary transformation strategies not previously detected.

This also justifies the high *macro-precision* values in comparison with the *macro-recall* values (Figure 8). The best results for the measures *F1*, *macro-F1* and *macro-recall* were achieved

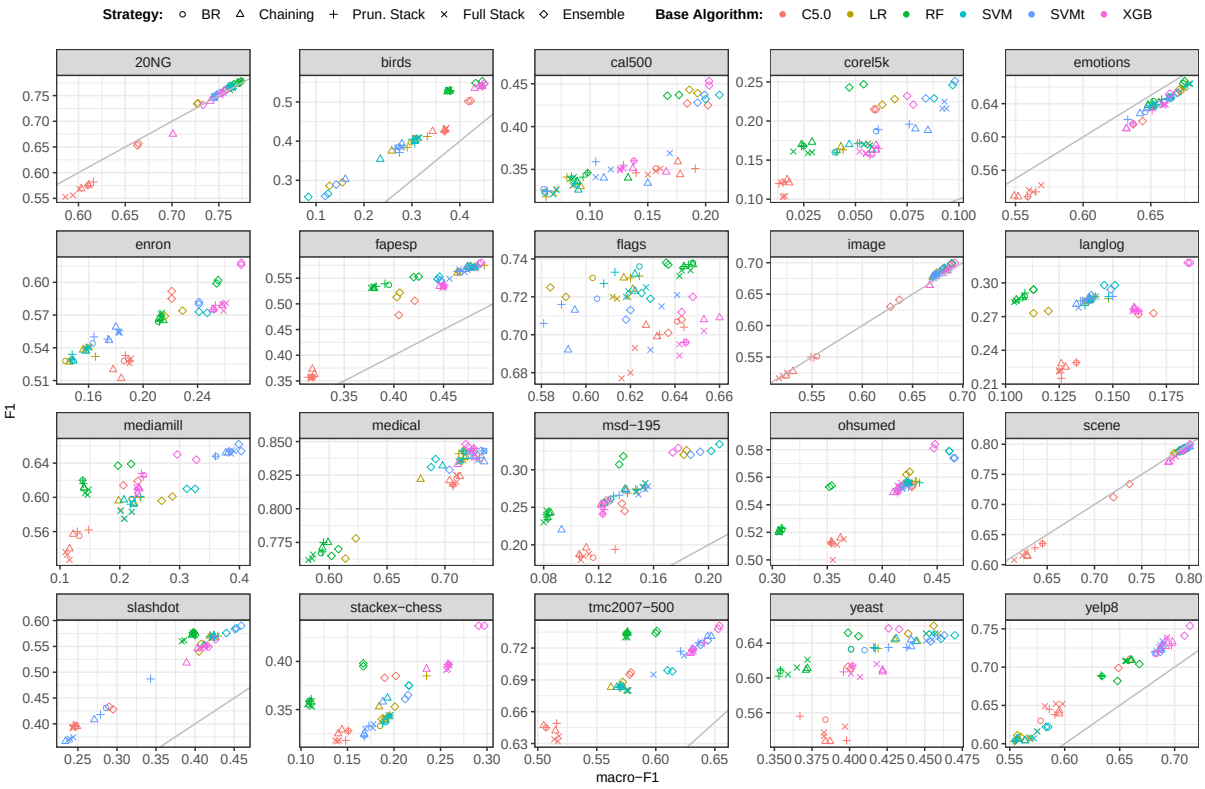


Figure 7 – Comparative results of the measures *F1* and *macro-F1* for all data sets and strategy/base-algorithm pairs.

Table 8 – Average label prediction problems results over all strategy/base-algorithm pairs.

Data set	MLP	WLP
flags	0.03	0.04
ohsumed	0.06	0.07
medical	0.06	0.10
yeast	0.10	0.11
fapesp	0.13	0.19
slashdot	0.15	0.20
birds	0.15	0.23
mediamill	0.17	0.20
msd-195	0.24	0.34
enron	0.29	0.44
stackex-chess	0.32	0.45
langlog	0.34	0.47
cal500	0.37	0.54
corel5k	0.55	0.73

by the strategy ensembles. Since they use an internal threshold technique for selecting relevant labels, their *recall* is enhanced and, consequently, their *F1* result is also higher. Additional studies are needed to test if this behavior is mainly due to this post-processing used by the ensembles.

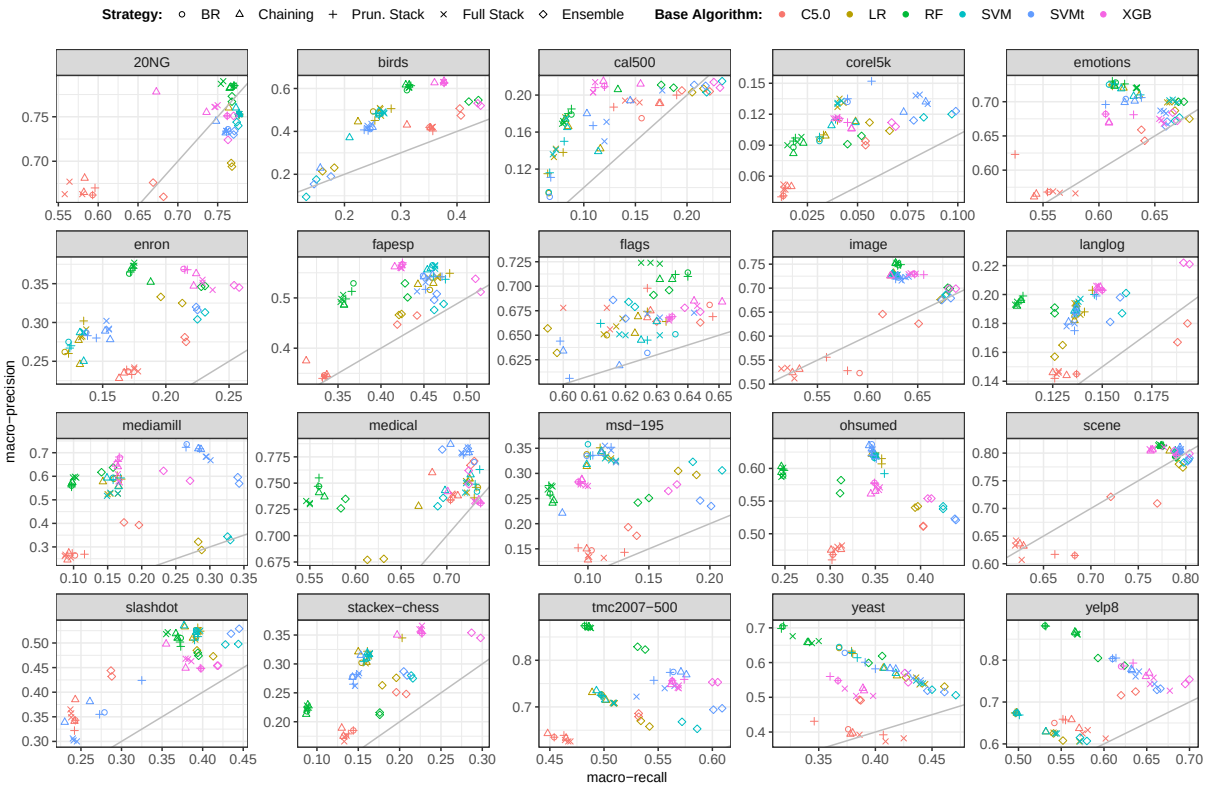


Figure 8 – Comparative results of the measures *macro-precision* and *macro-recall* for all data sets and strategy/base-algorithm pairs.

2.5.7 Summary

The main motivation for this study was to obtain a better understanding of how the base algorithm impacts the binary transformation strategies. The results presented in the previous sections show that the choice of the base algorithm can interfere in the behaviour of binary transformation strategies. Thus, by considering distinct base algorithms, an empirical study involving transformation strategies can become less biased.

Different rankings of strategies and statistical results were obtained by using different base algorithms. This, however, is not common practice in multi-label research. Usually transformation strategies are proposed and compared using a single base algorithm (READ *et al.*, 2011; MADJAROV *et al.*, 2012; MONTAÑÉS *et al.*, 2014; MOYANO *et al.*, 2018). The claim that by segmenting the comparison of base algorithms more consistent results can be obtained (MOYANO *et al.*, 2018) might actually be misleading. In addition, across all assessed measures, there was not a single base algorithm that obtained the best results for all strategies. Consequently, performing a comparison of strategies using only one fixed base algorithm should be avoided.

Nevertheless, it is still valid to compare the strategies using a fixed base algorithm, since it can help with understanding the scenarios in which a strategy is improved. For instance, a clear superiority of the ensembles EBR and ECC, regardless of the evaluation measure, was observed when the base algorithm C5.0 was used. On the other hand, when using the LR and

RF algorithms, ensemble strategies did not perform so well, showing that for a given base algorithm some strategies might not be suitable. Even though some base algorithms might obtain a better overall performance than others, the diversity of base algorithms is valid to determine the conditions in which each strategy is convenient. Furthermore, although predictive performance is very important, there are reasons one may consider different base classifiers. For example, decision trees provide good interpretation, logistic regression provides good probability estimates. Therefore, it is useful to consider the relative performance difference rather than simply the top performance.

Considering the large experimental scenario evaluated, the hyperparameter tuning procedure adopted was simple and did not achieve the best results for the optimized measure. The use of the SVMt base algorithm produced distinct results when compared to SVM, but when compared to others, such as RF and XGB, the SVMt results were more similar to SVM. Therefore, in this context, hyperparameter tuning can be seen secondary to base algorithm selection, provided reasonable default parameter settings can be identified for the selected base algorithm. However, we remark that this indeed depends on the model class in question; in which some models are more sensitive to initial hyperparameter settings than others. Ideally, if computational power allows for it, then the base algorithms should be tuned as part of the base-algorithm selection process, especially if the performance difference between them is not great. Of course, for large scale experimental comparisons, this may not be feasible due to the extra degree of complexity implied.

Auto-ML for MLC (SÁ; PAPPA; FREITAS, 2017; WEVER *et al.*, 2019) can be used to find the best combination between strategies and base algorithm. Furthermore, it can tune the hyperparameters of both of them, as well as the pipeline of the solution, in order to bring the best results for a given problem. Thus, Auto-ML tools is an answer to the question of how to give advice which multi-label classifier and base algorithm to use. However, it demands high computational resources, which may be limiting its use.

Regarding the closely-related strategies (BR and pruned stacking; chaining; full stacking; and the ensembles investigated here), their differences are shown to be subtle and circumstantial. Given the relatively small number of data sets that have been considered in empirical studies, finding characteristics of a problem that distinguishes strategies is not a trivial task. Thus, the choice of a strategy between those close-related might also be seen as merely a matter of convenience, potentially influenced by other performance considerations, such as memory or runtime cost.

The differences between strategies from distinct groups are very consistent for the different evaluation criteria. Therefore, for empirical studies involving binary transformation strategies in MLC, we strongly recommend the use of strategies from different groups, as well as various base algorithms. The selection between strategies in the same group is not an easy task. However, it is important to provide some guidance concerning which one to use. We decided to

use the average ranking considering all base algorithms (Appendix A).

Table 9 summarises the experimental results, describing good strategies for different evaluation measures. In practical applications, RF and XGB should be considered as base algorithms, in addition to the usual favorites, which include C5.0, LR, and SVM. We note that if the median rank for each base algorithm or another criterion were adopted, different recommendations would probably be observed but the predicted performance obtained would not be expected to be very different.

Table 9 – Suggestion of binary transformation strategies to be picked in empirical experiments. The recommendation is based on criteria such as dissimilarity and the strategies' average ranking considering all base algorithms.

Measure	Ranking of suggested strategies				
	1	2	3	4	5
F1	EBR	MBR	RDBR	BR	CC
macro-F1	EBR	RDBR	MBR	CC	BR
macro-precision	MBR	NS	RDBR	BR	ECC
macro-recall	EBR	RDBR	MBR	CC	BR
hamming-loss	PruDent	BR	CC	BR+	EBR
one-error	BR	PruDent	NS	DBR	EBR
ranking-loss	BR	NS	PruDent	DBR	ECC
subset-accuracy	RDBR	NS	PruDent	BR	ECC

2.6 Conclusion

This paper presented an extensive experimental evaluation of binary transformation strategies for multi-label classification. Different perspectives were considered in addition to the traditional approach of selecting just a single base algorithm when comparing multi-label strategies. Thus, bipartition predictions were compared, strategies were compared for fixed base algorithms, base algorithm were compared for fixed strategies, and all possible pairs of strategy and base algorithm were compared with each other.

The main conclusions to draw from this study are:

- Binary transformation strategies are strongly influenced by the base algorithm used. Consequently, empirical studies should always consider distinct and diversified base algorithms.
- RF and XGB, which showed high predictive performance across a number of strategies, should be considered in the subset of base algorithms selected to perform an empirical study in MLC.
- The investigated strategies and base algorithms always either misclassified or were unable to predict some of the labels. So far this problem has been ignored, mainly because the traditional evaluation measures are not able to capture this problem. Nevertheless, this is a problem that requires more attention in future studies.

More specific conclusions for multi-label strategies and evaluation measures include:

- Ensembles using internal threshold selection obtained good results for *F1*, *macro-F1* and *macro-recall*.
- Despite being considered a baseline in many studies, BR obtained the best predictive performance for the ranking measures, *one-error* and *ranking-loss*. In addition, BR obtained good results for the *macro-precision* and *hamming-loss* measures, depending on the choice of base algorithm.
- The full stacking strategies and the NS strategy, which uses a subset correction procedure, obtained the best results for the *subset-accuracy* measure.

Future work includes investigating the impact of the base algorithm on other transformations such as the label-powerset method. Recommendation of combinations of a strategy and a base algorithm based on a desired measure, as well as dataset characteristics is another promising direction. Finally, the two types of label prediction failure, MLP and WLP, need to be researched in more depth.

CHAPTER

3

CHARACTERIZING CLASSIFICATION DATASETS: A STUDY OF META-FEATURES FOR META-LEARNING

Collaborating authors

Luís P. F. Garcia
Department of Computer Science, University of Brasilia, Brasilia, Brazil

Carlos Soares
Fraunhofer AICOS and LIAAD-INESC TEC, University of Porto, Porto, Portugal

Joaquin Vanschoren
Eindhoven University of Technology, Eindhoven, Netherlands

André C. P. L. F. de Carvalho
University of São Paulo, São Carlos, Brazil

Abstract

Meta-learning is increasingly used to support the recommendation of machine learning algorithms and their configurations. Such recommendations are made based on meta-data, consisting of performance evaluations of algorithms on prior datasets, as well as characterizations of these datasets. These characterizations, also called meta-features, describe properties of the data which are predictive for the performance of machine learning algorithms trained on them. Unfortunately, despite being used in a large number of studies, meta-features are not uniformly described, organized and computed, making many empirical studies irreproducible and hard to compare. This paper aims to deal with this by systematizing and standardizing

data characterization measures for classification datasets used in meta-learning. Moreover, it presents MFE, a new tool for extracting meta-features from datasets and identifying more subtle reproducibility issues in the literature, proposing guidelines for data characterization that strengthen reproducible empirical research in meta-learning.

3.1 Introduction

Machine learning algorithms have an inductive bias: they each make assumptions about the data distribution and choose specific generalization hypotheses over several other possible generalizations, thus restricting the search space (MITCHELL, 1997; WOLPERT, 1992). Since the true data distribution is unknown, several techniques are typically tried to achieve a satisfactory solution for a particular task. This trial-and-error approach is laborious and subjective, given the many choices that need to be made. Alternatively, meta-learning (MtL) presents a data-driven, automatic selection of techniques, by using knowledge extracted from previous tasks (BRAZDIL *et al.*, 2009). For instance, a meta-model can be trained on prior tasks to recommend suitable techniques for a new task (VANSCHOREN *et al.*, 2012).

Such a recommender system requires a systematic collection of dataset characteristics, along with the corresponding performance of different algorithms. These characteristics extracted from the datasets, named meta-features, play a crucial role in the successful use of MtL (BENSUSAN; KALOUSIS, 2001; BILALLI; ABELLÓ; ALUJA-BANET, 2017). Many empirical studies have investigated the effectiveness of meta-features in different domains (BENSUSAN; GIRAUD-CARRIER, 2000; BENSUSAN; KALOUSIS, 2001; FILCHENKOV; PENDRYAK, 2015; FÜRNKRANZ; PETRAK, 2001; PENG *et al.*, 2002b; PFAHRINGER; BENSUSAN; GIRAUD-CARRIER, 2000; REIF; SHAFAIT; DENGEL, 2011; REIF *et al.*, 2014), and proposed different sets of meta-features to characterize a given MtL task.

Unfortunately, several aspects that affect the reproducibility and generalizability of these experiments have been neglected or ignored in the literature. These include details concerning the dataset characterization process, hyperparameter settings used to evaluate algorithms, as well as procedures that deal with data encoding and missing values. These aspects require additional and careful investigation, especially given the current reproducibility crisis in machine learning research (HUTSON, 2018).

The lack of a systematic approach to compute meta-features has obfuscated the analyses in empirical MtL studies. To overcome this limitation, Pinto, Soares and Mendes-Moreira (2016b) proposed a framework to systematize the extraction of meta-features, defining a meta-feature in terms of three components: *meta-function*, *object* and *post-processing*. In short, a *meta-function* (e.g. entropy) extracts conceptual information from the *object* (e.g. predictive attributes) and a *post-processing* function (e.g. mean) summarizes the result. Different variations of these three components result in different meta-features. The authors claim that all current meta-features

can be decomposed using these three components. However, this framework does not directly mitigate the reproducibility problem, since the formalization, categorization and development of the meta-features are not addressed in the framework.

A good initiative to overcome this problem is OpenML (VANSCHOREN *et al.*, 2013), an on-line research platform that supports a standard characterization of datasets. As such, OpenML allows the comparison of MtL studies, insofar as they use the meta-features computed by OpenML. This set of meta-features is itself not defined systematically, however, which may hamper their suitability for subsequent meta-learning studies.

This paper surveys a comprehensive list of meta-features and their usage in the data classification MtL literature, and systematically organizes and categorizes these meta-features in a taxonomy. Furthermore, it highlights the main strengths and weaknesses of each meta-feature, identifying important reproducibility issues related to them. Finally, the paper presents the Meta-Feature Extractor (MFE) tool to compute many of these meta-features. Publicly available as a package in Python[1] and in R,[2] MFE offers a flexible and standalone implementation of meta-features for MtL experiments.

The rest of the paper is structured as follows. Section 3.2 presents a formalization and taxonomy for the meta-features assessed in this text. Section 3.3 presents a bibliographical synthesis that covers the state of the art in meta-features. Section 3.4 discusses the main strengths, weaknesses and open issues of the use of meta-features in MtL experiments. Section 3.5 discusses the main tools available and the MFE package. Section 3.6 concludes this work summarizing its main contributions and pointing out avenues for future research.

3.2 Taxonomy

Let $\mathscr{D}$ be a dataset with n instances, such that $\mathscr{D} = \{(\vec{x}_i, y_i) \mid 1 \leq i \leq n\}$. Each instance $\vec{x}_i = [v_{i1}, v_{i2}, \ldots, v_{id}]$ is a vector with d predictive attribute values, paired with a target value, y_i. A meta-feature f is a function $f : \mathscr{D} \to \mathbb{R}^k$ that, when applied to a dataset $\mathscr{D}$, returns a set of k values that characterize the dataset, and that are predictive for the performance of algorithms when they are applied to the dataset. Function f can be detailed as

$$f(\mathscr{D}) \;=\; \sigma(m(\mathscr{D},\, h_m),\, h_s),$$

such that $m : \mathscr{D} \to \mathbb{R}^{k'}$ is a characterization measure; $\sigma : \mathbb{R}^{k'} \to \mathbb{R}^k$ is a summarization function; h_m and h_s are hyperparameters used for m and σ, respectively. Note that k' can be different than k. The summarization function is required in propositional scenarios when a fixed cardinality k is needed, regardless of the value of k'.

[1] <https://pypi.org/project/pymfe/>
[2] <https://CRAN.R-project.org/package=mfe>

Table 10 – Categories used to describe a measure or group of measures.

Level	Category Name	Options
Input	Task	Classification Supervised Any
	Extraction	Direct Indirect
	Argument	n Predictive Attributes (nP) All Predictive Attributes (∗P) Target Attribute (T)
	Domain	Numerical Categorical Both
	Hyperparameters	Yes, No
Output	Range	[min, max]
	Cardinality	k
	Deterministic	Yes, No
	Exception	Yes, No

Traditionally, no distinction has been made between the concepts of a meta-feature, f, and a characterization measure, m. This may be natural when a measure results in a single value ($k' = k = 1$) and σ is the identity function, thus $f = m$. However, when a measure m can extract more than one value from each dataset, i.e. k' can vary according to $\mathscr{D}$, these values still need to be mapped to a vector of fixed length k. For instance, when a characterization can be computed per attribute (e.g. the mutual information between an attribute and the target) many authors use $f \approx mean(m)$ (ALI; SMITH, 2006; CASTIELLO; CASTELLANO; FANELLI, 2005; SOHN, 1999). Other common summarization functions are histograms (KALOUSIS; THEOHARIS, 1999), minimum and maximum (TODOROVSKI; BRAZDIL; SOARES, 2000), and skewness and kurtosis (REIF; SHAFAIT; DENGEL, 2012).

These definitions allow the categorization of meta-features in a well-defined taxonomy, illustrated in Table 10. In this framework, all characterization measures are themselves described in terms of their required *inputs* and their *outputs*. While some of these categories are only descriptive, others define whether or not a meta-feature is suitable for a specific scenario.

Some measures are restricted to specific tasks, such as *classification*. Others can be more generically applied to *supervised* tasks, which includes regression problems. The measures classified as *any* are the most general and can also be applied to unsupervised tasks such as clustering, and semi-supervised problems. In *supervised* tasks, a target attribute is required to evaluate the meta-features, which is not necessary for meta-features of the type *any*.

The cardinality defines the number of possible values returned by a measure. A distinction between single-valued measures ($k = 1$) and multi-valued measures ($k > 1$) is important for data

analysis, mainly to define whether or not a summarization function must be applied. For most of the multi-valued measures, the cardinality is related to aspects such as the number instances, attributes or classes in the considered datasets.

Some measures are *non-deterministic*, meaning that there is no guarantee that the same result will be obtained for the same input in different runs. When reproducibility is necessary, the same randomization seed must be used for each run or the measures must be executed a number of times and averaged to account for the randomization effect.

Finally, while some measures are *robust*, others can generate *exceptions* for certain datasets, leading them not to emit valid values in all cases. This can occur in particular conditions, such as a division by zero or a logarithm of a negative number. The proper handling of these situations is still an open issue for several measures.

3.3 Meta-Features

A fundamental question in MtL is how to extract suitable information to characterize specific tasks. Researchers have been trying to answer this question by looking for dataset properties that can affect learning algorithm performance, measuring this performance outright (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000; PFAHRINGER; BENSUSAN; GIRAUD-CARRIER, 2000), investigating alternatives (KOPF; TAYLOR; KELLER, 2000; SOARES; PETRAK; BRAZDIL, 2001) and adapting/creating new measures based on existing ones (CASTIELLO; CASTELLANO; FANELLI, 2005; SOHN, 1999).

In all cases, the meta-features were organized in groups. These groups are subsets of data characterization measures (BRAZDIL *et al.*, 2009) that share similarities among them. However, they are not always clearly and strictly delimited. Hence, when two different studies mention using a certain group of measures, it does not mean that they use exactly the same measures (SMITH-MILES, 2008). Additionally, different names have been used to describe these groups of measures. In this work, we propose to organize the measures in six groups:

Simple: measures that are easily extracted from data (REIF *et al.*, 2014), commonly known, and do not require significant computational resources (REIF, 2012). They are also called *general* measures (CASTIELLO; CASTELLANO; FANELLI, 2005).

Statistical: measures that capture the statistical properties of the data (REIF *et al.*, 2014). These measures capture data distribution indicators, such as average, standard deviation, correlation and kurtosis. They are computed on numerical attributes only (CASTIELLO; CASTELLANO; FANELLI, 2005).

Information-theoretic: measures from the information theory field (CASTIELLO; CASTELLANO; FANELLI, 2005). These measures are based on entropy (SEGRERA; LUCAS;

GARCÍA, 2008), capturing the amount of information in the data and their complexity (SMITH-MILES, 2008). They can be used to characterize discrete attributes.

Model-based: measures extracted from a model induced using the training data (REIF *et al.*, 2014). Many of these are based on properties of decision tree (DT) models (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000; PENG *et al.*, 2002b), referred to as *decision-tree-based* meta-features (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000). Properties extracted from other models are also used (FILCHENKOV; PENDRYAK, 2015).

Landmarking: measures that use the performance of simple and fast learning algorithms to characterize datasets (SMITH-MILES, 2008). The algorithms must have different inductive biases and should capture relevant information with a low computational cost (FÜRNKRANZ; PETRAK, 2001; SOARES; PETRAK; BRAZDIL, 2001).

Others: measures not included in the previous groups, such as standalone measures, time-related measures (REIF; SHAFAIT; DENGEL, 2011), concept and case-based measures (MUÑOZ *et al.*, 2018; VANSCHOREN *et al.*, 2012), clustering and distance based measures (PIMENTEL; CARVALHO, 2019; VUKICEVIC *et al.*, 2016), among others.

The first three groups represent the most common and traditional approaches for data characterization (BRAZDIL *et al.*, 2009). They receive different names such as *basic* measures (FILCHENKOV; PENDRYAK, 2015), *DCT* (PENG *et al.*, 2002b), *standard* (ENGELS; THEUSINGER, 1998) and *STATLOG* measures (SMITH-MILES, 2008). In earlier work, statistical measures were also called *discriminant* meta-features (LINDNER; STUDER, 1999). The next two groups depend on machine learning algorithms to extract model complexity or performance measures, while the last group includes characterizations for specific types of data, such as time series. Vanschoren (2010) offers a more fine-grained categorization of meta-features, based on intrinsic biases of learning algorithms, such as *data normality*, *feature redundancy*, and *feature-target association*.

In the remainder of this section, a systematic definition and description of these measures are provided, using the taxonomy shown in Table 10. The formal definition of each measure is available in Appendix C. In the descriptions, $-\infty$ and ∞ are used when it is not possible to define the range of a measure, whereas *inherited* is used when the measure range is defined by the value range of specific dataset attributes. The use of an upper stroke bar in the range and cardinality indicates an approximated value. When the columns *Extract*, *Domain*, *Hyperp.*, *Excep.* and *Det.* describe a constant property, they are suppressed from the tables and identified in the caption. The section finishes with a description and an analysis of the main summarization functions.

3.3.1 Simple meta-features

The simple measures, listed in Table 11, are directly extracted from the data and represent basic information about the dataset. They are the simplest set of measures in terms of definition and computational cost (CASTIELLO; CASTELLANO; FANELLI, 2005; MICHIE; SPIEGELHALTER; TAYLOR, 1994; REIF, 2012; REIF *et al.*, 2014). They are also deterministic and free of hyperparameters. Semantically, the measures are related to the number of predictive attributes, instances, target classes and missing values.

Table 11 – Simple meta-features and their characteristics. They are directly extracted, free of hyperparameter and deterministic.

Acronym	Task	Argument	Domain	Range	Card.	Excep.
attrToInst	Any	$*$P	Both	$[0, \overline{d}]$	1	No
catToNum	Any	$*$P	Both	$[0, \overline{d}]$	1	Yes
classToAttr	Classif.	$*$P+T	Both	$[0, q]$	1	No
freqClass	Classif.	T	Categ.	$[0, 1]$	q	No
instToAttr	Any	$*$P	Both	$[0, \overline{n}]$	1	No
instToClass	Any	$*$P+T	Both	$[1, \overline{n}]$	1	No
nrAttr	Any	$*$P	Both	$[1, +\infty]$	1	No
nrAttrMissing	Any	$*$P	Both	$[0, d]$	1	No
nrBin	Any	$*$P	Both	$[0, d]$	1	No
nrCat	Any	$*$P	Both	$[0, d]$	1	No
nrClass	Classif.	T	Categ.	$[2, \overline{n}]$	1	No
nrInst	Any	$*$P	Both	$[q, +\infty]$	1	No
nrInstMissing	Any	$*$P	Both	$[0, n]$	1	No
nrMissing	Any	$*$P	Both	$[0, \overline{dn}]$	1	No
nrNum	Any	$*$P	Both	$[0, d]$	1	No
numToCat	Any	$*$P	Both	$[0, \overline{d}]$	1	Yes

The measures related to attributes are: number of attributes (*nrAttr*); number of binary attributes (*nrBin*); number of categorical attributes (*nrCat*); number of numeric attributes (*nrNum*); proportion of categorical versus numeric attributes (*catToNum*) and vice-versa (*numToCat*). These measures are relevant to characterize the main aspects of a dataset, providing information that can support the choice of an algorithm for a particular learning task.

The number of instances (*nrInst*) and the number of classes (*nrClass*) indicate the dataset size and its label diversity. When combined with the *nrAttr*, we can define *attrToInst* and *instToAttr*, which represent the dimensionality and sparsity of the data, respectively. The latter is a potential indicator for overfitting when its value is too small (KUBA *et al.*, 2002). The number of classes per attribute (*classToAttr*) and instances per classes (*instToClass*) measure properties of the target attribute distribution, such as class imbalance. Likewise, the frequency of instances in each class (*freqClass*) allows the extraction of measures such as the proportional frequency of the majority and minority class (ALI; SMITH, 2006), default accuracy/error (PENG *et al.*, 2002b) and standard deviation of the class distribution (LINDNER; STUDER, 1999). When combined with summarization functions, it can describe imbalanced learning scenarios.

Finally, some measures assess dataset quality, such as the number of attributes (*nrAt-*

trMissing) and instances (*nrInstMissing*) with missing values, as well as the total number (*nrMissing*). Since some ML algorithms can deal with missing values better than others, these measures can provide important information for algorithm selection, or indicate when data treatment is necessary, as discussed in Section 3.4.1.

Some authors have proposed modified versions of these measures. For instance, Todorovski, Brazdil and Soares (2000) use the log of the number of instances, Brazdil, Gama and Henery (1994) use the proportion of categorical attributes, and Kalousis and Hilario (2001b) use the proportion of numerical attributes. When using these measures for meta-learning, it may be necessary to normalize the values of these measures. With the exception of *nrAttr* and *nrInst*, the measures can be normalized using their theoretical maximum value, as shown in the column Range.

Finally, note that the *catToNum* and *numToCat* measures can only be computed for datasets which have both categorical and numeric attributes.

3.3.2 Statistical meta-features

Statistical measures can extract information about the performance of statistical algorithms (MICHIE; SPIEGELHALTER; TAYLOR, 1994) or about data distribution, for instance, central tendency and dispersion (CASTIELLO; CASTELLANO; FANELLI, 2005). They are the largest and the most diversified group of meta-features, as shown in Table 12. Statistical measures are deterministic and support only numerical attributes. Some measures require the definition of hyperparameter values, while others can generate exceptions, e.g. caused by division by zero. Some of them are indirectly extracted, and are closely related to the discriminant group reported in Lindner and Studer (1999). The others can be widely applied since they use only predictive attributes as input.

Correlation (*cor*) and covariance (*cov*) capture the interdependence of the predictive attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994). They are computed for each pair of attributes in the dataset, resulting in $(d-1)/2$ values. The former is a normalized version of the latter, and the absolute value of both measures are frequently used, which changes the range from $[-1,1]$ and $[-\infty,\infty]$, respectively, to the values reported in Table 12. High values indicate a strong correlation between the attributes, which can be interpreted as a level of redundancy in the data (KALOUSIS; HILARIO, 2001b). To represent this information, *nrCorAttr* computes the proportion of highly correlated attribute pairs.

Most statistical measures are extracted for each attribute separately. Measures of central tendency are composed by the *mean* and its variations such as the geometric mean (*gMean*), harmonic mean (*hMean*) and trimmed mean (*tMean*); and the *median*. Measures of dispersion consist of the interquartile range (*iqRange*), *kurtosis*, maximum (*max*), median absolute deviation (*mad*), minimum (*min*), *range*, standard deviation (*sd*), *skewness* and variance (*var*). While one

Table 12 – Statistical meta-features and their characteristics. They are deterministic and only accept numerical attributes.

Acronym	Task	Extract	Argument	Hyperp.	Range	Card.	Excep.
canCor	Classif.	Indirect	∗P+T	No	$[0,1]$	$\overline{d}$	No
cor	Any	Direct	2P	Yes	$[0,1]$	$\overline{d^2}$	Yes
cov	Any	Direct	2P	No	$[0,\infty]$	$\overline{d^2}$	No
nrDisc	Classif.	Indirect	∗P+T	No	$[0,d]$	1	No
eigenvalues	Any	Indirect	∗P	No	$[0,\infty]$	$\overline{d}$	No
gMean	Any	Direct	1P	No	$[0,\infty]$	d	Yes
hMean	Any	Direct	1P	No	*inherited*	d	No
iqRange	Any	Direct	1P	No	$[0,\infty]$	d	No
kurtosis	Any	Direct	1P	No	$[-3,\infty]$	d	Yes
mad	Any	Direct	1P	No	$[0,\infty]$,	d	No
max	Any	Direct	1P	No	*inherited*	d	No
mean	Any	Direct	1P	No	*inherited*	d	No
median	Any	Direct	1P	No	*inherited*	d	No
min	Any	Direct	1P	No	*inherited*	d	No
nrCorAttr	Any	Direct	∗P	Yes	$[0,1]$	1	Yes
nrNorm	Any	Direct	∗P	Yes	$[0,d]$	1	No
nrOutliers	Any	Direct	∗P	Yes	$[0,d]$	1	No
range	Any	Direct	1P	No	$[0,\infty]$	d	No
sd	Any	Direct	1P	No	$[0,\infty]$	d	No
sdRatio	Classif.	Indirect	∗P+T	No	$[1,\infty]$	1	Yes
skewness	Any	Direct	1P	No	$[-\infty,\infty]$	d	Yes
tMean	Any	Direct	1P	Yes	*inherited*	d	No
var	Any	Direct	1P	No	$[0,\infty]$	d	No
wLambda	Classif.	Indirect	∗P+T	No	$[0,1]$	1	No

points to the center of a distribution, the other shows how much the values are spread from the center, complementing themselves. Their range depends directly on the attributes' range, with few exceptions like kurtosis and skewness. These two, are suitable to capture the normality of the data attributes (VANSCHOREN, 2010).

A specific measure to capture the normality of the attributes is the *nrNorm*, which computes the number of attributes normally distributed. Similarly, *nrOutliers* counts the number of attributes that contain outliers. Normality and outliers may impact the behavior of learning algorithms, which make these measures useful in an MtL scenario.

The discriminant statistical measures present some specificities such as being exclusively used for classification tasks. By considering the target value and using the whole dataset as input, they result in a single value. Canonical correlations (*canCor*), the number of discriminant values (*nrDisc*), the homogeneity of covariances (*sdRatio*) and the Wilks lambda (*wLambda*) represent the discriminant measures. Finally, the *eigenvalues* from the covariance matrix only use the predictive data to be computed.

Concerning the hyperparameters, different correlation methods such as Pearson's correlation, Kendall's τ and Spearman's ρ coefficient (RODGERS; NICEWANDER, 1988), can be used to compute the *cor* measure. This is also applied to the *nrCorAttr* measure, which additionally

requires a threshold value to define high correlations. The *tMean* requires the definition of how much data should be discarded to compute the mean. Finally, the *nrNorm* and *nrOutliers* are dependent on the algorithm to compute whether or not a distribution is normal and has outliers. Even though *skewness* and *kurtosis* could be seen as algorithm dependent, their variations do not produce observable differences for large samples of data (JOANES; GILL, 1998).

Some measures can throw exceptions and due to this are not calculated correctly. The *cor*, *kurtosis*, *nrCorAttr* and *skewness* could generate an error with a constant attribute caused by division by zero. The *sdRatio* uses *log* in this formulation, and the possibility of obtaining a negative value makes the measure error-prone. The *gMean* can be computed in 2 different ways and both can generate errors, one using product and another using *log*. The former can obtain arithmetic overflow/underflow while the latter cannot support negative values.

As the majority of the statistical measures do not consider the class information, Castiello, Castellano and Fanelli (2005) proposed an indirect way to explore it. This approach splits the dataset according to the class labels and computes the measures for each subset. However, the authors are not aware of any empirical evaluation of this approach. Besides, many statistical measures need to be summarized since several possible values can be obtained. Finally, it is important to observe that the statistical measures only support numerical attributes. Datasets that contain categorical data must be either partially ignored or converted to numerical values.

3.3.3 Information-Theoretic meta-features

Information-theoretic meta-features capture the amount of information in the data. Table 13 shows the information-theoretic measures, which require categorical attributes and most of them are restricted to representing classification problems. Moreover, they are directly computed, free of hyperparameter, deterministic and robust. Semantically, they describe the variability and redundancy of the predictive attributes to represent the classes.

Table 13 – Information-theoretic meta-features and their characteristics. They are directly extracted, free of hyperparameter, robust, deterministic and support only categorical attributes.

Acronym	Task	Argument	Range	Card.
attrEnt	Any	1P	$[0, \log_2(n)]$	d
classEnt	Classif.	T	$[0, \log_2(q)]$	1
eqNumAttr	Classif.	$*$P+T	$[0, \infty]$	1
jointEnt	Classif.	1P+T	$[0, \log_2(n)]$	d
mutInf	Classif.	1P+T	$[0, \log_2(n)]$	d
nsRatio	Classif.	$*$P+T	$[0, \infty]$	1

The entropy of the predictive attributes (*attrEnt*) and the target values (*classEnt*) capture the average uncertainty present in the predictive and class attributes (SEGRERA; LUCAS; GARCÍA, 2008), respectively. In the former, all predictive attributes are assessed, thus its summarization can provide an overview of the attributes' capacity for class discrimination.

In the latter, it represents how much information, on average, is necessary to specify one class (CASTIELLO; CASTELLANO; FANELLI, 2005). In a learning perspective, a predictive attribute with a low entropy contains a low discriminatory power (MICHIE; SPIEGELHALTER; TAYLOR, 1994), whereas a target attribute with low entropy contains a high level of purity. These measures are usually normalized.

The joint entropy (*jointEnt*) and the mutual information (*mutInf*) compute the relationship of each attribute with the target values. While the former captures the relative importance of the predictive attributes to represent the target (ENGELS; THEUSINGER, 1998), the latter represents the common information shared between them, indicating their degree of dependency (MICHIE; SPIEGELHALTER; TAYLOR, 1994).

Finally, the equivalent number of attributes (*eqNumAttr*) and the noise signal ratio (*nsRatio*) capture information that is related to the minimum number of attributes necessary to represent the target attribute and the proportion of data that are irrelevant to describe the problem (SMITH *et al.*, 2001), respectively.

To extract these measures from numerical attributes, we must know their data distribution or discretize them (CASTIELLO; CASTELLANO; FANELLI, 2005). The latter is simpler. However, being user-defined needs the introduction of hyperparameters, which is discussed further in Section 3.4.1.

3.3.4 Model-Based meta-features

The meta-features of this group are information extracted from a predictive learning model, in particular, a DT model. They characterize a dataset by how complex is the model induced, which, for DT, can be the number of leaves, the number of nodes and the shape of the tree. Table 14 shows the DT model meta-features. They are designed to characterize supervised problems, all measures are deterministic, robust and require the definition of hyperparameters: the DT induction algorithm (together with its hyperparameter values) used to induce the DT model.

The measures based on leaves are identified with the prefix *leaves*, which describe, in some degree, the complexity of the orthogonal decision surface. Some measures result in a value for each leaf, and those measures are the number of distinct paths (*leavesBranch*), the support described in the proportion of training instances to the leaf (*leavesCorrob*) and the distribution of the leaves in the tree (*leavesHomo*).

The proportion of leaves to the classes (*leavesPerClass*) represents the classes complexity and the result is summarized per class. While *leavesCorrob* and *leavesPerClass* have a fixed range independent of the dataset, *leaves* and *leavesBranch* have a maximum value limited by the number of instances. In practice, the most observed limit is associated with the number of attributes, which also determines the cardinality of them. Only *leavesHomo* does not have a

Table 14 – Model-based meta-features and their characteristics. These meta-features are indirectly extracted, robust, deterministic, require the definition of hyperparameters and support both attribute types.

Acronym	Task	Argument	Range	Card.
leaves	Sup.	∗P+T	$[q, \overline{n}]$	1
leavesBranch	Sup.	∗P+T	$[1, \overline{n}]$	$\overline{n}$
leavesCorrob	Sup.	∗P+T	$[0, 1]$	$\overline{n}$
leavesHomo	Sup.	∗P+T	$[q, +\infty]$	$\overline{n}$
leavesPerClass	Classif.	∗P+T	$[0, 1]$	q
nodes	Sup.	∗P+T	$[q, \overline{n}]$	1
nodesPerAttr	Sup.	∗P+T	$[0, \overline{n}]$	1
nodesPerInst	Sup.	∗P+T	$[0, 1]$	1
nodesPerLevel	Sup.	∗P+T	$[1, \overline{n}]$	$\overline{n}$
nodesRepeated	Sup.	∗P+T	$[0, \overline{n}]$	$\overline{d}$
treeDepth	Sup.	∗P+T	$[1, \overline{n}]$	$\overline{n}$
treeImbalance	Sup.	∗P+T	$[0, 1]$	$\overline{n}$
treeShape	Sup.	∗P+T	$[0.0, 0.5]$	$\overline{n}$
varImportance	Sup.	∗P+T	$[0, 1]$	$\overline{d}$

defined limit of values.

The measures based on nodes, which extract information about the balance of the tree to describe the discriminatory power of attributes, are identified with the prefix *nodes*. Together with *nodes*, the proportion of nodes per attribute (*nodesPerAttr*) and the proportion of nodes per instance (*nodesPerInst*) result in a single value. The number of nodes per level (*nodesPerLevel*) and the number of repeated nodes (*nodesRepeated*) have the number of attributes at their maximum value. While *nodesPerLevel* describes how many nodes are present in each level, *nodesRepeated* represents the number of nodes associated with each attribute used for the model.

The measures based on the tree size, which extract information about the leaves and nodes to describe the data complexity, are identified with the prefix *tree*. The tree depth (*treeDepth*) represents the depth of each node and leaf, the tree imbalance (*treeImbalance*) describes the degree of imbalance in the tree and the shape of the tree (*treeShape*) represents the entropy of the probabilities to randomly reach a specific leaf in a tree from each one of the nodes.

Finally, the importance of each attribute (*varImportance*) represents the amount of information present in the attributes before a node split operation. The amount of information is defined by the randomization of incorrect labeling. This measure varies according to the DT algorithm. As an example, the C4.5 algorithm uses the information gain from the information-theoretic group to compute the importance of the attributes (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000) and the CART algorithm uses the Gini index (LOH, 2014).

Other model-based measures, using different learners, such as k-Nearest Neighbors (kNN) and Perceptron neural networks were presented in Filchenkov and Pendryak (2015). However, some of these measures have a very high computational cost. Some others have the concept already described by well-known groups. In Nguyen *et al.* (2012), the weights learned

by distinct feature selection algorithms were defined as model-based meta-features.

3.3.5 Landmarking meta-features

Landmarking is an approach that characterizes datasets using the performance of a set of fast and simple learners, different from the model-based meta-features, which extract information from the learning models. Although the performance of any algorithm can be used as a landmarking, including sophisticated algorithms, some of them have been specifically used as meta-features. Table 15 lists the most common landmarking measures. They characterize supervised problems and are indirectly extracted, thus the whole dataset is used as an argument. They require the definition of hyperparameters: the learning algorithm; the evaluation measure to asses the model performance; and, the procedure used to compute them (e.g. cross-validation). While the range is dependent on the evaluation measure (usually between 0 and 1), the cardinality is from the procedure, thereby it is user-defined. Since their training and test data samples are randomly chosen, all landmarking are non-deterministic.

Table 15 – Common landmarking meta-features and their characteristics. They are indirectly extracted, non-deterministic and require the definition of hyperparameters.

Acronym	Task	Argument	Domain	Range	Card.	Excep.
bestNode	Sup.	∗P+T	Both	$[0,1]$	*user*	No
eliteNN	Sup.	∗P+T	Both	$[0,1]$	*user*	No
linearDiscr	Sup.	∗P+T	Num.	$[0,1]$	*user*	Yes
naiveBayes	Sup.	∗P+T	Both	$[0,1]$	*user*	No
oneNN	Sup.	∗P+T	Both	$[0,1]$	*user*	No
randomNode	Sup.	∗P+T	Both	$[0,1]$	*user*	No
worstNode	Sup.	∗P+T	Both	$[0,1]$	*user*	No

The measures *bestNode*, *randomNode* and *worstNode* are the performance of a DT-model induced using different single attributes. Respectively, they use the following attributes: the most informative, a random one, and the least informative attribute. The aim is to capture information about the boundary of the classes and combine this information with the linearity of the DT-models induced with the worst and random attributes. The DT algorithm is a hyperparameter defined by the user since different algorithms could be used.

The elite-Nearest Neighbor (*eliteNN*) is the result of the 1NN model using a subset of the most informative attributes in the dataset, whereas the one-Nearest Neighbor (*oneNN*) is the result of a similar learning model induced with all attributes. The distance measure used by the kNN algorithm is a hyperparameter.

The Linear Discriminant (*linearDiscr*) and the Naive Bayes (*naiveBayes*) algorithms use all attributes to induce the learning models. The first technique finds the best linear combination of predictive attributes able to maximize the separability between the classes. For such, it uses a covariance matrix and assumes that the data follow a Gaussian distribution. This technique can generate exceptions if the data has redundant attributes. The second technique is based on the

Bayes' theorem and calculates, for each feature, the probability of an instance to belong to each class. The combination of all features and related probabilities for one instance returns the class with the highest probability.

Concerning the hyperparameters, an evaluation measure such as accuracy, balanced accuracy and Kappa is necessary to evaluate the models. Other measures such as precision, recall and F1 also could be used, however, for them, it is necessary to identify the class of interest in binary datasets. The procedures used to induce the model are *(i)* using the whole instances to train and test; *(ii)* holdout; and, *(iii)* cross validation. This information is rarely mentioned in MtL studies and their impact in the characterization measures are not yet known. In practice, it represents a trade-off between stable measures and computational costs.

Some variants are relative and subsampling landmarkings (FÜRNKRANZ; PETRAK, 2001) and their combined use (SOARES; PETRAK; BRAZDIL, 2001). Instead of using the absolute performance of the landmarkers as meta-features, a relative approach adopts the landmarkers' ranking, which is obtained using pairwise comparisons. Thus, the meta-feature can be a binary value indicating the winner, the difference between them or the ratio of the two performances. Besides, a meta-feature for each ranking position containing the name of the respective landmarker is of categorical type. On the other hand, subsampling landmarking works by applying traditional algorithms to a reduced subset of the original dataset.

The performance of the landmarkers can be represented as a learning curve, representing their use with different sampling sizes of a dataset (LEITE; BRAZDIL, 2005). Furthermore, in an algorithm recommendation scenario, their relative performance can be learned by meta-models and the prediction from these meta-models can be used as meta-features, analogous to a stacking-based approach (SUN; PFAHRINGER, 2013).

3.3.6 Other meta-features

Many other non-traditional characterization measures have been reported in the literature. Despite the fact they are not broadly used in MtL studies, e.g. due to a high computational complexity or domain bias, they can be useful for a particular learning scenario and MtL problem. Besides, some works show good results when using those characterization measures (GARCIA *et al.*, 2018; MORAIS; PRATI, 2013; PIMENTEL; CARVALHO, 2019). Here, they are arbitrarily presented in the following subgroups: *clustering and distance*; *complexity*; and *miscellaneous*.

3.3.6.1 Clustering and distance-based

Clustering and distance-based measures characterize the instance space using validation, also called index, measures that evaluate partitions produced by clustering algorithms and measures calculating the distance between instances. Clustering validation measures and distance-

based measures can be indirectly extracted characterization measures, requiring the set of hyperparameter values such as the clustering algorithm and the distance function, respectively. With few exceptions, they are computed using only the predictive attributes. According to the distance measure used, the meta-features can handle numerical and/or categorical attributes. Table 16 presents a list of clustering and distance-based measures.

Table 16 – Clustering and distance-based meta-features and their characteristics. They are robust and require the definition of hyperparameters.

Acronym	Task	Extract	Argument	Domain	Range	Card.	Determ.
AIC	Any	Indirect	∗P	Both	$[0,\infty]$	1	No
BIC	Any	Indirect	∗P	Both	$[0,\infty]$	1	No
compactness	Any	Indirect	∗P	Both	$[0,\infty]$	$\overline{n}$	No
connectivity	Any	Indirect	∗P	Both	$[0,n]$	1	No
distInst	Any	Direct	∗P	Both	$[0,\infty]$	$\overline{n^2}$	Yes
distCorrInst	Any	Direct	∗P	Num.	$[0,1]$	$\overline{n^2}$	Yes
gravity	Classif.	Indirect	∗P+T	Both	$[0,\infty]$	1	Yes
nrClusters	Any	Indirect	∗P	Both	$[1,\overline{n}]$	1	No
purityRatio	Classif.	Indirect	∗P+T	Both	$[0,1]$	q	No
silhouette	Any	Indirect	∗P	Both	$[-1,1]$	1	No
sizeDist	Any	Indirect	∗P	Both	$[0,1]$	$\overline{n}$	No
XB	Any	Indirect	∗P	Both	$[0,\infty]$	1	No

Given the data partition produced by a clustering algorithm, *nrCluster* represents the number of clusters, a simple informative measure, which is useful when this number is dynamically defined. When the clustering algorithm used has the number of clusters as a hyperparameter, a common option is to use the number of classes. The distribution of the clusters based on the instances' frequency is captured by the measure *sizeDist*. A distribution skewed to the right indicates a complex dataset (LER *et al.*, 2018).

Different validation measures are used to represent the quality of the partitions obtained, such as how compact each group is and how separated the groups are from each other (VUKICEVIC *et al.*, 2016). In a classification context, this information may indicate the separability of the instances, and possibly the classes. The Akaike Information Criterion (*AIC*) and the Bayesian Information Criterion (*BIC*) measures represent the relative quality of the partitions by estimating the amount of information lost by the model used to define the clusters. For both, lower values indicate a better generalization of a model. While *Compactness* measures how compact the clusters are, *Silhouette* and the Xie-Beni index (*XB*) add separation between clusters to the compactness. The lower the value, the better for *Compactness*, whereas, for *Silhouette* and *XB*, it is the opposite. Other often-used clustering validation measures are presented next.

Connectivity captures local densities by counting violations of the nearest neighbor relationship of instances in different partitions (HANDL; KNOWLES; KELL, 2005). When normalized by the number of instances, high values indicate that the clusters are not well separated. It could be an informative measure to characterize the suitability of the bias related to instance-based learning algorithms.

Although only these validation measures have been used to characterize datasets in MtL studies, there are many other clustering internal validation measures (HANDL; KNOWLES; KELL, 2005) that could be employed. These measures can also be used without a clustering algorithm, by considering the classes as partitions.

Differently, *purityRatio* is a clustering measure that looks at the instances' classes to evaluate the partitions. It is calculated for each class and captures the ratio of clusters that contain instances related to the respective class. Datasets with high values are more complex than those with low values since the classes are distributed across all partitions.

Another subset of measures, the distance-based measures (PIMENTEL; CARVALHO, 2019) are obtained computing the distance between all pairs of instances (*distInst*) and the correlations combined with the distances (*distCorrInst*). They indicate how close and related pairs of instances are, which may influence the decision boundaries of learning algorithms. Finally, the center of gravity (*gravity*) computes the dispersion among the groups of instances according to their class label. In this case, the groups are defined by the classes.

With few exceptions, all these measures have a high asymptotic computational complexity, which restricts their use. Additionally, they allow a wide number of choices, with different impacts in the value returned. In spite of being able to provide a good characterization, clustering and distance measures are underexplored in the MtL literature.

3.3.6.2 Complexity

Complexity measures were proposed in (HO; BASU, 2002) to capture the underlying difficulty of classification tasks, considering aspects such as class overlapping, the density of manifolds and the shape of decision boundaries. They were used to support data pre-processing, machine learning and recommender systems (GARCIA; CARVALHO; LORENA, 2016; GARCIA *et al.*, 2018; LUENGO; HERRERA, 2015; SMITH; MARTINEZ; GIRAUD-CARRIER, 2014). While the complete survey of the complexity measures can be found in Lorena *et al.* (2019), Table 17 summarizes the main characteristics of these measures.

Ho and Basu (2002) divide the complexity measures into three groups: (i) feature overlapping measures; (ii) measures of the separability of classes; and (iii) geometry, topology and density of manifolds measures. Following Lorena *et al.* (2019), we adopted a more granular organization: (i) feature-based measures; (ii) linearity measures; (iii) neighborhood measures; (iv) network measures; and (v) dimensionality measures.

Feature overlapping measures characterize how informative the predictive attributes are to separate the classes. They are: maximum Fisher's discriminant ratio (*F1*); directional-vector maximum Fisher's discriminant ratio (*F1v*); volume of overlapping region (*F2*); maximum individual feature efficiency (*F3*); collective feature efficiency (*F4*). The complexity is low if at least one predictive attribute can separate the classes.

Table 17 – Complexity meta-features and their characteristics. They are robust measures.

Acronym	Task	Extract	Argument	Domain	Hyperp.	Range	Card.	Determ.
clsCoef	Classif.	Indirect	∗P+T	Num.	Yes	$[0,1]$	1	Yes
graphDensity	Classif.	Indirect	∗P+T	Num.	Yes	$[0,1]$	1	Yes
F1	Classif.	Direct	1P+T	Both	No	$[0,1]$	1	Yes
F1v	Classif.	Indirect	∗P+T	Both	No	$[0,1]$	1	Yes
F2	Classif.	Direct	1P+T	Num.	No	$[0,1]$	1	Yes
F3	Classif.	Direct	1P+T	Num.	No	$[0,1]$	1	Yes
F4	Classif.	Direct	∗P+T	Num.	No	$[0,1]$	1	Yes
Hubs	Classif.	Indirect	∗P+T	Num.	Yes	$[0,1]$	1	Yes
LSC	Classif.	Direct	∗P+T	Num.	No	$[0,1-\frac{1}{n}]$	1	Yes
L1	Classif.	Indirect	∗P+T	Num.	No	$[0,1]$	1	Yes
L2	Classif.	Indirect	∗P+T	Num.	No	$[0,1]$	1	Yes
L3	Classif.	Indirect	∗P+T	Num.	No	$[0,1]$	1	No
N1	Classif.	Indirect	∗P+T	Num.	No	$[0,1]$	1	Yes
N2	Classif.	Direct	∗P+T	Both	No	$[0,1]$	1	Yes
N3	Classif.	Direct	∗P+T	Both	No	$[0,1]$	1	Yes
N4	Classif.	Direct	∗P+T	Both	No	$[0,1]$	1	Yes
T1	Classif.	Direct	∗P+T	Num.	No	$[0,1]$	1	Yes
T2	Any	Direct	∗P	Both	No	$[0,\overline{n}]$	1	Yes
T3	Any	Indirect	∗P	Num.	No	$[0,\overline{n}]$	1	Yes
T4	Any	Indirect	∗P	Num.	No	$[0,1]$	1	Yes

Linearity measures quantify whether the classes are linearly separated. They include sum of the error distance by linear programming (*L1*); error rate of linear classifier (*L2*); non-linearity of a linear classifier (*L3*). To obtain the linear classifier, a linear Support Vector Machine (SVM) is often used.

Neighborhood measures analyze the neighborhoods of individual examples and try to capture class overlap and the shape of the decision boundary. They include fractions of Borderline Points (*N1*); ratio of intra/extra class nearest neighbor distance (*N2*); error rate of the nearest neighbor classifier (*N3*); non-Linearity of the nearest neighbor classifier (N4); fraction of hyperspheres covering data (*T1*); local set average cardinality (*LSC*). All of them use a distance matrix between all pairs of points in the dataset to define the instances' neighborhoods according to their classes.

The network measures transform a dataset into a graph and extract structural and statistical information from the graph. In this new representation, each example from the dataset corresponds to a node, whilst undirected edges connect pairs of examples and are weighted by the distances between them. These measures include average density of the network (*graphDensity*) and Hub score (*hubs*). Other complex network measures are presented by Morais and Prati (2013), however they are not detailed and we did not find other works using them.

Finally, the dimensionality measures evaluate data sparsity according to the number of instances relative to the predictive attributes of the dataset. The measures include the average number of points per dimension (*T2*); the average number of points per Principal Component Analysis (PCA) dimension (*T3*); the ratio of the PCA dimension to the original dimension (*T4*).

While *T2* is the *instToAttr* meta-features, the *T3* and *T4* differ from *T2* by using a transformed dataset instead of the original.

These complexity measures look at different complexity aspects in a dataset. Thus, they can be related to other groups of measures presented in this study. A variation of them to characterize the classes individually instead of the whole dataset is found in Barella *et al.* (2018). They are appropriate to represent the complexity of imbalanced datasets. These complexity measures are free of hyperparameters and do not require the use of summarization functions, since some of them directly adopt a summarization procedure, e.g. *F1* which uses the maximum value. However, their extraction usually has a high computational cost, which restricts their use in MtL studies.

3.3.6.3 Miscellaneous

In this section, we included other characterization measures found in our review, which did not fit in the previous groups and were used in a small number of MtL studies. These measures are summarized in Table 18.

Table 18 – Other miscellaneous meta-features and their characteristics. They are robust measures.

Acronym	Task	Extract	Argument	Domain	Hyperp.	Range	Card.	Determ.
Data distribution measures								
attrConc	Any	Direct	2P	Categ.	No	$[0,1]$	$\overline{d^2}$	Yes
classConc	Classif.	Direct	1P+T	Categ.	No	$[0,1]$	d	Yes
propPCA	Any	Indirect	$*$P	Num.	Yes	$[0,1]$	1	Yes
sparsity	Any	Direct	1P	Both	No	$[0,1]$	d	Yes
Case base measures								
consistencyRatio	Supervised	Direct	$*$P+T	Both	No	$[0,1]$	1	Yes
incoherenceRatio	Any	Direct	$*$P	Both	Yes	$[0,1]$	1	Yes
uniquenessRatio	Any	Direct	$*$P	Both	No	$[0,1]$	1	Yes
Concept based measures								
cohesiveness	Classif.	Direct	$*$P+T	Both	Yes	$[0,\overline{n}]$	n	Yes
wgDist	Any	Direct	$*$P	Both	Yes	$[0,\infty]$	n	Yes
Structural Information								
oneItemset	Any	Indirect	$*$P	Both	No	$[0,1]$	d	Yes
twoItemset	Any	Indirect	$*$P	Both	No	$[0,1]$	$\overline{d^2}$	Yes
Time based measures								
infotheoTime	Any	Indirect	$*$P	Categ.	No	$[0,\infty]$	1	No
landTime	Supervised	Indirect	$*$P+T	Both	No	$[0,\infty]$	$\overline{7}$	No
modelTime	Supervised	Indirect	$*$P+T	Both	No	$[0,\infty]$	1	No
statTime	Any	Indirect	$*$P	Num.	No	$[0,\infty]$	1	No

Data distribution measures assess how the data is distributed in the predictive attribute space. One of these measures is the concentration coefficient, also known as *Goodman and Kruskal's* τ (KALOUSIS; HILARIO, 2001b), which is applied to each pair of attributes (*attrConc*) and to each attribute and the class (*classConc*). In the former $d(d-1)$ values are obtained,

since it is not symmetric, whereas in the latter, d values are obtained, given that each attribute is associated with the class. Semantically, they represent the association strength between the attributes in each pair of attributes and between each predictive attribute and the target attribute.

Other related measures are the proportion of principal components that explain a specific (e.g. 95%) variance of the dataset (*propPCA*) and the *sparsity*, which extracts the degree of discreetness in each attribute. The former is another measure for capturing the redundancy of predictive attributes, whereas, the latter indicates the variance in the values of the attributes.

Case base measures compare the instances with each other to identify properties that might make the learning process more difficult (KOPF; IGLEZAKIS, 2002). Most of them are originally proposed as logical measures, however, instead of only capturing the occurrence (or not) of each property, we propose small changes to quantify each occurrence. The *consistencyRatio* quantifies the proportion of repeated instances with different targets, where zero is an ideal value. The *uniquenessRatio* is a generalization of *consistencyRatio*, since it uses only the predictive attributes. To measure how dissimilar the instances are in their attribute space, *incoherenceRatio* computes the proportion of instances that do not overlap with any other instances in a predefined number of attributes. Values close to 1 are preferred in a dataset since it shows that the instances are scattered through the input space.

The concept-based measures characterize the sparsity and the irregularity of the input-output distribution (VILALTA; DRISSI, 2002a). An irregular distribution is observed when neighboring instances have distinct target values (MUÑOZ *et al.*, 2018). The weighted distance (*wgDist*) captures how dense or sparse the distribution of the instances is (VILALTA, 1999). It could be defined as a distance-based measure. *Cohesiveness* measures the density of the example distribution (VANSCHOREN, 2010). Another measure of this subgroup, the concept variation (VILALTA; DRISSI, 2002a) is defined by the cohesiveness average of all possible instances in the input space, therefore unfeasible. Its version using the existing instances is captured by the summarization function *mean*.

Structural information works well in identifying similar datasets (WANG; SONG; ZHU, 2015), by characterizing binary itemsets to capture the distribution of values of both single attributes (*oneItemset*) and pairs of attributes (*twoItemset*) (SONG; WANG; WANG, 2012). They capture different and complementary aspects of the dataset. *oneItemset* captures information of each individual's attributes, whereas, *twoItemset* captures possible correlations concerning pairs of attributes. Association rules can also be applied to the transformed dataset to characterize other relations between attributes (BURTON *et al.*, 2014; MUÑOZ *et al.*, 2018).

Time-based measures comprise the elapsed time to compute the previous groups of measures (REIF; SHAFAIT; DENGEL, 2011), such as statistical, information-theoretic, model-based and landmarking. In this case, the same hardware should be used to compute the meta-features from different datasets, which can be very restrictive. Another option is to use the number of float point operations, but it is not always possible.

3.3.7 Summarization Functions

In this study, the purpose of summarization functions is to normalize the cardinality of meta-features and to characterize other meta-feature aspects, such as tendency, distribution and variability of the results. Given that many measures are multi-valued and that their cardinalities vary according to the dataset, comparisons between multiple datasets can be infeasible. Consequently, the summarization transforms non-propositional data to propositional (TODOROVSKI; BRAZDIL; SOARES, 2000), making them suitable to be organized in a meta-base, for instance. In the literature, summarization functions have been called meta-level attributes (TODOROVSKI; BRAZDIL; SOARES, 2000), meta^2-features (REIF; SHAFAIT; DENGEL, 2012) and post-processing functions (PINTO; SOARES; MENDES-MOREIRA, 2016b).

It is worth noting that in some studies (CASTIELLO; CASTELLANO; FANELLI, 2005; FILCHENKOV; PENDRYAK, 2015; KUBA *et al.*, 2002), to cite a few, the mean function is used as part of the meta-feature definition and it is the only way used to summarize the results. Other studies have used distinct subsets of summarization functions, such as histogram (KALOUSIS; THEOHARIS, 1999); minimum, mean and maximum (TODOROVSKI; BRAZDIL; SOARES, 2000); minimum, maximum, mean and standard deviation (FEURER; SPRINGENBERG; HUTTER, 2014; GARCIA; CARVALHO; LORENA, 2016; PENG *et al.*, 2002a); mean, standard deviation and quartiles 1, 2 and 3 (BILALLI *et al.*, 2018); minimum, maximum, mean and standard deviation, kurtosis and skewness (REIF; SHAFAIT; DENGEL, 2012).

Table 19 presents a non-exhaustive list of the summarization functions, their range, cardinality and a brief description. The *quantiles* and *histogram* result in multiple values. The former summarizes a measure by representative values of the measure distribution, whereas the latter uses the proportion of values in each range of data. A hyperparameter specifying the number of bins in which the results are split (KALOUSIS; THEOHARIS, 1999) defines the cardinality of the *histogram*. Some functions such as *count*, *histogram* and *kurtosis* change the range of the characterized measure, while others inherit the range of the measure in which they summarize, such as *max*, *mean* and *min*. The *identity function* is conceptually used when a characterization measure results in a single value ($k' = 1$).

Pinto, Soares and Mendes-Moreira (2016b) proposed that the summarization functions should be organized in groups: *descriptive statistical* includes the most common functions and summarizes a set of values in a single result such as *max*, *min*, *mean*, *median*, *sd*, *skewness*, *kurtosis*, *iqRange*, among others; *distribution* characterizes the distribution of the measure using multiple values. For this purpose, the use of histogram with a fixed number of bins (KALOUSIS; THEOHARIS, 1999) and the use of quartiles to summarize the set of values (BILALLI *et al.*, 2018) are alternatives observed in the literature; *hypothesis test* assesses an assumption about a set of values, resulting in one or more values, as the p-values and/or the tests result. However, its use has not been observed in the literature.

Table 19 – Main summarization functions.

Acronym	Range	Cardinality	Brief description
count	$[1,k]$	1	Computes the cardinality of the measure, suitable when the cardinality is variable.
histogram	$[0,1]$	*user*	Describes the distribution of the measured values, suitable for measures with high cardinality.
iqRange	$[0,\infty]$	1	Computes the interquartile range of the measured values.
kurtosis	$[-3,\infty]$	1	Describes the shape of the measured values distribution.
max	*inherited*	1	Results in the maximum values of the measure.
mean	*inherited*	1	Computes the averaged values of the measure.
median	*inherited*	1	Results in the central value of the measure.
min	*inherited*	1	Results in the minimum value of the measure.
quartiles	*inherited*	5	Results in the minimum, first quartile, median, third quartile and maximum of the measured values.
range	$[0,\infty]$	1	Computes the range of the measured values.
sd	$[0,\infty]$	1	Computes the standard deviation of the measured values.
skewness	$[-\infty,\infty]$	1	Describes the distribution shape of the measured values in terms of symmetry.

Conceptually, any function that offers guarantees of a fixed cardinality, regardless of the number of values received by it, can be applied as a summarization function. Thus, even though a *post-processing* function (PINTO; SOARES; MENDES-MOREIRA, 2016b) can also generate indiscriminate number of values, a summarization function cannot. The summarization functions presented in Table 19 can be applied to all multi-valued measures indiscriminately. Some combinations measure/summarization-function explore semantic concepts, e.g. the standard deviation of the classes proportion (LINDNER; STUDER, 1999). Particular summarization functions, suitable for a specific measure, such as the *nrCorAttr* statistical meta-feature, that summarizes the *cor*, is better instantiated as a meta-feature. Section 3.4.4 addresses this matter as an open issue and shows possible insights concerning their use and exploration.

3.4 Discussion

In machine learning, it is expected that all information necessary to reproduce empirical experiments, obtaining similar results, should be clearly reported. For MtL, the information's need to maintain the reproducibility is even greater, since this research topic also includes all the machine learning analysis plus the recommendation system which is based on the characterization of several datasets and the performance assessment from a set of algorithms over the datasets. However, many details related to them are frequently ignored or subtly addressed in the literature.

This section focuses on six aspects of the characterization process, most of them strictly related to the taxonomy proposed in Section 3.2. Frequently ignored details, the unspoken decisions taken by researchers, are reviewed, along with the enumeration of gaps that demand further analysis whether theoretical, empirical or both.

3.4.1 Input Domain

The input domain defines the data type supported by a meta-feature. For instance, statistical meta-features support only numerical data while information-theoretic meta-features support only categorical data. The alternatives adopted to handle non-supported data types have rarely been reported in the literature, as observed in Smith *et al.* (2001), Ali and Smith (2006), Reif *et al.* (2014), Garcia, Carvalho and Lorena (2016). Besides the fact that such choices affect the reproducibility of MtL experiments, their impact on the outcomes is unknown.

Figure 9 summarizes the options adopted in the literature to deal with the data type. The options consist of ignoring (KALOUSIS; THEOHARIS, 1999) or transforming the data (CASTIELLO; CASTELLANO; FANELLI, 2005). By ignoring the attributes, two problems are faced: *(i)* if a dataset contains only attributes with the ignored data type, all respective measures will have missing values; *(ii)* in an MtL context, the algorithms/techniques recommended may support the ignored data. In favor of this choice, it can be argued that to using only the meta-features that are able to characterize such data is a natural choice since they can properly represent the data (MICHIE; SPIEGELHALTER; TAYLOR, 1994). Besides, their inability to process some types of data may be aligned with the limitations of some algorithms, therefore representing useful information. Alternatively, the datasets can be segmented by type (only numerical, only categorical and mixed) where only the suitable measures for each group are used (BILALLI; ABELLÓ; ALUJA-BANET, 2017; KOPF; IGLEZAKIS, 2002).

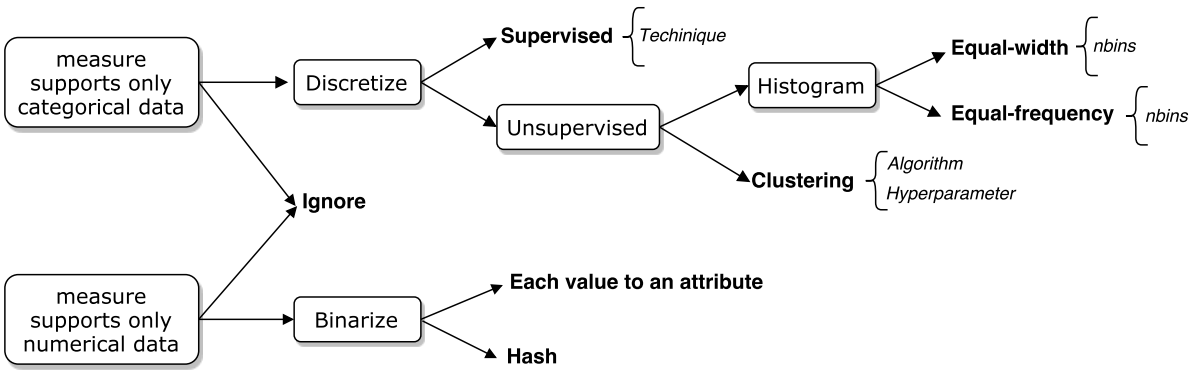


Figure 9 – Options to handle the input data type that are not supported by the meta-features.

By transforming the attributes, the meta-features can support any data types using a *binarization* or *discretization* approaches. It leads to new decisions since there are different alternatives used to transform the data, including the possibility of combining them together.

The most common transformation of categorical attributes into numerical ones is called *binarization* (AGGARWAL, 2015). In this process, ϕ new binary attributes are created to represent each different category in the data, where ϕ is the number of distinct categories in the attribute. For each instance, only one of the new attributes is assigned to "1" while the others are assigned to "0". Its use to transform categorical attributes with a high number of distinct values

is not recommended, since it generates a large number of new attributes. Alternatively, each category can be mapped to an integer and then represented in a binary hash, where $\log_2(\phi)$ new attributes are used to represent the bits values of the represented information (TAN; STEINBACH; KUMAR, 2005). The unintended relationships among the new attributes can be a deficiency of this approach, considering the meaninglessness of these relations.

Similarly, some meta-features support only categorical attributes, and the transformation from numeric to categorical attributes may be necessary. For such, *discretization* techniques can be used. These techniques distribute numeric values in ϕ distinct intervals, which correspond to the new categories (AGGARWAL, 2015). As a result, order relations in the original values and variations within the same interval are lost. In an unsupervised approach, the intervals can be defined using *equal-width* or *equal-frequency*, where they have the same interval width or the number of values, respectively. Other techniques such as clustering, correlation analysis and decision tree analysis can also be used for value discretization (FAYYAD; IRANI, 1993; HAN; KAMBER, 2006). The last two, which are supervised approaches, use the target attribute to define the categories.

The discretization procedure has a larger number of alternatives than the binarization procedure, which makes the result even more biased when they are arbitrary-defined. Most known methods are based on supervised and unsupervised techniques. The unsupervised techniques include the histogram and the clustering strategy. Given that in each transformation there is a loss of information and a good discretization process can minimize it (JIN; BREITBART; MUOH, 2009). Because the unsupervised approaches are the simplest alternatives to discretize the data, more information are lost in the process, however, they have a lower cost than the supervised approaches.

The presence of missing values in the original datasets also demands attention, considering that many meta-features do not support the defective records. The alternatives to address this issue are: *(i)* imputation of values provided by a preprocessing step and *(ii)* removal of attributes and/or records with missing values. This topic is also frequently ignored in MtL papers.

3.4.2 Hyperparameter values

Another aspect that impacts the reproducibility of MtL experiments is the lack of details with regards to the hyperparameter values required by the measures. Possibly, this occurs because a value is used by default.

Tables 12, 14, 15 and 18 identify the measures that require the definition of hyperparameter values. Some statistical measures have specific hyperparameter values. All model-based and landmarking meta-features, on the other hand, have hyperparameter values that affect the whole group. For the model-based, different DT algorithms can be used to induce the model and each algorithm requires additional configurations. For the landmarking, the validation strategy, the

evaluation measure and also the algorithms hyperparameters can be modified. In these cases, the same set of configurations is usually adopted for all measures of the group, but not necessarily by more than one author.

Other decisions concerning the use of meta-features and summarization functions can also be seen as hyperparameters. For instance, how to handle the unsupported data type, as described in Subsection 3.4.1, and the transformation by class (CASTIELLO; CASTELLANO; FANELLI, 2005) proposed to explore the target information, affect the statistical and information-theoretic groups and can also be defined as hyperparameters. Additionally, the *histogram* summarization function also has a hyperparameter that defines the number of bins to represent the measures.

In summary, the effects of such choices in the data characterization process are unknown. Alternatives, such as tuning the different parameters of the measures, using distinct instances of the same measure and evaluating the amount of information captured by them, have not been explored.

3.4.3 Range of the Measures

The data range has been frequently ignored in MtL studies, which suggests that meta-features have been used directly without transformation or it has not been properly reported. Although meta-features have a different range of values, they are used together in a meta-base. Considering that some algorithms are influenced by attributes with different ranges (HAN; KAMBER, 2006; WANG *et al.*, 2013), the meta-data can be transformed by min-max scaling or z-score normalization, as illustrated by the vertical axis in Figure 10.

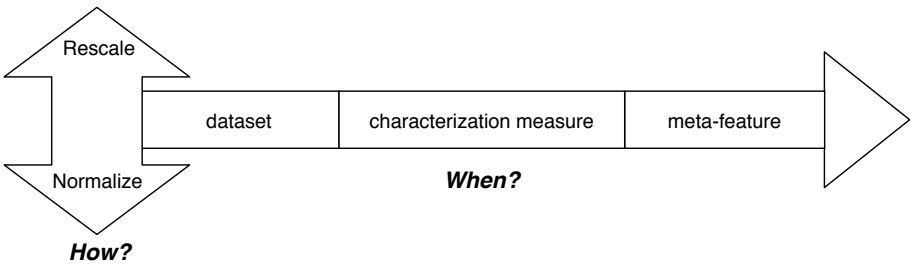


Figure 10 – Options to transform the range of the measures.

The transformation can occur in three distinct moments: *(i)* in the *dataset*, before any computation; *(ii)* in the result of the characterization measure, before the summarization function; and *(iii)* in the meta-base, after computing the meta-feature. These moments are represented by the horizontal axis in Figure 10. They have some implications in the result regardless of how the transformation occurs.

The dataset transformation is an alternative for the measures whose scale is determined by the values present in the dataset (range is *inherit*). Changes in the original data range will reflect on the outcome of these meta-features. The second alternative transforms the result of the

characterization measures. It is more suitable for multi-valued measures. Both alternatives are not recommended for meta-features using summarization functions on a particular scale, such as *kurtosis* and *skewness*. Finally, the most conventional approach is to transform the meta-features result, which requires the characterization of all datasets beforehand.

Some rescaled meta-features are used along with (or instead of) their original version. The proportion of numeric and categorical attributes (BRAZDIL; GAMA; HENERY, 1994; KALOUSIS; HILARIO, 2001b), the proportion of attributes with outliers and normal distribution (BRAZDIL; SOARES; COSTA, 2003; SALAMA; HASSANIEN; REVETT, 2013) and the normalized entropy (CASTIELLO; CASTELLANO; FANELLI, 2005), are some examples found in the literature. However, only a few measures have their rescaled version named. The theoretical maximum and minimum values from the measures with a non-infinity range can be modified with the min-max scaling. The transformation of meta-features for some dataset characteristic (e.g. the number of instances) using absolute or relative values can be a better alternative.

In summary, the lack of information about the procedures adopted concerning the meta-data transformation is also a barrier to reproducible MtL studies. The different alternatives to transform the meta-features can suit some meta-features better than others. Although this topic does not contribute directly to the reproducibility issue, it is a very important research question that has not been satisfactorily addressed in the MtL literature.

3.4.4 Summarization Functions

In most MtL studies, summarization functions are combined with meta-features, either implicitly or explicitly. Implicitly when they are defined as part of the meta-feature formalization (CASTIELLO; CASTELLANO; FANELLI, 2005; FILCHENKOV; PENDRYAK, 2015; KUBA *et al.*, 2002; PENG *et al.*, 2002a), where the average result is the most natural solution used. Explicitly when studies show the effectiveness of using other options to summarize measures (KALOUSIS; THEOHARIS, 1999; PINTO; SOARES; MENDES-MOREIRA, 2016b; REIF; SHAFAIT; DENGEL, 2012; TODOROVSKI; BRAZDIL; SOARES, 2000), as reported in Section 3.3.7.

Some combinations of meta-features and summarization functions have a semantic meaning. For instance, the standard deviation (*sd*) applied to the frequencies of the classes (*freqClass*) shows how uniform the class distribution is, which may also indicate that the classes are unbalanced. Other combinations are meaningless, as the use of the cardinality of the measure (*count*) to summarize the joint entropy (*jointEnt*), since the measure has a fixed cardinality. There are also some possible problematic combinations, such as the use of *histograms* to summarize meta-features with low cardinality and/or with the range that is defined according to a dataset characteristic. In this case, the histogram bins can be sparse and represent different scales of values for each dataset.

The use of many functions to summarize a measure proportionally increases the number of meta-features obtained. As many measures are multi-valued, hundreds of results can be easily obtained when combined with multiple summarization functions. The relatively low number of meta-instances usually observed in MtL experiments together with the high number of meta-features could generate meaningless models due to the curse of dimensionality (TAN; STEINBACH; KUMAR, 2005). The use of a feature-selection algorithm can be an alternative to deal with this problem (LEMKE; BUDKA; GABRYS, 2015; PINTO; SOARES; MENDES-MOREIRA, 2016b).

Even though summarization functions are not strictly related to reproducibility issues, they are relevant to reproducibility because different choices can be made in a characterization process. The empirical analysis of summarization functions and the exploration of new ways to summarize meta-features should be the subject of future research.

3.4.5 Exceptions

As discussed previously, some measures can be incorrectly computed for some datasets. Their use requires specific conditions that cannot always be guaranteed. Operations such as division by zero and logarithm of negative values are the main causes of exceptions.

Alternatives to deal with problematic measures are: *(i)* assuming it results in a missing value; *(ii)* using a default value; *(iii)* if the measure is multi-valued, ignore it. The first option results in a meta-base with missing values, which eventually will be filled using some pre-processing technique (HAN; KAMBER, 2006) or removed from the meta-base. The other two alternatives fix the problem of having a missing value during the computation of the meta-feature.

The use of a default value to represent exceptional cases can be positive when it properly characterizes the measure and the phenomenon that generates the exception. Table 20 presents default values, suggested by the authors, to be used when a meta-feature cannot characterize a dataset. With the exception of *sdRatio*, the values are in the range of their measures, assuming a semantic meaning as explained in the column *Meaning*.

The previous alternatives can introduce noise in the predictive meta-data. This does not occur when the defective results can be removed before the summarization. As a drawback, this alternative is valid only for the multi-valued measures. Furthermore, to discard few values for measures with high cardinality, the final result will not change drastically, but for the measures with low cardinality, this approach may lead to distortions in results.

Summarization functions can also generate exceptions. This is the case of *sd*, *kurtosis* and *skewness*. The *sd* cannot be applied to single values while the *kurtosis* and *skewness* cannot be applied to constant vectors. The alternatives *i* and *ii* can also be adopted for them. The value 0 is the default value suggested to fill the problematic cases, which represents no deviations for *sd* and constant values for *kurtosis* and *skewness*.

Table 20 – Suggested values to fill the missing cases for the meta-features with exceptions.

Group	Measure	Default	Meaning
Mono-valued measures			
Simple	*catToNum*	`d`	All attributes are categoric.
	numToCat	`d`	All attributes are numeric.
Statistical	*nrCorAttr*	`0`	No pair of attributes is highly correlated.
	sdRatio	`-1`	Invalid result.
Multi-valued measures			
Statistical	*cor*	`0`	No correlation.
	gMean	*mean*	Mean value.
	kurtosis	`0`	Constant values.
	skewness	`0`	Constant values.
Landmarking	*linearDiscr*	`0`	Low predictive performance.

In summary, the use of these measures and summarization function does not imply that they will generate exceptions during the extraction of meta-features. However, there is an absence of information about the occurrence or lack of occurrence in empirical studies in MtL. Thereby, it is strictly related to the reproducible of the MtL studies, given that it has a technical bias and is related to the implementation and use of meta-features.

3.4.6 Meta-feature Space

The ratio between the number of meta-features and the number of meta-instances in MtL experiments is usually higher than in conventional ML experiments. Furthermore, it is well known that the most suitable meta-features varies for different MtL tasks (BILALLI; ABELLÓ; ALUJA-BANET, 2017). Thus, some studies have investigated the use of feature selection techniques (PINTO; SOARES; MENDES-MOREIRA, 2016b; SALAMA; HASSANIEN; REVETT, 2013) and the transformation of the meta-features' space (BILALLI; ABELLÓ; ALUJA-BANET, 2017; MUÑOZ *et al.*, 2018) to reduce the dimensionality of meta-bases, as well as to increase the predictive performance of meta-models (KALOUSIS; HILARIO, 2001a).

Meta-feature selection is just an instance of feature selection (LEMKE; BUDKA; GABRYS, 2015). Among the different approaches for meta-feature selection, wrapper appeared more often in our literature review (TODOROVSKI; BRAZDIL; SOARES, 2000; KALOUSIS; HILARIO, 2001a; BRAZDIL *et al.*, 2009; REIF *et al.*, 2014; FILCHENKOV; PENDRYAK, 2015; GARCIA; CARVALHO; LORENA, 2016) than the use of a filter (PENG *et al.*, 2002b; LEE; GIRAUD-CARRIER, 2008; PINTO; SOARES; MENDES-MOREIRA, 2016b).

In Muñoz *et al.* (2018), the authors followed a new approach for meta-feature selection. They investigated the behaviour of several meta-features in 12 classification challenges. By modifying a dataset to increase/decrease each investigated problem, the variance of the meta-features is statistically assessed, revealing those that better characterize each variation. After the repetition of the process using different datasets, the most relevant features for each challenge

are obtained.

Another work for the meta-feature dimensionality reduction used PCA (HOTELLING, 1933) to obtain latent meta-features (BILALLI; ABELLÓ; ALUJA-BANET, 2017). After computing the principal components, the most relevant (according to the cumulative total variance) are selected. The authors later used a filter based on correlation with the target to select a subset of the latent meta-features.

As PCA does not take into account the target variable to transform the data, Muñoz *et al.* (2018) used optimization to transform a set of previously selected meta-features into a 2-D space. For such, the authors used the performance of several learning algorithms. Named instance space, it enables the visualization of the set of datasets used in an MtL study.

Most of the studies found for this study use wrapper. Few studies use transformation approaches in MtL. Some works have compared groups of meta-features (ABDELMESSIH *et al.*, 2010; KOPF; IGLEZAKIS, 2002; REIF; SHAFAIT; DENGEL, 2011; REIF *et al.*, 2014), with different findings. For instance, landmarkings and model-based meta-features were the most important characterization measures in Reif *et al.* (2014) and Filchenkov and Pendryak (2015), respectively. In contrast, feature selection wrapper did not improve the predictive performance of the meta-models in Garcia, Carvalho and Lorena (2016).

To estimate the importance of a meta-feature, Filchenkov and Pendryak (2015) uses a significance measure that associates the predictive performance of a model induced using each meta-feature alone. This process is repeated several times and the average performance obtained for each meta-feature is the meta-feature significance value. Pimentel and Carvalho (2019) define the meta-feature importance as the number of times it is selected when the Random Forest algorithm is applied to the meta-base. In Salama, Hassanien and Revett (2013), Peng *et al.* (2002b), the authors use the correlation between them and the meta-target to select the meta-features (PENG *et al.*, 2002b; SALAMA; HASSANIEN; REVETT, 2013).

The decision of whether to use reduction and/or transformation is an important issue in the reproducibility and performance of MtL experiments. When used, a detailed specification of the procedures adopted is essential for the replication of the experiments. Moreover, while meta-feature selection may improve the interpretability of the meta-models, the same is not the case when a transformation is used.

3.4.7 Outline

The previous subsections discussed the main aspects related to the reproducibility of MtL experiments. They refer to the alternatives and decisions taken that need to be properly reported. Furthermore, some gaps were identified, mainly because it is unknown how the different choices could impact the characterization process. Below, each topic regarding the reproducible issues and gaps are summarized. The details can be seen in the respective subsection.

Input domain: Some measures support only categorical data while others, only numeric. The alternatives to handle with this issue are *ignoring*; *transforming*, which implies in other decisions (see Figure 9); *segmenting* the experiments and datasets. The impact of such choices in the statistical and information-theoretic meta-features is unknown. Furthermore, datasets may have missing values, which will require imputation of values or the removal of the defective records.

Hyperparameters: Some meta-features or groups of them require the definition of hyperparameters (see Table 21). The way the hyperparameters affect the model-based and landmarking meta-features is unknown. Also, approaches like tuning and the use of different hyperparameters values for the same measure have not been explored yet.

Range of the measures: The meta-features have distinct range of values. The alternatives to handle with this issue are *ignoring* or *transforming*. In the latter (see Figure 10), the *min-max rescaling* and *z-score normalization* are procedures that can be used; the *dataset*, *characterization measure* and the *meta-feature* represent the objects to be transformed. The gaps are concerned with identifying suitable combinations between the two dimensions and the normalization of the meta-features.

Summarization functions: Different functions can be employed to summarize the measures result. The investigation of how the summarization functions affect the measures' results are still incipient. Furthermore, finding new alternatives to summarize the measures may increase the discriminative power of the meta-features.

Exceptions: Some measures cannot be computed for all datasets. The alternatives to handle this issue are *ignoring* or *replacing*. In the latter, the alternatives are *applying a preprocessing technique*; *using a default value*; *removing the missing values* (only for multi-valued measures). However, the impact of such choices in the characterization result is unknown.

Meta-feature space: Meta-feature dimensionality reduction can be performed using a *feature-section* and/or *transformation* approach. In the former, *wrapper* is more often used than *filter*. In the latter, although PCA is most commonly used, it is used in a small number of studies. While the use of feature selection allows model interpretability, transformation usually has a lower computational cost.

We reinforce that many of those issues have not been properly reported in the MtL literature. This list can be used as a guideline for future studies involving dataset characterization. The next section addresses the characterization tools that contribute directly to reproducible empirical research in MtL.

3.5 Tools

Characterization tools have an important role in the development of research in MtL. Besides simplifying an essential step of the work, their use corroborates the reproducibility of MtL experiments. However, the approach used in the development of the tool can generate two different perspectives: *(i)* a black box tool with abstracted choices, which promotes reproducibility, but only for the users that use the same tool or, *(ii)* a white box tool that exposes all the options to the user promoting reproducibility even with different tools, but forcing them to make the explicit decisions about the parameter values.

The Data Characterization Tool (DCT)[3] (LINDNER; STUDER, 1999) is the most referenced characterization tool in the MtL literature (BENSUSAN; GIRAUD-CARRIER, 2000; KOPF; IGLEZAKIS, 2002; PFAHRINGER; BENSUSAN; GIRAUD-CARRIER, 2000; REIF *et al.*, 2014), to cite a few. The DCT contains a representative subset of meta-features from simple, statistical and information-theoretic groups.

Matlab Statistics Toolbox (MATHWORKS, 2001) have also been used to characterize statistical measures (ALI; SMITH, 2006; ALI; SMITH-MILES, 2006; SMITH-MILES, 2008). Weka (HALL *et al.*, 2009), RapidMiner (MIERSWA *et al.*, 2006) and other general data mining tools can be employed to compute landmarking meta-features (ABDELMESSIH *et al.*, 2010; BALTE; PISE; KULKARNI, 2014).

Nowadays, OpenML (VANSCHOREN *et al.*, 2013) is the most robust tool available to characterize datasets, though it has a broader purpose. Many of the reported measures are available in the platform, which is also a benchmarking repository that contains the characterization of several datasets. OpenML uses an extension of the Fantail library (SUN; PFAHRINGER, 2013), also available on GitHub.[4] A drawback may be that the characterization process is performed automatically when a new dataset is submitted to the platform, which abstracts the users' choices. On the other hand, anyone can compute and upload their meta-features to OpenML through its API.[5]

The framework proposed by Pinto, Soares and Mendes-Moreira (2016b) is available as an open GitHub project[6], but without the implementation of the meta-features, which could be an expensive task. Except for it, all the reviewed tools are black-box tools.

In parallel, many authors have used their implementation of the meta-features (FILCHENKOV; PENDRYAK, 2015; GARCIA; CARVALHO; LORENA, 2016; REIF *et al.*, 2014; TODOROVSKI; BRAZDIL; SOARES, 2000), without reporting and making publicly available their implementation. This practice negatively affects reproducibly and comparison of results. Besides, without source code and widespread use, there is a chance that the implementations work as

[3] <https://github.com/openml/metafeatures/dct>
[4] <https://github.com/quansun/fantail-ml>, <https://github.com/openml/EvaluationEngine>
[5] <https://www.openml.org/api_docs#!/data/post_data_qualities>
[6] <https://github.com/fhpinto/systematic-metafeatures>

they should. A positive step towards reproducibility is the *"Paper with code"*[7] platform, which provides code repository. However, comparing the number of MtL related works published in the last 5 years with the number of codes available at the *"Paper with code"* website,[8] the practice of publishing the code/results is unfortunately still incipient.

3.5.1 MFE Tool

Aiming to offer a robust, flexible and standalone data characterization tool, the authors developed the Meta-Feature Extractor (MFE) tool[9] that contains the implementation of most of the meta-features and summarization functions described in this paper. MFE also implements solutions for some of the issues discussed in Section 3.4 and provides a simple and flexible tool specifically designed to characterize datasets.

MFE allows the user to compute a specific, a group of or all meta-features available. It is possible to define which summarization functions should be computed and, optionally, to obtain all computed values for a given set of measures, without summarizing the results. Many of the hyperparameters can be changed according to the user's preferences, as shown in Table 21, which also includes the default values adopted for all of them. It is worth highlighting that the robustness of these choices, regarding the characterization process, is usually unknown, although they are consistent with the literature and the authors' experience. The column "Details" presents the rationale behind the decisions taken.

As a limitation, MFE does not support to characterize non-classification datasets and does not accept datasets with missing values. An extension to other meta-features needs to follow the discussion described in Section 3.4. The authors believe that MFE can be used in any MtL experiment that requires the characterization of datasets, similar to DCT in the past, but with more flexibility.

3.6 Conclusion

The recommendation of techniques by using MtL is an effective alternative to deal with the selection of the most suitable techniques among a large number of possibilities. However, many MtL studies adopt different methodologies and design approaches, which affect the reproducibility of the experiments. By discussing topics that have been frequently ignored in the MtL literature and suggesting possible alternatives to approach them, this paper reviewed the main characterization measures and important issues related to the reproducibility of MtL experiments, in addition to the proposal of a new taxonomy for meta-features and the MFE tool.

[7] <https://paperswithcode.com/task/meta-learning/>

[8] Using Scopus, we found 412 papers related to MtL in the last 5 years, whereas we found only 65 works in the *"Paper with code"* website.

[9] Available in Python (<https://pypi.org/project/pymfe/>) and R (<https://cran.r-project.org/package=mfe>) languages

Table 21 – Hyperparameters and their adopted default values in the MFE tool.

Measure	Hyperparameter	User	Details
Statistical			
all	transform = `TRUE`	Yes	Defined according to an exploratory analysis, to reduce the number of missing values in the meta-features. By setting it as *true* the categorical attributes will be binarized using simple transformation, whereas with *false* they will be ignored.
	by.class = `FALSE`	Yes	Enables the measure extraction by class, as proposed by Castiello, Castellano and Fanelli (2005).
cor	method = "pearson"	Yes	Options: "kendal" and "spearman"
nrCorAttr	method = "pearson"	Yes	Options: "kendal" and "spearman"
	threshold = `0.5`	No	As defined in Salama, Hassanien and Revett (2013)
nrNorm	W-Test for normality	No	Details in Royston (1995)
propNorm	W-Test for normality	No	Details in Royston (1995)
nrOutliers	Tukey's boxplot	No	Details in Rousseeuw and Hubert (2011)
propOutliers	Tukey's boxplot	No	Details in Rousseeuw and Hubert (2011)
tMean	trim = `0.2`	No	As defined in Ali and Smith-Miles (2006)
Information-theoretic			
all	transform = `TRUE`	Yes	Defined according to an exploratory analysis, to reduce the number of missing values in the meta-features. By setting it as *true* the numeric attributes will be discretized using equal-frequency histogram transformation, whereas with *false* they will be ignored. The number of bins is set to $\sqrt[3]{n}$.
Model-based			
all	algorithm = Cart	No	Details in Breiman *et al.* (1984).
Landmarking			
all	Cross-validation	No	Methodology used in order to obtain more stable results.
	folds = 10	Yes	Also defines the measures cardinality.
	score = "accuracy"	Yes	Options: "balanced.accuracy" and "kappa".
bestNode	algorithm = Cart	No	Details in Breiman *et al.* (1984).
randomNode	algorithm = Cart	No	Details in Breiman *et al.* (1984).
worstNode	algorithm = Cart	No	Details in Breiman *et al.* (1984).
Miscellaneous			
gravity	distance = "euclidian"	No	As defined in Ali and Smith (2006).

The new taxonomy organized and formalized the current meta-features and their usefulness across different types of task, domain, range, and several other characteristics that can impact MtL tasks. Based on this review, the authors enumerated the main decisions a researcher faces when using meta-features. Moreover, a detailed discussion is provided on the cutting edge subgroup of meta-features, their predictive power and the use cases where these measures have been applied. In addition to this study, the MFE package was proposed to support the data characterization process implementing the framework proposed with the main meta-features included in the discussion.

Future work shall investigate meta-features for other types of tasks, such as regression and clustering; increase the interpretability of the meta-features; and explore empirical analysis

showing how some choices related to the hyperparameters, cardinality and the summarization functions can affect dataset characterizations to best distinguish the performance of meta-models. A review of regression and clustering meta-features could improve the task representation and could also look at a different perspective and validate the taxonomy proposed. The exploration of interpretability of the meta-features and the empirical analysis over hyperparameters, cardinality and summarization function could improve the meta-model representation and performance.

CHAPTER

4

LABEL OPERATION FOR MULTI-LABEL OPTIMIZATION

Collaborating authors

Carlos Soares
Fraunhofer AICOS and LIAAD-INESC TEC, University of Porto, Porto, Portugal

Bernhard Pfahringer
University of Waikato, Hamilton, New Zealand

André C. P. L. F. de Carvalho
University of São Paulo, São Carlos, Brazil

Abstract

In multi-label learning, instances are associated with different labels simultaneously. A common approach to deal with this situation is the transformation of the original dataset into a set of single-label datasets associated with each label. Next, a base, single-label, classification algorithm is applied to each dataset. The results obtained for each single-label dataset are merged, returning the overall multi-label prediction. By looking only at the overall multi-label predictive performance, we can overlook some internal problems. One of them is that, in the single-label classification tasks, some labels can be never or always predicted. So far, very little attention has been paid to this issue, which may produce misleading results. This paper investigates some alternatives to deal with these label prediction problems. In addition to using a traditional approach (thresholding calibration), we introduce a new transformation that, by combining labels, improves the predictive performance for a particular evaluation measure. Two operations, label expansion and label reduction, are proposed to enhance the predictive performance for the label

measures AUC, F1 and precision. Considering that the labels are individually optimized, we expect to reduce the label prediction problems. According to the empirical results, the proposed approaches can mitigate the label prediction problems. We show that they can improve predictive performance for different evaluation measures, including the non-trivial AUC, reducing the label prediction problems regardless of the base algorithm used.

4.1 Introduction

Many real-world data science applications, from different domains, such as text, multimedia and biology, are frequently related to multiple concepts simultaneously (GALINDO; VENTURA, 2014). Multi-Label Classification (MLC) strategies have been largely used to deal with them.

A common approach to support MLC tasks consists of transforming the original multi-labeled data into single-labeled data and applying conventional classification algorithms to solve each task separately, combining the results at the end (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). Binary Relevance (BR) (BOUTELL *et al.*, 2004) is the simplest and most popular transformation strategy (LUACES *et al.*, 2012; MONTAÑÉS *et al.*, 2014). BR generates an independent binary dataset for each label, using the one-versus-all approach, which is used to induce a learning model that is able to predict the relevance of such labels. However, it has been criticized as it does not explore the labels' dependencies and generates highly imbalanced datasets due to the one-versus-all transformation (ZHOU; TAO; WU, 2012; ZHANG *et al.*, 2018).

Recently, we have observed for a particular dataset that some labels are never correctly predicted, just as others are always predicted, regardless of the strategies and base algorithms (RIVOLLI; SOARES; CARVALHO, 2018a). Furthermore, such label prediction problems seem to be recurrent for distinct datasets (RIVOLLI; SOARES; CARVALHO, 2018b), despite the fact that no previous study has investigated them.

Thus, this paper investigates alternatives in order to address label prediction problems. In addition to the use of a traditional approach, called threshold calibration (FAN; LIN, 2007), we propose and formalize a multi-label transformation, called label operation. Considering: *(i)* the intrinsic relationships between labels; *(ii)* the infrequent labels, such that some of them are not properly represented by instances; *(iii)* the possibility of noises being introduced during the labeling process; we believe that the combination of specific pairs of labels can lead to improvement predictive performance. We propose and experimentally investigate here two different approaches for this combination: label expansion and label reduction. While the former increases the number of instances associated with a label, the latter reduces the number of instances that are not associated with it. These operations use another, possibly related, label to guide the transformation, indirectly exploring the labels' dependencies. As consequence, they

obtain more balanced datasets and can reduce label noises.

In a previous study (RIVOLLI; SOARES; CARVALHO, 2018b), we introduced the label expansion as an alternative to deal with the label prediction problems. By generalizing it in the label operation; investigating a new operation, the label reduction; and performing an in-depth and broader set of experiments; the main contributions from this new study include:

- Proposition of two operations: label expansion and label reduction; that are empirically evaluated in order to optimize the AUC, F1 and precision label measures.
- Formalization of these label operations as a multi-label optimization procedure.
- Experimental comparison of the proposed label operations with an alternative approach found in the literature: threshold calibration.

Experimental results show that the investigated approaches are able to reduce the label prediction problems for most of the datasets. Overall, label operations was shown to be more robust than the threshold calibration for different base algorithms. In practical terms, the operations act on the learning process, whereas threshold calibration only affects the final result. Consequently, they can also be combined.

The rest of the paper is organized as follows: Section 4.2 formally defines MLC. Section 4.3 formalizes the label expansion, label reduction and their use as a multi-label optimization procedure. Section 4.4 presents the experimental evaluation process, describing the datasets, evaluation measures and procedures adopted in the empirical study. The results, which include the performance of the label operations and threshold calibration, are presented in Section 4.5. Afterward, Section 4.6 discusses the effect of these mechanisms, highlighting the main properties and implications of the label operations. The paper ends with Section 4.7, which summarizes the findings and future work directions.

4.2 Multi-label Learning

A MLC task is a classification task in which each instance can be simultaneously classified in more than one of the existing class labels (CARVALHO; FREITAS, 2009), using an induced predictive model $h: \mathscr{X} \rightarrow \mathscr{Y}$ from a set of labeled instances $\mathscr{D} = \{(\vec{x}_1, Y_1), ..., (\vec{x}_n, Y_n)\}$. In this equation, $\vec{x}_i \in \mathscr{X}$ is a vector with characterization features that describes the i^{th} instance and $Y_i \subseteq \mathscr{Y}$ are the set of labels associated with it, such that $\mathscr{Y} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ is the set of all q labels λ_j, representing concepts from a given domain.

Different strategies have been proposed to induce a predictive model h for an MLC task. They can be organized into two groups (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010): *problem transformation* and *algorithm adaptation*. The former transforms the original multi-label

dataset into a set of single-label datasets, which can be modelled with conventional binary classification approaches. For this reason, they can be seen as *algorithm independent* (CARVALHO; FREITAS, 2009). The latter modifies existing machine learning algorithms to intrinsically support the multi-labeled data.

The transformation process is usually performed using the one-versus-all, one-versus-one and multi-class approaches.. One-versus-all generates at least one dataset per label, in which each binary dataset $\mathscr{D}'_j = \phi(\mathscr{D}, \lambda_j)$ is related to the label λ_j. The instances associated with the λ_j label are labeled with class 1, and the others with class 0, such that

$$\begin{aligned} \phi(\mathscr{D}, \lambda_j) =& \{(\vec{x}_i, I(\lambda_j \in Y_i)) \mid (\vec{x}_i, Y_i) \in \mathscr{D}\}, \text{ where} \\ I(\cdot) =& \begin{cases} 1 & \text{if the predicate is true,} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.1}$$

From this transformation, the BR strategy uses the dataset $\mathscr{D}'_j$ to induce a binary model θ_j for each label λ_j. The final prediction is performed combining the predictions from all binary models as follows:

$$h_{br} = \{\lambda_j \mid \theta_j(\vec{x}) = 1,\ 1 \leq j \leq q\}.$$

Without loss of generality, a binary model θ_j predicts a class value from a real-valued function f_j. This value indicates a score of relevance, confidence or probability to a given instance to belong to the class λ_j. Hence, $\theta_j(\vec{x}) = 1 \iff f_j(\vec{x}) \geq \tau_{\lambda_j}$, where τ_{λ_j} is the threshold value for label λ_j. By default, the middle value of a range (e.g. $\tau_{\lambda_j} = 0.5$ for a [0,1] interval) is used as a decision point. Nevertheless, different evaluation measures can be optimized by varying this value (PILLAI; FUMERA; ROLI, 2013).

A widely threshold calibration strategy used in MLC tasks is the Score-Cut (SCUT) (YANG, 2001). From a validation set $\mathscr{D}_v$ and a given evaluation measure β that is maximized, a label-wise optimization with SCUT is performed as

$$\tau^*_{\lambda_j} = \arg\max_{\tau_{\lambda_j}} \beta(\theta_j(\mathscr{D}_v, \tau_{\lambda_j}), \mathscr{D}_v). \tag{4.2}$$

In this context, a label-wise optimization means that a threshold value τ_{λ_j} will be defined for each label. When τ_{λ_j} is low (close to 0), more instances are predicted with the respective label, which favors the recall. On the contrary, a high value (close to 1) is more restrictive, possibly favoring the precision.

In a recent study (RIVOLLI; SOARES; CARVALHO, 2018a), the authors observed the occurrence of three problems regarding the inability of an MLC model to properly predict some labels. The *Constant Label Prediction (CLP)* measures the proportion of labels predicted for all instances (Equation 4.3), whereas the *Missing Label Prediction (MLP)* measures the proportion of labels never predicted for any instance (Equation 4.4). Its generalization is the *Wrong Label*

Prediction (WLP), which measures the proportion of labels predicted incorrectly for all instances (Equation 4.5). Unless the individual performance of the labels is reported, the current MLC measures cannot identify these problems.

$$CLP = \frac{1}{q}\sum_{j=1}^{q} I(TN_j + FN_j == 0), \tag{4.3}$$

$$MLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j + FP_j == 0), \tag{4.4}$$

$$WLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j == 0). \tag{4.5}$$

These definitions use the confusion matrix values: TP_j, FP_j, TN_j and FN_j that, respectively, represent the true positive, false positive, true negative and false negative counts of the label λ_j; I is defined in Equation 4.1. These measures should be computed on a separate validation or test set.

Threshold calibration is one of the approaches investigated to solve the label prediction problems. Among thresholding strategies used to define the decision points (AL-OTAIBI; FLACH; KULL, 2014), we investigated the previously described SCUT strategy. Moreover, two label operations are proposed to tackle this issue. They are detailed in the next section.

4.3 Label Operation

A frequent assumption of MLC is that the labels are dependent (MONTAÑÉS *et al.*, 2014; PAPAGIANNOPOULOU; TSOUMAKAS; TSAMARDINOS, 2015; MENC'iA; JANSSEN, 2016). Different approaches address this assumption in different ways; for instance, Classifier Chains (READ *et al.*, 2011) use the set of already predicted labels as features in the prediction of another label. Other strategies (MONTAÑÉS *et al.*, 2014; CHERMAN; METZ; MONARD, 2012) use the stacked generalization approach to augment the input space of a given label considering all the other labels. In this sense, we propose a data manipulation approach, label operation, that changes the target values of a label based on other related label.

Different operations can define alternatives to guide the transformation. For instance, given two correlated labels that are assigned to a small number of instances, one can use as positive examples all the instances assigned to the other. This can bring more representativeness to the expanded label. In another scenario, the removal of the instances assigned exclusively with one of two labels can work as a noise reduction for the learning process of the other label.

Through the label operation, the transformation can be defined as a function $\phi(\mathscr{D}, \lambda_j, \lambda_k)$ such that, the label λ_k will be used to modify the binary dataset $\mathscr{D}'_j$ relative to the label λ_j. Two basic operations, expansion and reduction, are respectively detailed in Subsections 4.3.1 and

4.3.2. Finally, Subsection 4.3.3 discusses how to perform the operations since it is necessary to find the pairs of labels that are able to optimize a given evaluation measure.

4.3.1 Label Expansion

The Label Expansion (LE) operation between two labels ($\lambda_j + \lambda_k$) uses instances labeled with any of them as being assigned to the λ_j, for the transformation of the dataset D'_j. Consequently, it increases the number of instances associated with the expanded label λ_j (class 1) and reduces the number of instances with the class 0. Figure 11 shows the LE operation (box 'c') using two labels, $L1$ and $L2$, for an illustrative MLC dataset. Boxes 'a' and 'b' show the traditional transformation for both labels involved in the LE operation.

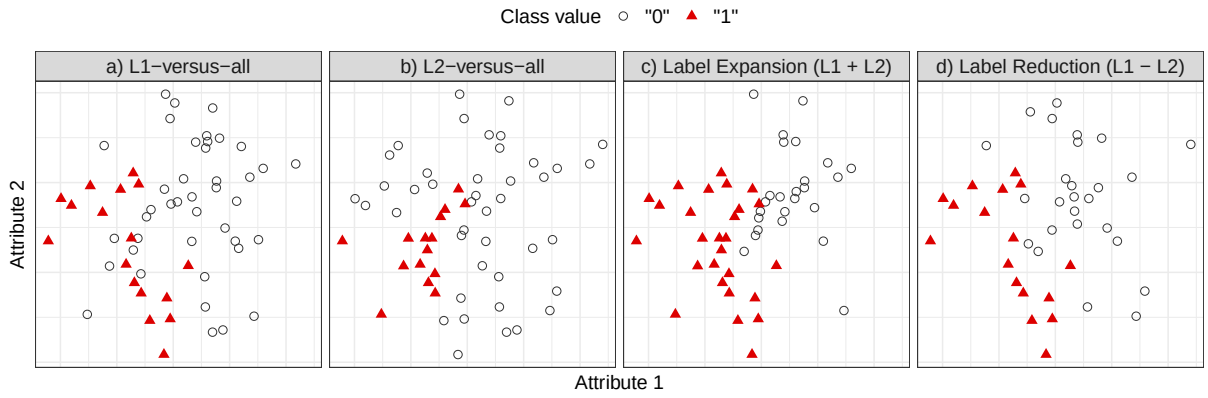


Figure 11 – Illustrative example of the default one-versus-all, label expansion and label reduction transformations for a multi-label data with two predictive attributes.

Formally, LE transformation is defined as

$$\phi_{LE}(\mathscr{D}, \lambda_j, \lambda_k) = \{(\vec{x}_i, I(\lambda_j \in Y_i \vee \lambda_k \in Y_i)) \mid (\vec{x}_i, Y_i) \in \mathscr{D}\}.$$

The transformation is symmetric ($\lambda_j + \lambda_k = \lambda_k + \lambda_j$), such that the same training data is obtained for both labels involved in the operation, leading to the same predictive model. However, the models are used in different problems, namely to predict different labels. More specifically, they will be evaluated using a different test set. Thus, λ_k may be used to expand λ_j to enhance an evaluation measure, but the opposite is not necessarily true.

Intuitively, LE can be suitable for expanding labels with few instances and between pairs of labels with an "is-a" relationship. By using instances from another label, it is expected to create better decision boundaries, despite the fact that some level of noise is also added to the target. The LE operation increases the cardinality and density of the training data which can improve the predictive performance of some MLC strategies (RODOVALHO; BERNARDINI, 2014). Moreover, it is a subtle way to explore the relationship between labels during the transformation process.

4.3.2 Label Reduction

The Label Reduction (LR) operation between two labels ($\lambda_j - \lambda_k$) removes the instances associated with the λ_k that is not related to the λ_j, for the transformation of the dataset D'_j. It reduces the number of instances with the class label 0 without changing the number of instances with the class label 1. The LR operation is also illustrated in Figure 1 (box 'd').

Formally, LR transformation is defined as

$$\phi_{LR}(\mathscr{D}, \lambda_j, \lambda_k) = \{(\vec{x}_i, I(\lambda_j \in Y_i)) \mid (\vec{x}_i, Y_i) \in \mathscr{D}, (\lambda_j \in Y_i \vee \lambda_k \notin Y_i)\}.$$

The LR transformation is asymmetric ($\lambda_j - \lambda_k \neq \lambda_k - \lambda_j$), thus the same pair of labels can result in two different training data according to the label that is reduced. While LE manipulates label values, LR removes instances; these are orthogonal dimensions. Nevertheless, the rationale is the same here, such that λ_k may be used to enhance an evaluation measure for the label λ_j, but the opposite is not necessarily true.

Intuitively, LR can be suitable to obtain transformed datasets with a better imbalance rate of the target. Different from LE, the LR does not introduce noise in the target, but, possibly eliminates it when the removed instances are close to the decision boundary. Thus, it may reduce possible overfitting in the learning process by removing similar instances associated with different target labels.

4.3.3 Performing Operations

Despite being simple, both label operations cannot be applied randomly. They require a procedure to identify the labels that can be expanded/reduced and the labels that can be used to expand/reduce other labels. Distinct approaches can be used to find the best options. A validation procedure can test several pairs of labels to identify the best combinations. However, as it has a high computational cost, heuristics can reduce the number of labels assessed in the validation procedure. Moreover, deterministic rules can be investigated in order to provide a reasonable solution.

The validation procedure requires a binary evaluation measure β.[1] Assuming that each label is independent and β is maximized, the procedure is performed in the following way

$$\arg\max_{\lambda_k} \beta(\theta_j(\mathscr{D}_v), \mathscr{D}_v) \quad | \quad \phi(\mathscr{D}_t, \lambda_j, \lambda_k) \rightarrow \theta_j, \tag{4.6}$$

where θ_j is the induced learning model for the label λ_j, $\mathscr{D}_t$ and $\mathscr{D}_v$ are, respectively, the training and validation datasets, respectively. This procedure can be used during the transformation and

[1] Despite the fact that a multi-label measure could also be optimized (FAN; LIN, 2007), in this work only binary evaluation measures are considered given its simplicity and direct association with the macro-averaged evaluation measures.

applied for each label. In the worst case, when $\lambda_k = \varnothing$, the default one-versus-all transformation (Equation 4.1) is applied.

This is a high-cost procedure, since it requires q^2 induced models, considering all pairs of labels, including the default transformation. As LE is symmetric, the number of induced models can be reduced to $q(q/2)$, which is still high for datasets with a large number of labels. Heuristics could be used to reduce the search space, thus avoiding testing all pairs of labels. Moreover, meta-learning (BRAZDIL *et al.*, 2017) is an alternative to deal with the exhaustive search. Their investigations are suggested as future works, thus the complexity involved to find the best pairs of labels will be overlooked from now on.

4.4 Experimental Evaluation

This section presents the procedures used to carry out the empirical evaluation of the proposed label operations. Besides, the traditional SCUT threshold calibration is investigated as alternative to deal with the label prediction problems. It describes the selected datasets, followed by a short overview of the selected measures and evaluation procedures used. Finally, it explains the performance estimation procedure adopted and the computational environment setup.

4.4.1 Datasets

Table 22 lists the 20 MLC datasets selected to be used in the experiments. They are from distinct domains (column *Domain*) and present a wide diversity in their characteristics. The columns *Inst*, *Attr* and *Lbl* are the number of instances, attributes and labels, respectively. Labelsets (*lSets*) is the amount of distinct label combinations; label cardinality (*lCard*) measures the average number of labels per instance; label density (*lDen*) describes the average frequency of labels; and dependency (*Dep*) shows the average unconditional labels' dependencies (LUACES *et al.*, 2012), illustrating how correlated is the subset of labels.

These datasets are frequently used as benchmarks for MLC experiments. They come from the Cometa repository (CHARTE *et al.*, 2018), an exhaustive collection of MLC datasets, integrated with the tools used in this work. The exceptions are the datasets `fapesp` and `msd-195` obtained with their respective authors, and `yelp8` from the Kaggle website.[2] The datasets were preprocessed with three operations. First, the labels with less than 10 instances were removed to ensure a minimum of instances related to each label in the training and test partitions. Next, the instances with no labels were also removed. Finally, the predictive attributes with constant values were removed.

[2] see <https://www.kaggle.com/c/yelp-restaurant-photo-classification>.

Table 22 – Characteristics of the MLC datasets.

Dataset	Domain	Inst	Attr	Lbl	lSets	lCard	lDen	Dep
20ng	text	19300	1006	20	55	1.03	0.05	0.08
birds	audio	337	260	15	115	1.84	0.12	0.08
cal500	audio	502	68	141	502	25.54	0.18	0.14
corel5k	image	4995	499	218	2940	3.37	0.02	0.16
emotions	audio	593	72	6	27	1.87	0.31	0.28
enron	text	1702	1001	42	722	3.34	0.08	0.12
fapesp	text	251	7286	18	61	1.35	0.08	0.11
flags	other	194	19	7	54	3.39	0.48	0.15
foodtruck	other	407	21	12	116	2.29	0.20	0.14
image	image	2000	294	5	20	1.24	0.25	0.15
langlog	text	1197	916	38	223	1.31	0.03	0.06
medical	text	949	1421	20	55	1.20	0.06	0.19
msd-195	audio	2901	180	38	267	2.47	0.07	0.24
ohsumed	text	13929	1002	23	1147	1.66	0.07	0.04
scene	image	2407	294	6	15	1.07	0.18	0.11
slashdot	text	3776	1079	18	149	1.18	0.07	0.05
stackex-chess	text	1612	585	78	725	2.07	0.03	0.10
tmc2007-500	text	28596	500	22	1172	2.22	0.10	0.11
yeast	biology	2417	103	14	198	4.24	0.30	0.25
yelp8	image	10784	668	8	117	2.26	0.28	0.11

4.4.2 Evaluation

The evaluation of the predictive performance of MLC strategies requires using specific measures that are able to explore their particularities (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). Three macro-averaged label-based evaluation measures are considered: *macro-AUC*, *macro-F1* and *macro-precision*. They averages the result of a binary evaluation measure applied for each label, such that

$$macro{-}\beta = \frac{1}{q}\sum_{j=1}^{q}\beta_j, \tag{4.7}$$

where $\beta_j = \{AUC_j \mid F1_j \mid precision_j\}$ are given by Equations 4.8, 4.9 and 4.10, respectively. The macro-averaged version summarizes the label results by giving the same weight to all labels (YANG, 1999). It assesses the consistency across them, even though it may be impacted by the performance of the least common labels (JACKSON; MOULINIER, 2002). Furthermore, the optimization of an individual label measure enhances the respective macro-average multi-label measure (DÍEZ *et al.*, 2015).

$$\begin{aligned} AUC_j = & \frac{\mid \left\{(\vec{x},\vec{z}) \;\mid\; f_j(\vec{x}) \geq f_j(\vec{z}), (\vec{x},\vec{z}) \in \mathscr{Z}_j \times \overline{\mathscr{Z}_j}\right\} \mid}{\mid \mathscr{Z}_j \mid\mid \overline{\mathscr{Z}_j} \mid} \\ & \text{where, } \mathscr{Z}_j = \{\vec{x}_i \mid \lambda_i \in Y_i, 1 \leq i \leq n\}, \\ & \overline{\mathscr{Z}_j} = \{\vec{x}_i \mid \lambda_i \notin Y_i, 1 \leq i \leq n\}, \end{aligned} \tag{4.8}$$

$$F1_j = \frac{2TP_j}{2TP_j + FP_j + FN_j}, \tag{4.9}$$

$$precision_j = \frac{TP_j}{TP_j + FP_j}. \tag{4.10}$$

The *Area Under ROC Curve* (*AUC*) evaluates the predictive models regardless of the choice of threshold values. On the other hand, *F1* and *precision* use the confusion matrix values computed from the bipartitions. Considering that the labels associated with an instance are the relevant labels. Semantically, *precision* measures the fraction of relevant labels among those predicted. High precision indicates the ability of a model to correctly predict the labels, although not necessarily all of them. In turn, *F1* measures the harmonic mean between precision and recall, such that a model with a high F1 can predict the relevant labels accurately and only them, since recall measures the fraction of relevant labels that have been predicted by the total amount of relevant labels.

To assess the statistical relevance of the results, the Bayesian hierarchical correlated t-test (BENAVOLI *et al.*, 2017) is used to compare two different strategies over multiple datasets. The test outputs probabilistic decisions about which strategy is better (left, rope and right) for a particular evaluation measure. The rope is a region of practical equivalence, without any significant difference in the performance of the alternatives. In Benavoli *et al.* (2017), the authors suggest the interval $[-0.01, 0.01]$, which consists of a difference of 1% for a measure whose range is $[0, 1]$. This interval is used for the three macro-averaged evaluation measures.

Moreover, the constant label prediction CLP (Equation 4.3) and the wrong label prediction WLP (Equantion 4.5) are investigated. Given that WLP is a generalization of the missing label prediction (MLP), only the former is considered. The elimination of MLP without the elimination of WLP has no effective gain in practice.

4.4.3 Procedures and Setup

In order to understand the effectiveness of the label operations on MLC predictive results, different analyses are performed. For such, we adopted the 5x2-fold cross-validation with stratified paired folds procedure. The iterative algorithm (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011) is used for stratifying the MLC data in order to ensure similar label distribution between training and test data.

For LE and LR, a validation procedure is used to identify suitable combinations of labels to individually optimize the label measures AUC, F1 and precision. Using only the training data, the 5x2-fold cross-validation is applied, and the most probable combination is considered for the test set. Similarly, the SCUT threshold calibration (Equation 4.2) for F1 and precision is also performed using the same validation procedure.

Thus, the results are relative to the strategies BR, BR+T$_\beta$, LE$_\beta$, LE+T$_\beta$, LR$_\beta$, LR+T$_\beta$, in which the subscript $\beta = \{AUC, F1, prec\}$ indicates the optimized measure and the suffix +T

indicates the use of an optimized threshold for the respective measure.[3]

To reduce the effect that the choice of a unique base algorithm would cause in the analysis, 4 base algorithms: Decision Tree C5.0, Random Forest (RF), Support Vector Machine (SVM) and eXtreme Gradient Boosting (XGB) are considered. When they are combined with the 3 evaluation measures (AUC, F1 and precision), 12 combinations of ML algorithm and predictive performance evaluation measure are assessed.

The experiments are carried out in the R environment. The packages `mldr` (CHARTE; CHARTE, 2015a) and `utiml` (RIVOLLI; CARVALHO, 2018) provided the code for the multi-label resources used in the experiments. The implementation of the base algorithms come from packages `C50`, `randomForest`, `e1071` and `xgboost` for C5.0, RF, SVM and XGB, respectively. Default hyperparameter values were used according to their respective packages.

4.5 Results

To investigate how to overall improve the individual label performances and mitigate the label prediction problems, label operations and threshold calibration are empirically examined. Following the methodology presented in Section 4.4, Section 4.5.1 presents the upper bound results of the label operation, revealing the potential of both, LE and LR. Next, Section 4.5.2 details the results relative to the optimized tasks. Finally, Section 4.5.3 reports the label prediction problem results.

4.5.1 LE and LR Upper Bounds

Finding the right matches between labels that are able to optimize the MLC result comprises the key point of the label operation procedure. In order to understand the potential gains to be made concerning these operations, an extensive study about the combinations is performed. The results from this study also set the empirical upper bound of the LE and LR operations.

Using an oracle that knows the best combination for each label operation, Figure 12 presents the proportional performance gain compared to the BR strategy for the respective optimized measure. Each boxplot summarizes the distribution of performance gain for the 20 datasets. The optimization tasks (x-axis) are sorted from the lowest to the highest performance improvement, whereas the average improvement (y-axis) is presented on a log scale to offer better visualization of the results.

For the different base algorithms and evaluation measures investigated, both operations achieved a considerable gain in relation to BR, mainly for the F1 measure. The smallest improve-

[3] For the sake of clarity, the notation LE_β+T_β was reduced to LE+T_β, since the same measure is always used in both optimization tasks. The same is valid for the LR operation. Moreover, when the subscript can be removed without impairing the understanding of the text, only the name of the strategy is used.

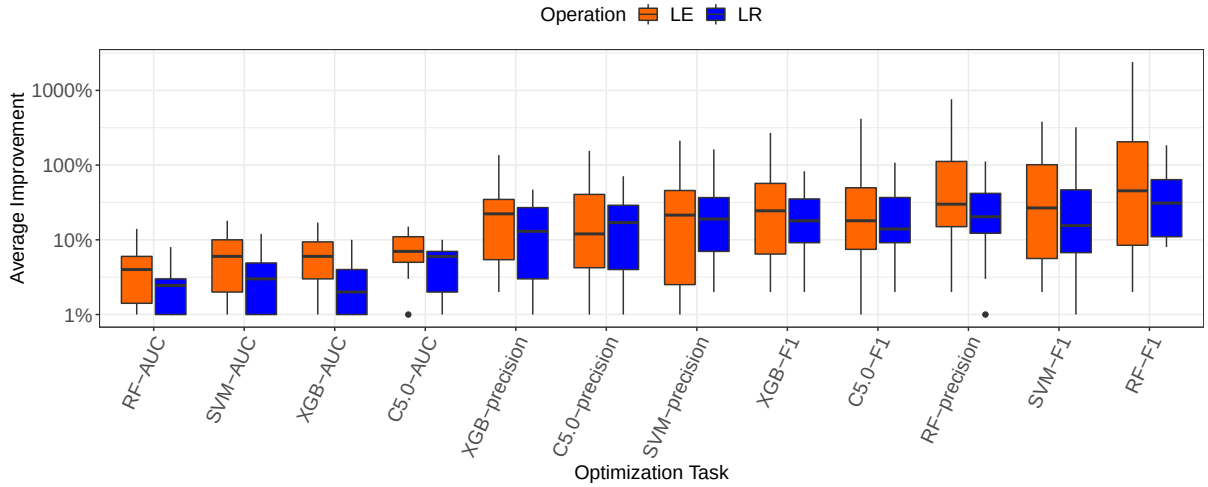


Figure 12 – Distribution of LE and LR performance gain in relation to BR for the 20 datasets considering different optimization tasks.

ments were observed for the AUC measure. Concerning the label operations, LE showed greater improvement over BR than LR. In turn, the improvement obtained by LR is more consistent among the datasets, since its interquartile range is slightly lower than LE for many tasks.

Despite the task, the Bayesian statistical test revealed that both LE and LR statistically improved the BR strategy when the best pairs of labels are used in the expansions and reductions. When compared to each other, LE and LR are statistically similar in most of the tasks. Exceptionally, only when using RF, LE_{F1} was better than LR_{F1} with a probability of 98%, according to the statistical results.

In another perspective, the oracle is used to identify the labels that can be optimized and the candidate labels that when combined can result in an improvement. Despite the fact that the previous analysis showed that LE obtains subtle better results than LR, from this analysis the LR operation showed to be more robust than the LE, at least for the AUC and F1 measures. In other words, more labels can be improved and are used to improve other labels when the operation is the LR. For most of the datasets, more than 50% of the labels can be improved, however this value is lower in terms of the number of candidate labels. The F1 is the easiest measure according to both criteria, the number of improved labels and the number of labels used to improve another.

Additionally, the similarity between the LE and LR operations is compared in two ways: *i)* the best label used to enhance each label and *ii)* the set of suitable labels able to enhance each label. The Jaccard measure is used to compare the similarity between two sets, which is given by the ratio of the size of their intersection and the size of their union.[4] Figure 13 presents the similarity in both cases. Datasets (y-axis) and optimization tasks (x-axis) are sorted according to their average similarity.

[4] $\text{Jaccard}(A,B) = (A \cap B)/(A \cup B)$

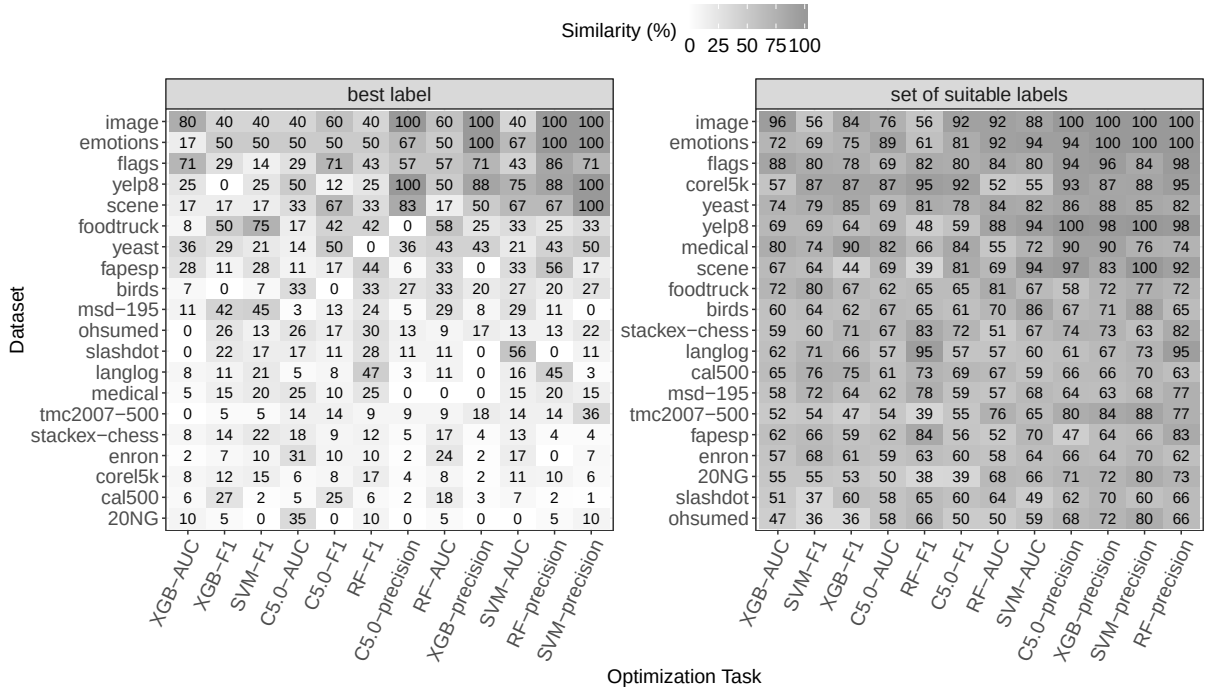


Figure 13 – Similarity between the LE and LR for each dataset and optimization task.

The best combinations of labels are different in terms of LE and LR operations. The cases in which they are more similar to each other comprise datasets with a small number of labels and tasks related to precision optimization. Precision was the measure in which the smallest number of matches between pairs of labels were found, therefore empty sets of selected labels were observed frequently. Possibly this fact justifies that precision obtained the most similar pairs of labels. This is also valid when comparing the suitable set of labels as a greater similarity between the two operations was obtained. This result indicates that for many datasets, most of the combinations between pairs of labels are able to enhance both operations, however the greatest improvements, comparing LE and LR, were usually obtained for distinct combinations.

Finally, in an attempt to measure the variance of the operations, the oracle was consulted two times using distinct seeds to sample the data into the folds. Figure 14 presents the average similarity for each task when the two runs are compared to each other. Again, the Jaccard similarity is used. The cross symbol indicates the average value from the previous comparison between LE and LR (Figure 13). It is used as a reference value for comparative purposes.

The comparison of the best pair of labels obtained a lower similarity than the set of suitable labels. According to the results, LE showed to be more robust than LR, which is observed by the more elongated boxplots, larger standard deviation and lower similarity rates observed for the LR. This is consistent with the fact that for LR, a larger set of labels are available to be combined when compared to LE. Thereby, a higher number of options leads to greater variability in the choices. Regarding the crosses used as a baseline for this comparison, the similarity between two executions of the same operation is consistently higher than between the operations.

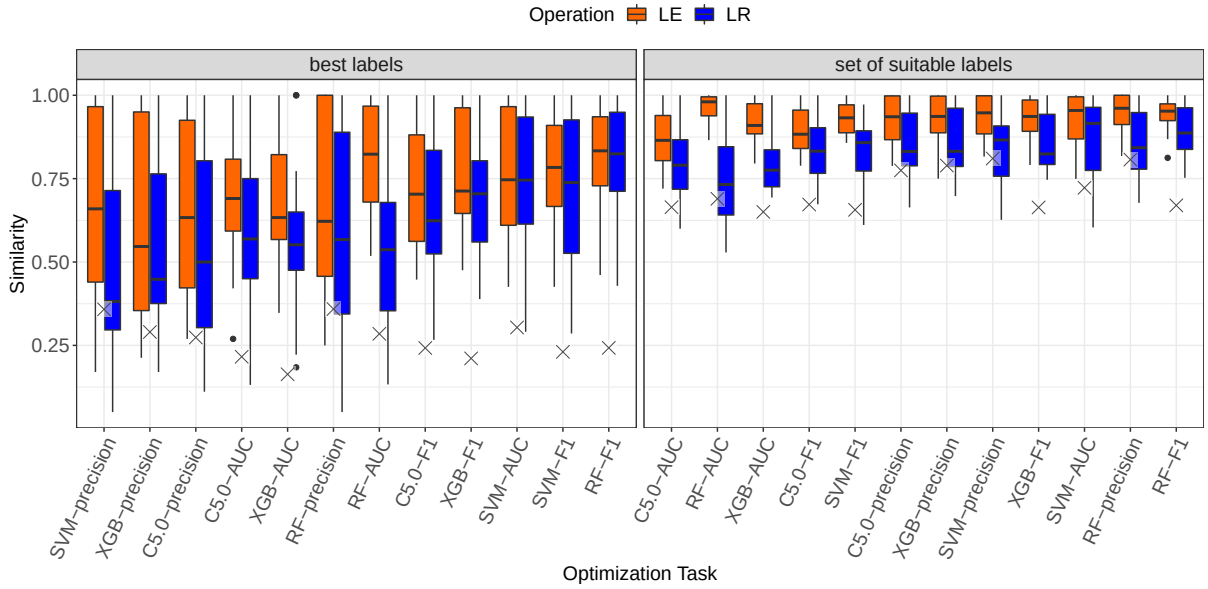


Figure 14 – Similarity between two distinct executions of the label operations.

Which can indicate that the choices of labels are not completely a matter of randomness.

However, as the similarities observed in the best label comparison were not as high as expected, the inconsistency of the choices is gauged. Table 23 presents the percentage of occurrences in which the best labels selected in a run were not present in the set of suitable labels in the other run. The percentages are small for F1 optimization, regardless of the base algorithm and operation, which is a good result. The worst result was obtained for LR_{AUC} with RF, such that 30% of the selected labels in a run were not suitable in the other run. Slightly better, LE showed once more that it was more robust than LR.

Table 23 – Percentage of occurrences in which the best label selected in a run is not present in the set of suitable labels in the other run.

Task		**LE**	**LR**	**Task**		**LE**	**LR**
C5.0	AUC	11%	13%	RF	AUC	8%	30%
	F1	6%	6%		F1	1%	2%
	precision	16%	18%		precision	11%	19%
SVM	AUC	9%	20%	XGB	AUC	9%	15%
	F1	4%	8%		F1	3%	5%
	precision	17%	23%		precision	20%	19%

In summary, the results revealed that both operations were able to enhance the predictive performance of the optimized measures. Even though they presented some variability concerning the choice of best label combination and the subset of suitable labels, for most of the datasets (and labels) the performance was improved. As the improvement was consistent over base algorithms, datasets, measures and operations, it is feasible to assume that they are not exclusively obtained by chance.

4.5.2 Optimization Tasks

The previous results showed that a considerable improvement, in different tasks, can be obtained when the right pairs of labels are combined. However, as the oracle able to indicate the best combinations is not available, a validation procedure is needed. Using a 5x2-fold cross-validation with the training data, all pairs of labels are assessed and the best combinations are used to expand/reduce the labels. If for a given label no other label improved its result in the validation procedure, then the default transformation is used for it. Moreover, a threshold selection is performed using the same procedure for BR, LE and LR when F1 and precision are the optimized measures.

The complete results for the measures macro-AUC, macro-F1 and macro-precision are, respectively, presented in Appendix D (Figures 34, 35 and 36). Table 24 summarizes the average improvement of each strategy for different optimization tasks. Between parentheses is reported the number of datasets for which the operation led to improved performance in relation to BR. The overall result is reported in the "All" base algorithm row, representing the total number of experiments for which improvement is obtained. The Bayesian statistical results are reported in Table 55. The next subsections discusses the results for each evaluation measure.

Table 24 – Average improvement of the operations compared to BR. The number of improved datasets is shown between parentheses. The underlined numbers highlight the strategy with the largest improvement.

Macro	Base	BR+T	LE	LE+T	LR	LR+T
AUC	C5.0	-	3% (16)	-	2% (17)	-
	RF	-	-0.2% (5)	-	0.3% (14)	-
	SVM	-	-0.1% (9)	-	-0.4% (16)	-
	XGB	-	0.9% (11)	-	0.6% (16)	-
	All	-	0.9% (51%)	-	0.6% (80%)	-
F1	C5.0	10% (12)	19% (12)	17% (11)	14% (17)	15% (14)
	RF	61% (18)	21% (12)	35% (14)	18% (18)	59% (17)
	SVM	45% (18)	31% (11)	38% (14)	25% (16)	44% (16)
	XGB	23% (16)	11% (11)	17% (11)	12% (17)	23% (16)
	All	35% (80%)	20% (57%)	27% (62%)	17% (85%)	35% (79%)
prec.	C5.0	3% (11)	1% (7)	3% (13)	3% (9)	5% (16)
	RF	8% (17)	-7% (4)	-3% (11)	1% (11)	7% (15)
	SVM	11% (15)	1% (7)	4% (12)	3% (8)	9% (16)
	XGB	6% (16)	-4% (4)	0.8% (13)	0.5% (8)	6% (16)
	All	7% (74%)	-2% (28%)	1% (61%)	2% (45%)	7% (79%)

4.5.2.1 Macro-AUC

AUC is a threshold independent measure, since it aggregates the results for multiple values of the threshold. Therefore, only the results from BR, LE and LR are analysed. Figure 15 presents the relative macro-AUC improvement of the LE and LR operations over the BR strategy,

as well as the difference to the corresponding upper bound (indicated by their respective symbols). This plot explains the results summarized in Table 24.

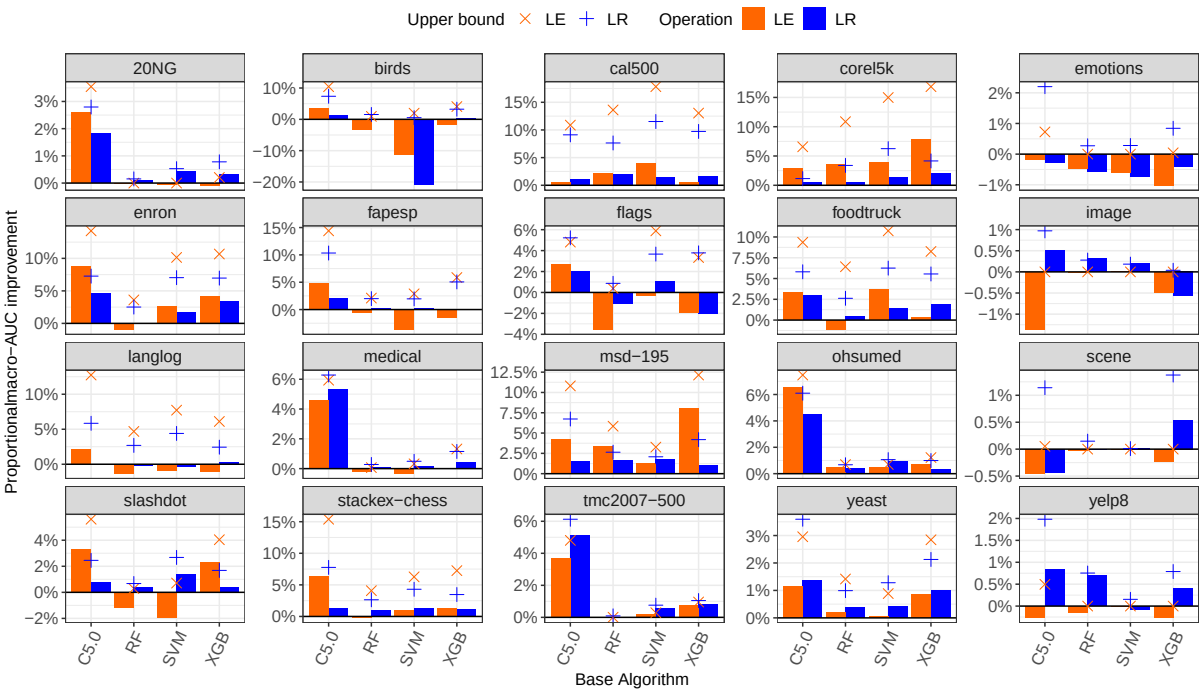


Figure 15 – Proportional macro-AUC improvement obtained by using LE and LR compared to BR.

For *macro-AUC*, both operations showed a moderated improvement when compared to BR. Slight superior, LE achieved better results, whereas LR improved more datasets regardless of the base algorithm. C5.0 is the base-algorithm that presented the most improvement. On the other hand, only 5 from 20 datasets were enhanced when LE was combined with RF. A possible explanation is due to the fact that BR with RF obtained a good performance for many datasets, such that the gap of improvement is very low. This is corroborated by the fact that the C5.0 base algorithm obtained the lowest performance for the BR strategy.

According to the results obtained by the base algorithms, the behavior of the operations varied considerably for the same dataset. For instance, with the `flags` dataset: *i)* LE is better than LR and both of them improve BR using C5.0; *ii)* both of them do not improve BR using RF and XGB; *iii)* LR is better than LE, and only the former improves BR using SVM. Only in some cases (`corel5k`, `tmc2007-500` and `yeast`) is the same behavior between LE and LR observed for all base algorithms.

Statistically, LE and LR outperform BR only with C5.0. For the other base algorithms, they are considered equivalent. Even though LR, with SVM and XGB, is better than BR in 16 datasets, the improvement is not large enough for the Bayesian test to indicate some statistical difference. Between LE and LR, no statistical difference is observed.

4.5.2.2 Macro-F1

The F1 measure, which represents the trade-off between precision and recall, in the macro-averaged version averages the performance of all labels. Figure 16 shows the relative improvement of operations compared to the BR strategy. Here, the result is consistently better, such that for many datasets the performances obtained are closer to the upper bound references. The scale of the improvement is substantially higher in the cases where the BR's performance is too low. Again, in terms of overall improvement LE is superior to LR. However, in terms of the number of datasets improved, LR is the best option, even compared to the other strategies using threshold calibration.

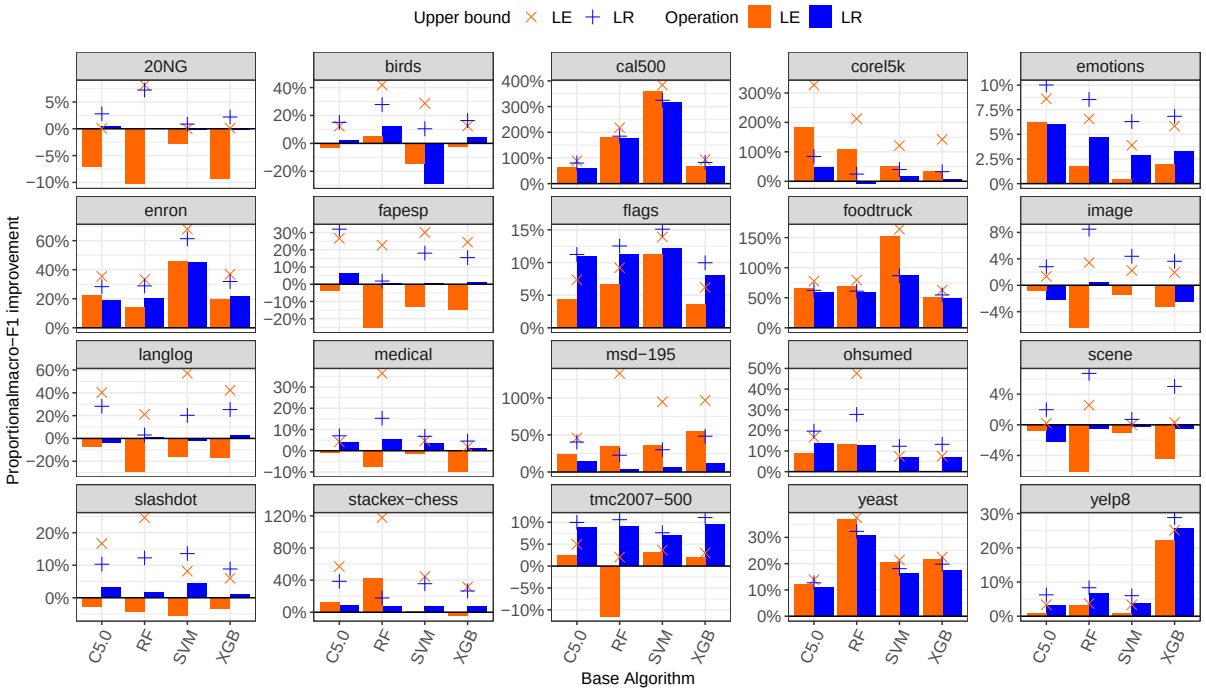


Figure 16 – Proportional macro-F1 improvement obtained by using LE and LR.

Although the macro-F1 improvement obtained from LE and LR was significant, it was clearly inferior to the threshold calibration (BR+T), with the exception of the C5.0 base-algorithm, which was least robust in this analysis. For C5.0, LE followed by LR obtained the best ranking positions, which possibly indicates that when the base algorithm does not predict good scores, label operations might work better than BR+T for macro-F1 optimization. When the label operations are combined with threshold calibration, LE and LR are also optimized for most of the cases. However, the exception is LE with C5.0.

BR achieved a very low result, regardless of the base algorithm, for some datasets: `cal500`, `corel5k`, `enron`, `foodtruck`, `langlog` and `msd-500`. From them, only `langlog` was not consistently enhanced, however their performance remains low even with the improvement. As with most of them, macro-precision and macro-recall are also low, they characterize a set of hard problems to be learned.

The Bayesian statistical test indicates that, for this measure, all label operations and threshold calibration outperform BR regardless of the base algorithm. For the other comparisons, the statistical test indicates that the results are statistically different between C5.0 and the other algorithms. Concerning C5.0, all other strategies are similar to each other. For the rest: BR+T and LR+T $\prec$ LR $\prec$ LE+T $\prec$ LE.[5] The exception is LE+T $\prec$ LR with SVM.

4.5.2.3 Macro-precision

Compared to the previous measures, LE and LR obtained improvements in a smaller number of datasets. Moreover, the improvement is quite low, mainly for LE such that the highest differences to the upper bound values are observed. Figure 17 shows the relative improvement of LE and LR in comparison to BR. For many cases, a decline in performance is obtained, revealing that the validation procedure failed to find suitable pairs of labels for this measure.

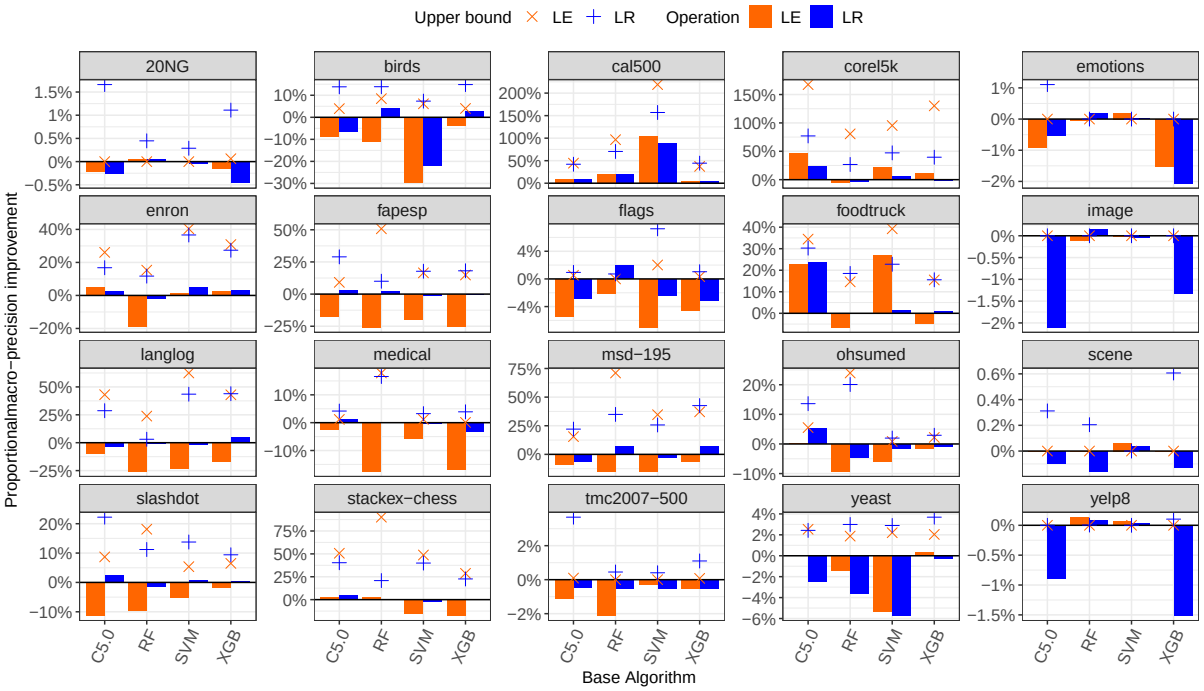


Figure 17 – Macro-precision improvement obtained by using LE and LR.

The best ranking positions were obtained for BR+T and LR+T indiscriminately from the base algorithm. The statistical test results indicate that only strategies using threshold calibration were able to statistically outperform BR, such that BR+T, LE+T and LR+T $\prec$ BR, for all base algorithms. Concerning the other strategies, BR+T and LR+T $\prec$ LE and LR. For RF, BR+T and LR+T $\prec$ LE+T $\prec$ LE, whereas for the rest LE+T $\prec$ LE and LR.

[5] A $\prec$ B means that A outperforms B.

4.5.3 Label Prediction Problems

Two label prediction problems are investigated in this study. The CLP and WLP, respectively, indicate the proportion of labels constantly predicted and the proportion of labels that are always predicted wrongly. Table 25 presents the BR performance for CLP and WLP, which represents the baseline for this analysis. For a better visualization, the no occurrence of problems (value 0) is omitted from the table, as well as the datasets with no problems for all base algorithms were removed. According to the results, WLP is a more recurrent problem than CLP, such that for most of the datasets in the table, the WLP occurred indiscriminately for all base algorithms. Differently, the CLP is observed only in 4 datasets and 3 base algorithms.

Table 25 – Label prediction problem results for the BR strategy. The datasets with no problems were removed from the table. An empty cell indicates the occurrence of no problem.

Dataset	CLP				WLP			
	C5.0	RF	SVM	XGB	C5.0	RF	SVM	XGB
birds					0.093	0.173	0.293	0.060
cal500	0.009	0.001	0.028		0.394	0.615	0.837	0.377
corel5k					0.918	0.829	0.814	0.759
enron					0.517	0.374	0.581	0.483
fapesp					0.178	0.311	0.189	0.139
flags	0.214		0.229			0.043	0.057	
foodtruck			0.017		0.575	0.517	0.708	0.475
langlog					0.437	0.576	0.497	0.326
medical					0.110	0.175	0.095	0.030
msd-195					0.295	0.497	0.376	0.371
ohsumed					0.243	0.196		0.035
slashdot					0.428	0.144	0.122	0.100
stackex-chess					0.565	0.622	0.428	0.296
tmc2007-500					0.027			
yeast		0.029			0.100	0.143	0.143	0.129
Total	2	2	3	0	14	14	13	13

Some datasets (cal500, corel5k, foodtruck, langlog and stackex-chess) had more than 50% of their labels wrongly predicted, on average. Usually overlooked, this is a considerable amount of labels to be neglected. For instance, the dataset corel5k has 218 labels. According to the results, BR with C5.0 predicted approximately 18 correctly, on average. This explains the macro-F1 and macro-precision result of 0.014 and 0.04, respectively.

Considering that the threshold calibration and the label operations can mitigate both label prediction problems, 12 distinct strategies are analysed compared to BR. They are BR with threshold calibration: BR+T_{F1} and BR+T_{prec}; Label expansion: LE_{AUC}, LE_{F1} and LE_{prec}; Label reduction: LR_{AUC}, LR_{F1} and LR_{prec}; LE and LR combined with threshold calibration: LE+T_{F1}, LE+T_{prec}, LR+T_{F1} and LR+T_{prec}.

Ideally, a good strategy should reduce WLP without impairing CLP and vice-versa. Therefore, Figure 18 presents the strategies' performance considering both measures averaged. The strategies are sorted according to their median performance for each base algorithm.

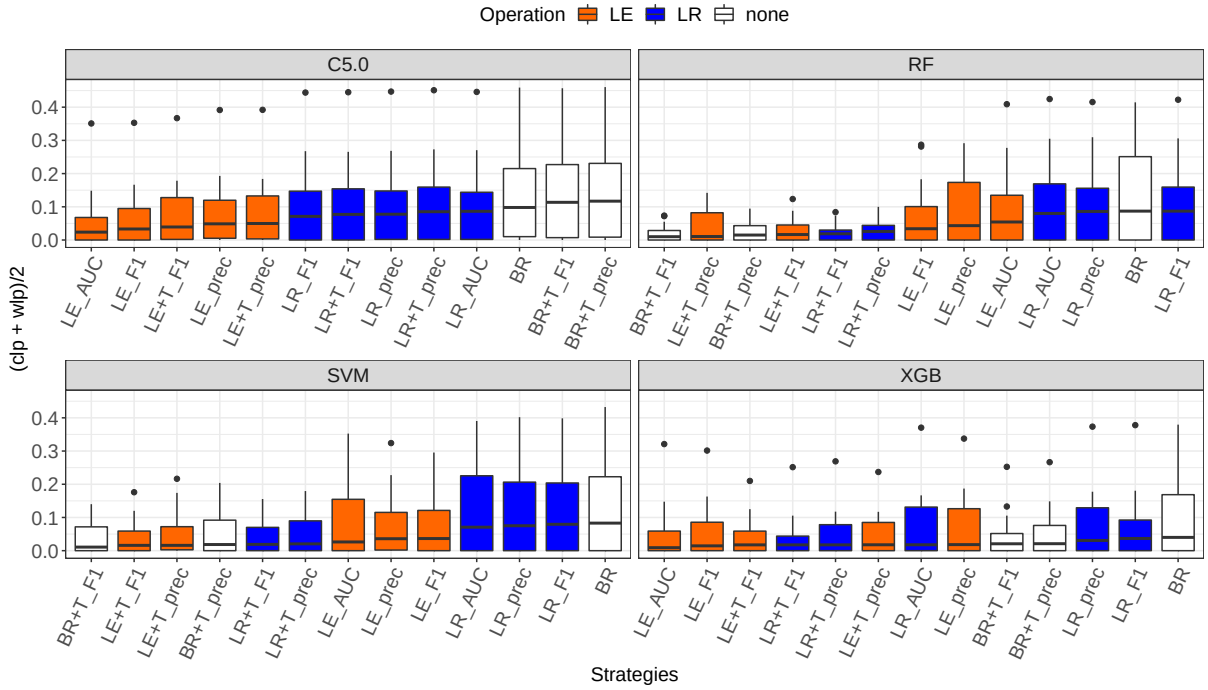


Figure 18 – Strategies' performance according to their trade-off between CLP and WLP.

Surprisingly, LE_{AUC} with C5.0 and XGB obtained the best combined performance, overcoming even F1 and precision optimizations. For C5.0, the strategies are completely segmented such that all variations of LE outperformed LR, that in turn, outperformed BR and BR+T. For RF and SVM, the best options mix the threshold calibration plus LE optimizing F1 and precision.

To summarize and quantify the improvement obtained for each dataset, Table 26 presents the CLP and WLP results from the best strategies previously identified for each base algorithm. Therefore, LE_{AUC} is used with C5.0 and XGB, whereas BR+T_{F1} is used with RF and SVM. To make the comparison with Table 25 easier, values similar to BR were removed and the underline markup highlights the cases in which the problem is increased instead of solved.

Despite the fact that only in a few datasets (`ohsumed`, `tmc2007-500` and `yeast`) the WLP was completely solved, the reduction observed is considerable, in many cases. For instance, BR+T_{F1} with RF reduced from 0.829 to 0.122 the WLP of `corel5k`. In the number of labels, it consists of more than 150 labels that started to be correctly predicted for some test instances. As expected, the improvement in the WLP led to a slight decline in CLP, mainly for RF and SVM using threshold calibration. On the other hand, compared to C5.0 and XGB, they obtained a greater improvement in terms of WLP.

Without looking specifically at the amount of improvement, but at the ability of the strategies to improve one measure without impairing another, three scenarios are analyzed in Table 27. Considering the number of datasets that are improved and deteriorated compared to BR, the column "↑⇓" indicates the cases that one measure is improved without impairing the other. The column "↑↓" indicates the cases that one measure is improved and the other is damaged. The

Table 26 – Label prediction problem performance when the best strategy is selected for each base algorithm. An empty cell indicates that the result is similar to BR. The underlined numbers show a degradation in performance when compared to BR.

Dataset	CLP				WLP			
	C5.0	**RF**	**SVM**	**XGB**	**C5.0**	**RF**	**SVM**	**XGB**
birds			0.087		0.040	0.033	0.033	0.013
cal500	0.013	0.010	0.156		0.106	0.055	0.124	0.091
corel5k		0.022	0.004		0.702	0.122	0.276	0.642
enron			0.002		0.055	0.026	0.129	0.133
fapesp					0.117	0.011	0.011	0.106
flags	0.186	0.014	0.129	0.014		0.000	0.014	
foodtruck			0.075		0.117	0.058	0.083	0.167
langlog			0.011		0.297	0.147	0.134	0.295
medical					0.015	0.010	0.015	0.015
msd-195					0.082	0.042	0.061	0.113
ohsumed					0.004	0.000		0.000
slashdot					0.267	0.044	0.028	0.061
stackex-chess					0.271	0.110	0.146	0.215
tmc2007-500					0.000			
yeast	0.014	0.050	0.007		0.014	0.007	0.000	0.021
Strategy	LR_{AUC}	$BR{+}T_{F1}$	$BR{+}T_{F1}$	LR_{AUC}	LR_{AUC}	$BR{+}T_{F1}$	$BR{+}T_{F1}$	LR_{AUC}

column "↗↓" indicates the cases where one measure is damaged without enhancing the other. The strategies are sorted according to a score over all base algorithms, given that each occurrence of ↗↓ = -1, ↑↓ = 1 and ↑↙ = 2. The higher the score the safer the use of the strategy is.

Table 27 – Number of datasets improved and deteriorated compared to BR concerning the trade-off between CLP and WLP. The arrows ↑↗↓↙ indicate that a measure is improved, not improved, deteriorated and not deteriorated, respectively.

	Strategy	C5.0			RF			SVM			XGB		
		↑↙	↑↓	↗↓	↑↙	↑↓	↗↓	↑↙	↑↓	↗↓	↑↙	↑↓	↗↓
1	$LE{+}T_{prec}$	13	2	0	14	0	0	11	2	2	12	1	1
2	LE_{AUC}	13	2	0	9	3	2	10	3	0	13	0	1
3	$LE{+}T_{F1}$	9	6	0	12	2	0	8	5	0	10	3	1
4	LE_{prec}	13	2	0	10	3	1	10	3	2	11	0	2
5	LE_{F1}	12	2	1	10	3	1	9	4	0	11	1	1
6	LR_{AUC}	12	1	1	8	3	3	10	2	1	13	0	1
7	$LR{+}T_{prec}$	3	9	1	13	1	0	11	2	1	6	7	0
8	LR_{prec}	12	1	1	7	3	2	8	3	2	12	1	1
9	$BR{+}T_{prec}$	1	10	0	13	1	0	10	3	1	5	8	0
10	$LR{+}T_{F1}$	1	13	0	10	4	0	4	9	0	5	8	1
11	$BR{+}T_{F1}$	0	12	2	10	4	0	6	7	0	4	9	1
12	LR_{F1}	10	3	1	6	3	5	7	3	3	11	1	2

The best strategy, according to this criteria, over all base algorithms is the $LE{+}T_{prec}$. It is followed by other LE strategies using different optimization measures and then some LR strategies. The BR+T strategies appear in positions 9 and 11 in this ranking. The label operations, mainly LE, showed a good balance between them to reduce WLP without increasing CLP, regardless of the base algorithm. On the contrary, using a threshold calibration procedure is

suitable only for robust base algorithms such as RF and SVM and it tends to reduce WLP increasing CLP. Concerning the evaluation measure, the optimization of precision showed to have a better compromise between the CLP and WLP.

4.6 Discussion

The results will now be analyzed according to 2 perspectives: the label operation strategies, LE and LR, as an optimization procedure and the ability of the strategies to reduce the label prediction problems. It is worth highlighting that this experiment considered a high number of datasets and, mainly, base algorithms compared to other MLC experiments. Therefore, the analysis is not restricted by a bias of a unique base algorithm, enabling a broader understanding of the investigated strategies.

Empirically, it was shown that LE and LR can enhance BR for different base algorithms and evaluation measures (Figure 12). However, in practice the improvement obtained with the validation procedure adopted was quite far from the computed upper bound (Figures 15, 16 and 17). The variability of the choices became, in fact, a source of instability to the label operations, mainly for the optimization of macro-precision. An additional test was performed using a different validation procedure (5-folds CV) with a subset of datasets, but similar results were obtained. A possible explanation is due to the fact that many labels are associated with a few instances, which makes the variability intrinsic to the nature of the MLC datasets.

In an exploratory analysis, we tried to find heuristics to restrict the search space and possible rules to define the label combinations in a deterministic way. Thus, the correlation between the improvement and characterization measures, such as label frequency, imbalance ratio, co-occurrence and distance between labels was assessed. It was expected, for instance, that the frequency and/or the label imbalanced ratio would affect the optimization procedure, however no strong correlation was found. Other approaches, such as meta-learning (BRAZDIL *et al.*, 2017), association rules (AGRAWAL; SRIKANT, 1994) and clustering techniques (ROKACH, 2010) were not employed at this moment, however their investigation is suggested as future work. Potentially, they can reduce the complexity of the solution and provide a gain in performance, which are currently the main limitations of LE and LR.

To discard the possibility that the combinations are a result of chance, we looked for semantic combinations between pairs of labels. Analyzing the `fapesp` dataset, in which interdisciplinary scientific papers are labeled with distinct branches of knowledge, it was found that using the label *physics* to expand the label *astronomy*, enhances its F1 and precision. In this particular case, the opposite is true for the LR operation, such that by using the label *astronomy* to reduce the label *physics*, the latter is enhanced. It is reasonable to assume that there is a relationship of specialization/generalization between these two labels. Although the expansion of the specialized label with the generic label does not seem intuitive, the reduction of the instances

related to the specialized label that is not tagged with the generic label looks completely valid. Other cases were also observed, for instance, the expansion of the label *medicine* with the labels *chemistry* or *genetics*.

Overall, the relations vary according to the base algorithm, evaluation measure and label operation regardless of the dataset. We believe that it is mainly due to the bias of the base algorithm, that may be more or less susceptible to the noise introduced (or removed) by the label operations and the transformations caused in the hyperspace for them.

When compared and combined with the threshold calibration, label operations showed competitive results. First of all, LE and LR can optimize score based measures, such as AUC. Furthermore, they can work better for base algorithms that are not so robust, such as C5.0, differently from the threshold calibration that worked better for RF and SVM. Finally, the number of datasets improved was usually greater with label operations than threshold calibration, mainly for LR.

Concerning the label prediction problems, all the investigated strategies were able to reduce the WLP, which were the most recurrent problems observed. Broadly speaking, LE worked satisfactorily for all base algorithms, even when it did not achieve a good performance in the optimized measure. Furthermore, LE combines good performance for WLP without impairing the CLP, differently from the threshold calibration.

From the results, it can be observed that an undesirable high WLP is strongly related to a very low macro-F1 and macro-precision values. Most of the datasets with a high WLP have a subset of infrequent labels. For these labels, a threshold calibration may not work well (FAN; LIN, 2007). Label operations face the same problem since a rare label is more susceptible to the variability of the choices, because there are few instances in the validation set. But differently from the threshold calibration, the upper bound analysis showed that it is possible to improve such labels provided that the right choice of labels is made.

Therefore, more effort must be exerted to boost the choice of the right label to operate. It is reasonable to assume that by choosing the right labels pairs, both operations can mitigate the WLP even more. Other alternatives such as exploring more complex operations, combining LE and LR with distinct MLC strategies and exploring strategies able to deal with the labels' imbalance problem are possible alternatives to solve the WLP. This is still an incipient problem, and therefore the results reported in this article comprise the first in-depth investigation to solve the label prediction problems.

4.7 Conclusion

This study has shown that label operations and threshold calibration are suitable solutions to reduce label prediction problems, particularly the WLP, in which some labels are never

correctly predicted. For such, 20 datasets and 4 base algorithms were empirically evaluated considering the optimization of three different evaluation measures: AUC, F1 and precision. It comprises the first investigation of the matter in the MLC literature.

By optimizing the threshold, when the base algorithm predicted good labels scores, BR obtained better macro-F1 and macro-precision than both label operations. On the other hand, LE and LR improved BR moderately, but consistently, for all base algorithms considered. Furthermore, by using BR, the score-based measures were optimized such as the non-trivial AUC measure.

Specifically to the label operations, finding the best set of labels to combine to the given label is a hard problem. We explored a simple procedure, consisting of selecting the best label to operate on the given label, using a search guided by the predictive performance on a validation set. It is computationally expensive and may lead to suboptimal solutions. Several heuristic search methods could be used to improve it but this is suggested as future work. The operations were evaluated in terms of predictive performance as well as its impact on the label prediction problems.

Concerning the label prediction problems, the WLP showed to be the problem to be solved. Although no strategy completely removes the problem for most datasets, LE showed the best trade-off regarding achieving good WLP performance without harming CLP.

Some issues remain open, such as how good the label operations might be when combined with other MLC strategies, for instance ECC (READ *et al.*, 2011) and DBR (MONTAÑÉS *et al.*, 2014). For such, it is necessary to find alternatives to reduce the search space of candidate labels and increase the confidence of the choices. In another perspective, using strategies designed to deal with MLC imbalance (CHARTE *et al.*, 2014; CHARTE *et al.*, 2019) to solve the WLP looks like a promising direction.

CHAPTER

5

RECOMMENDING LABEL OPERATIONS FOR MULTI-LABEL CLASSIFICATION

Collaborating authors

Carlos Soares
Fraunhofer AICOS and LIAAD-INESC TEC, University of Porto, Porto, Portugal

Bernhard Pfahringer
University of Waikato, Hamilton, New Zealand

André C. P. L. F. de Carvalho
University of São Paulo, São Carlos, Brazil

Abstract

In multi-label learning, a task contains instances related to multiple concepts simultaneously, the labels. However, recent studies showed that some labels have never been correctly predicted regardless of the algorithm, which is identified as the Wrong Label Prediction (WLP). Alternatives to mitigating this problem consist of using label expansion and reduction operations. By modifying the instances related to a given label, an evaluation measure can be optimized, and consequently, WLP can be reduced. Nevertheless, all pairs of labels must be combined in order for the best matches to be found, which requires a myriad of computation resources. To handle this issue, meta-learning is used to previously detect the occurrence of the WLP and to recommend when an operation can optimize a given label. The empirical results show that it is possible to previously identify the label prediction problem and also to reduce the computational cost of the label operations by selecting the right labels to operate. Thus, operations can be more

competitive when compared to other strategies, mainly the label expansion that can better reduce the WLP.

5.1 Introduction

Multi-label learning deals with classification tasks in which instances are simultaneously classified into more than one class (CARVALHO; FREITAS, 2009; TSOUMAKAS; KATAKIS; VLAHAVAS, 2010; GALINDO; VENTURA, 2014). Each class is called 'label' and represents a specific concept from the task's domain. Distinct kinds of applications related to text (KLIMT; YANG, 2004; PESTIAN *et al.*, 2007), multimedia (DUYGULU *et al.*, 2002; ZHOU; ZHANG, 2006; BRIGGS *et al.*, 2013) and biology (ELISSEEFF; WESTON, 2001) are intrinsically multi-label.

In a recent study (RIVOLLI; SOARES; CARVALHO, 2018a), we observed that for many datasets some labels have never been correctly predicted (Wrong Label Prediction - WLP), and others are always predicted (Constant Label Prediction - CLP), which occur less frequently. These problems are observed regardless of the strategy and base algorithm used. To deal with them, we proposed the label expansion and reduction operations that proved to be a reasonable alternative to mitigate mainly the WLP.

The label expansion (LE) increases the number of instances associated with a label, whereas the label reduction (LR) reduces the number of instances that are not associated with it. Both use another, possibly related, label to guide the transformation, indirectly exploring the labels' dependency and obtaining more balanced datasets. The pairs of labels are selected by optimizing a label evaluation measure, which consequently reduces the label prediction problems. However, this process can demand a high computation cost, since each pair of labels should be assessed in a validation procedure.

Aiming to understand the occurrence of the problems and reduce the cost of the operations, meta-learning (MtL) is used as a recommendation system. MtL supports the automatic selection of machine-learning algorithms by using knowledge from the previous application of such algorithms to several datasets (BRAZDIL *et al.*, 2009). In a nutshell, a meta-base is created using descriptive characteristics (RIVOLLI *et al.*, 2019) extracted from similar tasks along with the identification of the meta-target recommended. A meta-model is induced from this meta-base, so that it can perform predictions for unseen cases, as well as provide an explanation for the learned data.

In this paper, two MtL tasks are investigated. First, the labels that can never be correctly predicted are learned, so that some actions can be taken to deal with them in advance. Moreover, a meta-model is induced to predict if each label can be improved by using a label operation. In this case, not all labels will be exhaustively evaluated during the validation procedure, but only the most likely ones indicated by the meta-learner.

Therefore, the contributions of this paper lead to a better understanding of the WLP using an MtL system, and reducing the costs involving the label operation optimization. Using an empirical approach, different experiments and analyses are performed in order to validate our claims. The results show that it is possible to predict both scenarios with a certain confidence. For instance, the occurrence of the WLP is predicted with more than 90% of accuracy by a relatively simple learning model. The number of labels considered in the validation procedure was reduced to more than 50% in some scenarios without impairing the operations' performance.

Concerning the MtL research, the investigated problems pose an unusual challenge, which is the presence of highly correlated meta-instances. Given that each meta-instance represents a label and many labels come from the same multi-label dataset, they share more information between them than labels from different datasets. This fact requires some changes in the experimental procedure, as well as caution when analyzing the results.

The rest of the paper is organized as follows: Section 5.2 formally defines the main concepts relevant for multi-label learning and meta-learning. Section 5.3 describes the experimental design, including datasets, procedures, evaluation and tools. Section 5.4 presents the empirical results from the meta-learning experiments in both levels: base and meta. Finally, in Section 5.5 conclusions are drawn concerning relevant findings from the experimental study and future work directions.

5.2 Background

This section presents the main concepts relevant for understanding this work well, which comprises defining multi-label learning, formalizing the strategies used in the empirical study, as well as the label expansion and reduction operations. Finally, MtL is presented.

5.2.1 Multi-label classification

In multi-label classification (MLC) tasks, an instance can be simultaneously classified into many of the existing labels (CARVALHO; FREITAS, 2009). The learning process consists of inducing a predictive model $h: \mathscr{X} \rightarrow \mathscr{Y}$ from a set of labeled instances $\mathscr{D} = \{(\vec{x}_1, Y_1), ..., (\vec{x}_n, Y_n)\}$. In this equation, $\vec{x}_i \in \mathscr{X}$ is a vector with characterization features that describes an instance and $Y_i \subseteq \mathscr{Y}$ are the set of labels associated with it, such that $\mathscr{Y} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ is the set of all q labels λ_j, which represent concepts from a given domain. Without loss of generality, the labels associated with the i^{th} instance, also called label set, can be seen as a binary vector $y_i = (y_{i1}, y_{i2}, \cdots, y_{iq}) \in \{0,1\}^q$, where $y_{ij} = 1$ *iff* $\lambda_j \in Y_i$ and $y_{ij} = 0$ *iff* $\lambda_j \notin Y_i$.

The strategies to induce a predictive model h for an MLC task are organized into two groups (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010): *problem transformation* and *algorithm adaptation*. The former transforms the original multi-label dataset into a set of single-label

datasets, where conventional machine learning algorithms can be used. For this reason, they can be seen as *algorithm independent* (CARVALHO; FREITAS, 2009). The latter modifies existing machine learning algorithms to intrinsically support the multi-labeled data.

The transformation process can be performed using one-versus-all, one-versus-one and multi-class approaches. The one-versus-all strategies are characterized by using at least one binary dataset per label, whereas the one-versus-one strategies generate $q(q-1)/2$ binary datasets, combining all pairs of labels. The multi-class strategies use the label sets (or part of them) as classes, such that a high number of classes are commonly obtained.

5.2.2 MLC Strategies

Several MLC strategies have been proposed to support classification tasks with multiple labels. In this work, some of them are selected to be empirically assessed.

The Binary Relevance (BR) strategy (BOUTELL *et al.*, 2004) uses the one-versus-all transformation in its simplest way. Each binary dataset $\mathscr{D}'_j = \phi(\mathscr{D}, \lambda_j)$ is related to the label λ_j. The instances associated with the λ_j are labeled with the class value "1", and the others with the class value "0", such that

$$\begin{aligned} \mathscr{D}'_j =& \{(\vec{x}_i, I(\lambda_i \in Y_i)) \mid (\vec{x}_i, Y_i) \in \mathscr{D}\}, \text{ where} \\ I(\cdot) =& \begin{cases} 1 & \text{if the predicate is true,} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{5.1}$$

BR uses the dataset $\mathscr{D}'_j$ to induce a binary model θ_j for each label λ_j. The prediction is performed using the values of all binary models as follows:

$$h_{br} = \{\lambda_j \mid \theta_j(\vec{x}) = 1,\ 1 \leq j \leq q\}.$$

As drawbacks, BR does not explore the relationship between labels and usually produces imbalanced binary datasets (ZHOU; TAO; WU, 2012; ZHANG *et al.*, 2018). In order to explore the labels' dependencies, *Dependent Binary Relevance* (DBR) (MONTAÑÉS *et al.*, 2014) is a full stacking strategy that uses two rounds of binary transformation. In the first round, the process is similar to the BR strategy. In the second round, the input space is augmented by the labels' information obtained in the first round. To illustrate how this works, let ψ_j be a function that removes label j from vector y, such that

$$\begin{aligned} \mathscr{D}''_j =& \left\{([\vec{x}_i, \psi_j(y_i)], y_{ij}) \mid 1 \leq i \leq n\right\}, \text{ where} \\ \psi_j(y) =& (y_1, \cdots, y_{(j-1)}, y_{(j+1)}, \cdots, y_q). \end{aligned} \tag{5.2}$$

DBR predicts the labels using the second round binary models that use the labels obtained from the first round binary models. Using the ψ function presented in Equation 5.2, the prediction

is obtained as follows:

$$h_{dbr} = \{\lambda_j \mid \theta_j^{''}([\vec{x}_i, \psi_j(h_{br}(\vec{x}_i))]) = 1,\ 1 \leq j \leq q\}.$$

An alternative to tackle the multi-label imbalance problem is the *REsampling MultilabEl datasets by Decoupling highly ImbAlanced Labels* (REMEDIAL) algorithm (CHARTE *et al.*, 2015b). Although it is not a traditional oversampling method, empirical results show that its use, when combined with the BR strategy, can improve the performance of the minority labels. Specifically, it is suitable in scenarios in which the majority labels are present in the instances related to the minority labels. In short, such instances are duplicated and the labels are reorganized according to their frequency, such that one instance will have the most frequent labels and the other instance the least frequent labels.

Ensemble of Classifier Chains (ECC) uses bagging, chooses different random subsets of the attributes for each bagging iteration and induces Classifier Chain (CC) models with a random order of chains (READ *et al.*, 2011). The CC strategy (READ *et al.*, 2009; READ *et al.*, 2011) increases the original input space of the transformed dataset for a given label with the values of all previous labels organized in a chain. Thus, the dataset is transformed as follows:

$$\mathscr{D}'_j = \left\{([x_i, y_{i1}, y_{i2}, \cdots, y_{i(j-2)}, y_{i(j-1)}], y_{ij}) \mid 1 \leq i \leq n\right\}.$$

The models increase their input space by adding $j-1$ new attributes, where j is the position of the respective label in the chain. During the prediction phase, as the labels are predicted, their values are used to increase the input space, as shown next

$$\begin{aligned} h_{cc} &= \{\lambda_j \mid \hat{y}_j = 1,\ 1 \leq j \leq q\}, \text{where} \\ \hat{y}_j &= \theta_j([x, \hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{(j-2)}, \hat{y}_{(j-1)}]). \end{aligned} \tag{5.3}$$

To illustrate how ECC computes predictions, let m be the number of models in the ensemble and φ_i a function for selecting a random subset of attributes:

$$\begin{aligned} h_{ecc} &= \{\lambda_j \mid \left(\frac{1}{m}\sum_{l=1}^{m} \hat{y}_{lj}\right) > \tau,\ 1 \leq j \leq q\}, \text{where} \\ \hat{y}_l &= h_{cc}^l(\varphi_l(x)), \end{aligned}$$

$\hat{y}_{lj}$ is the predicted value of the CC model l for the label λ_j and τ is a threshold value.[1] Different chains are used to avoid the influence that choosing an inappropriate chain could have on the results.

Finally, *RAndom k-labEL sets* (RAkEL) is an ensemble of *Label Powerset* (LP) models (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011). By using a multi-class transformation,

[1] It can either be a predefined value, such as 0.5 (READ *et al.*, 2011) or dynamically defined using the cardinality value of the training dataset (READ *et al.*, 2009).

each member of the ensemble maps a small subset of labels into classes. RAkEL obtains m transformed datasets, in which the subset of the label set of each instance is mapped to its class, such that

$$D'_l = \{(\vec{x}_i, LP(Y_i \cap C_l)) \mid 1 \leq i \leq n\}, \text{ where}$$
$$LP(\mathscr{L}) = \sum_{j=1}^{q} I(\lambda_j \in \mathscr{L}) 2^{j-1},$$
$$\mathscr{C} = \{C_l \mid C_l \subseteq \mathscr{Y},\ |C_l| = k,\ 1 \leq l \leq m\}$$

and C_l is one of the m subsets of k labels randomly selected from $\mathscr{Y}$. Here, m and k are required hyperparameters.

The internal LP models $\{\theta_1, \cdots, \theta_m\}$ induced from the respective datasets are used to compute a positive or negative vote for each label. Given a new instance, RAkEL uses the ratio of positive votes to decide if the label is relevant, such that

$$h_{RAkEL} = \{\lambda_j \mid \frac{T_{\lambda_j}}{T_{\lambda_j} + F_{\lambda_j}} \geq \tau, \lambda_j \in \mathscr{L}\}, \text{ where}$$

$$\theta_l(\vec{x}) = \begin{cases} T_{\lambda_j} = T_{\lambda_j} + 1, & \text{iff } \lambda_j \in S \subseteq C_l, \theta_l(\vec{x}) = LP(S) \\ F_{\lambda_j} = F_{\lambda_j} + 1, & \text{iff } \lambda_j \in C_l. \end{cases}$$

The default threshold suggested by RAkEL's authors is $\tau = 0.5$. The approximate number of votes for each label is given by mk/q, thus the performance of the strategy is influenced by the definition of the m and k values.

5.2.3 Label operation

Label operation is a modification of the one-versus-all transformation, such that instances and/or target values are specifically modified for each label in order to optimize a given evaluation measure. Illustratively, a label operation can be seen as a one-versus-some transformation. During the binary transformation of each label, not all instances related to the other labels will be labeled with the class "0". Different operations can define alternatives to guide the transformation.

A transformation can be defined as $\phi(\mathscr{D}, \lambda_j, \lambda_k)$ such that the label λ_k will be used to modify the binary dataset $\mathscr{D}'_j$. Two operations, expansion and reduction, are investigated in this study.

The Label Expansion (LE) operation between two labels $(\lambda_j + \lambda_k)$ uses instances labeled with any of them as being related to the λ_j, for the transformation of dataset D'_j. Consequently, it increases the number of instances associated to the expanded label λ_j (class value "1") and reduces the number of instances with the class value "0". Formally, LE transformation can be defined as

$$\phi_{LE}(\mathscr{D}, \lambda_j, \lambda_k) = \{(\vec{x}_i, I(\lambda_j \in Y_i \vee \lambda_k \in Y_i)) \mid (\vec{x}_i, Y_i) \in \mathscr{D}\}.$$

Despite the fact that the transformation is symmetric ($\lambda_j + \lambda_k = \lambda_k + \lambda_j$), the result is not, since the learning model induced from the transformed dataset will be used to predict different labels. Thus, λ_k may be used to expand λ_j to enhance an evaluation measure, but the opposite does not necessarily happen.

The Label Reduction (LR) operation between two labels ($\lambda_j - \lambda_k$) removes the instances associated to the λ_k that are not related to the λ_j, for the transformation of dataset D'_j. It reduces the number of instances associated to class “0" without changing the number of instances associated to class “1". Formally, LR transformation can be defined as

$$\phi_{LR}(\mathscr{D}, \lambda_j, \lambda_k) = \{(\vec{x}_i, I(\lambda_j \in Y_i)) \mid (\vec{x}_i, Y_i) \in \mathscr{D}, (\lambda_j \in Y_i \vee \lambda_k \notin Y_i)\}.$$

The LR transformation is asymmetric ($\lambda_j - \lambda_k \neq \lambda_k - \lambda_j$), thus the same pair of labels can result in two different datasets according to the label that is reduced. Nevertheless, the same rationale is true here, such that λ_k may be used to enhance an evaluation measure for label λ_j, but the opposite does not necessarily happen.

Despite being simple, both label operations cannot be applied randomly. They require a procedure to identify the labels that can be expanded/reduced and the labels that can be used to expand/reduce other labels. A validation procedure can test several pairs of labels to identify the best combinations. It requires a binary evaluation measure β,[2]. Assuming that each label is independent and β is maximized, the procedure is performed in the following way

$$\arg\max_{\lambda_k} \beta(\theta_j(\mathscr{D}_v), \mathscr{D}_v) \quad | \quad \phi(\mathscr{D}_t, \lambda_j, \lambda_k) \rightarrow \theta_j, \tag{5.4}$$

where θ_j is the induced learning model for the label λ_j, $\mathscr{D}_t$ and $\mathscr{D}_v$ are, respectively, the training and validation dataset. This procedure can be used during the transformation and applied for each label. In the worst case, when $\lambda_k = \varnothing$, the default one-versus-all transformation (Equation 5.1) is applied.

5.2.4 Meta-learning

Meta-Learning (MtL) has been largely used in recent years to support the automatic selection of algorithms and define the configuration process (HUTTER; KOTTHOFF; VANSCHOREN, 2019). By using knowledge from the previous applications (BRAZDIL *et al.*, 2009), a meta-learning system can, for instance, recommend a suitable algorithm for a new problem (ALI; SMITH, 2006; WANG; SONG; ZHU, 2015); estimate the performance of different algorithms (LEITE; BRAZDIL, 2005; GARCIA *et al.*, 2016; BILALLI *et al.*, 2018); and define machine-learning pipelines (MANTOVANI *et al.*, 2015; MANTOVANI *et al.*, 2019).

[2] Despite the fact that a multi-label measure could also be optimized (FAN; LIN, 2007) in this work only binary evaluation measures were considered given their simplicity and direct association with the macro-averaged evaluation measures.

The abstract model to represent a meta-learning pipeline includes some components, such as: the problem space P; the feature space F; the algorithm space A; the performance space Y; and the meta-learning algorithms (SMITH-MILES, 2008). Following a data-driven process, the performance (Y) of a set of algorithms (A) over several datasets (P) is associated with characteristics of such datasets (F), the meta-features. A learning model, induced from this meta-data, is able to predict a solution for a new dataset.

The meta-features used in a MtL system depends on the problem domain, since it needs to capture the characteristics that may impact the performance of the considered algorithms. Therefore, the set of meta-features play a crucial role in the successful use of MtL (BENSUSAN; KALOUSIS, 2001; BILALLI; ABELLÓ; ALUJA-BANET, 2017). Without loss of generality, a meta-feature f is a function $f : \mathscr{D} \rightarrow \mathbb{R}^k$ that, when applied to a dataset $\mathscr{D}$, returns a set of k values that characterize the dataset. Function f can be detailed as

$$f(\mathscr{D}) = \sigma(m(\mathscr{D})),$$

such that, $m : \mathscr{D} \rightarrow \mathbb{R}^{k'}$ is a characterization measure; and $\sigma : \mathbb{R}^{k'} \rightarrow \mathbb{R}^k$ is a summarization function (RIVOLLI *et al.*, 2019). The main meta-features are organized into five groups:

Simple: meta-features that are easily extracted from data (REIF *et al.*, 2014), with low computational cost (REIF, 2012). They are also called *general* measures (CASTIELLO; CASTELLANO; FANELLI, 2005).

Statistical: meta-features that capture statistical properties of the data (REIF *et al.*, 2014), mainly indicators of localization and distribution, such as the average, standard deviation, correlation and kurtosis. They can only characterize numerical attributes (CASTIELLO; CASTELLANO; FANELLI, 2005).

Information-theoretic: meta-features based on information theory (CASTIELLO; CASTELLANO; FANELLI, 2005), usually entropy estimates (SEGRERA; LUCAS; GARCÍA, 2008), which capture the amount of information in (subsets of) a dataset (SMITH-MILES, 2008).

Model-based: meta-features extracted from a model induced from the data (REIF *et al.*, 2014). They are often based on properties of decision tree models (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000; PENG *et al.*, 2002b), when they are referred to as *decision-tree-based* meta-features (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000).

Landmarking: meta-features that use the performance of simple and fast learning algorithms to characterize the datasets (SMITH-MILES, 2008). The algorithms must have different biases and should capture relevant information with a low computational cost.

A full description of the meta-features is found in Rivolli *et al.* (2019). According to the arguments used as input, the meta-features can be related to the predictive attributes, to

the target attribute and to the whole dataset (BRAZDIL *et al.*, 2009). Due to the fact that the binary datasets transformed from a multi-label data share the same predictive attributes, the meta-features that consider only them are not considered in the experiments. More details about the MtL components along with the procedures adopted to define the MtL study are specified in Section 5.3.3.

5.3 Experimental procedures

This section presents the procedures used to carry out the empirical evaluation of the MtL systems and the investigated strategies. It describes the MLC datasets, followed by a short overview of MLC measures and evaluation procedures. Next, the MtL methodology is explained. Finally, it presents the environmental setup of the comparison among the strategies.

5.3.1 Datasets

Table 28 lists the 20 MLC datasets selected by the authors to be used in the experiments. They are from distinct domains (column *Domain*) and their characteristics are diverse. The columns *Inst*, *Attr* and *Lbl* are respectively the number of instances, attributes and labels. Label sets (*lSets*) is the amount of distinct label combination, label cardinality (*lCard*) measures the average number of labels per instance, label density (*lDen*) describes the average frequency of labels and dependency (*Dep*) shows the average unconditional labels' dependency (LUACES *et al.*, 2012).

Table 28 – Characteristics of the MLC datasets.

Dataset	Domain	Inst	Attr	Lbl	lSets	lCard	lDen	Dep
20ng	text	19300	1006	20	55	1.03	0.05	0.08
birds	audio	337	260	15	115	1.84	0.12	0.08
cal500	audio	502	68	141	502	25.54	0.18	0.14
corel5k	image	4995	499	218	2940	3.37	0.02	0.16
emotions	audio	593	72	6	27	1.87	0.31	0.28
enron	text	1702	1001	42	722	3.34	0.08	0.12
fapesp	text	251	7286	18	61	1.35	0.08	0.11
flags	other	194	19	7	54	3.39	0.48	0.15
foodtruck	other	407	21	12	116	2.29	0.20	0.14
image	image	2000	294	5	20	1.24	0.25	0.15
langlog	text	1197	916	38	223	1.31	0.03	0.06
medical	text	949	1421	20	55	1.20	0.06	0.19
msd-195	audio	2901	180	38	267	2.47	0.07	0.24
ohsumed	text	13929	1002	23	1147	1.66	0.07	0.04
scene	image	2407	294	6	15	1.07	0.18	0.11
slashdot	text	3776	1079	18	149	1.18	0.07	0.05
stackex-chess	text	1612	585	78	725	2.07	0.03	0.10
tmc2007-500	text	28596	500	22	1172	2.22	0.10	0.11
yeast	biology	2417	103	14	198	4.24	0.30	0.25
yelp8	image	10784	668	8	117	2.26	0.28	0.11

These datasets are frequently used as benchmarks for MLC experiments. They come from the Cometa repository (CHARTE *et al.*, 2018), an exhaustive collection of MLC datasets, integrated with the tools used in this work. The exceptions are the datasets `fapesp` and `msd-195` obtained from their respective authors, and `yelp8` from the Kaggle website.[3] The datasets were preprocessed with three operations. First, the labels with less than 10 instances were removed to ensure a minimum of instances related to each label in the training and test folds. Next, the instances with no labels were also removed. Finally, the predictive attributes with constant values were removed.

5.3.2 Evaluation Measures

The evaluation of the predictive performance of MLC strategies requires using specific measures that are able to explore their particularities (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). In this study, we are interested in three macro-averaged measures: *macro-F1*, *macro-precision* and *macro-recall*. They are label-based measures that summarize the performance of the respective binary evaluation measure over all labels, such that

$$macro{-}\beta = \frac{1}{q}\sum_{j=1}^{q}\beta_j, \tag{5.5}$$

where $\beta_j = \{F1_j \mid precision_j \mid recall_j\}$. Let TP_j, FP_j, TN_j and FN_j be, respectively, the true positive, false positive, true negative and false negative values of the label λ_j, the F1, precision and recall specification are given by Equations 5.6, 5.7 and 5.8, respectively.

$$F1_j = \frac{2TP_j}{2TP_j + FP_j + FN_j}, \tag{5.6}$$

$$precision_j = \frac{TP_j}{TP_j + FP_j}. \tag{5.7}$$

$$recall_j = \frac{TP_j}{TP_j + FN_j}. \tag{5.8}$$

Considering that the labels associated with an instance are the relevant labels. Semantically, *macro-precision* measures the fraction of relevant labels among those predicted. High precision indicates the ability of a model to correctly predict the labels, although not necessarily all of them. *Macro-recall* measures the fraction of relevant labels that have been predicted out of all the relevant labels. A high recall indicates that a model predicts many labels correctly, but not necessarily only the relevant labels. In turn, *macro-F1* measures the harmonic mean between precision and recall, such that a model with a high *macro-F1* can predict the relevant labels accurately and only them. As the macro-averaged measures give the same weight to all

[3] see <https://www.kaggle.com/c/yelp-restaurant-photo-classification>.

labels (YANG, 1999), they are more sensitive to the performance in the least common labels, which is usually low (JACKSON; MOULINIER, 2002).

In a recent work (RIVOLLI; SOARES; CARVALHO, 2018a), the authors observed the occurrence of problems regarding the inability of an MLC model to properly predict some labels. Given a test set, the Constant Label Prediction (CLP) measures the proportion of labels indiscriminately predicted for all instances, whereas, the Wrong Label Prediction (WLP) measures the proportion of labels incorrectly predicted for all instances. Equations 5.9 and 5.10 present the CLP and WLP, respectively. I is defined in Equation 5.1.

$$CLP = \frac{1}{q}\sum_{j=1}^{q} I(TN_j + FN_j == 0), \tag{5.9}$$

$$WLP = \frac{1}{q}\sum_{j=1}^{q} I(TP_j == 0). \tag{5.10}$$

Unless the individual performance of the labels is reported, the other MLC measures cannot identify such problems. However, the respective measures require using the whole validation or test set.

5.3.3 Meta-Learning procedures

In this work, the MtL is used with 2 distinct purposes. First, to predict for a given label if the BR strategy will be able to predict it correctly for some test instance (directly related to the WLP measure). Second, to recommend for a given label if it can be optimized by an operation (expansion and reduction) or not.

In the former, MtL is used to detect the wrong label prediction. Set A consists of a single algorithm, which is the base algorithm used by the BR strategy. Thus, the options are $A = \{Yes, No\}$ defining if the label will be predicted wrongly for all test instances or not. The performance space comprises the recall measure, given by Equation 5.8, such that a recall equal to zero is mapped to *Yes*, otherwise *No* is used. The idea behind this MtL task is identifying the problematic labels before inducing the model, so that some action can be adopted.

In the latter, MtL is used to recommend if a label should be expanded or reduced by another label, thus $A_{LE} = \{Yes, No\}$ and $A_{LR} = \{Yes, No\}$. After predicting yes for LE and/or LR, a validation procedure should be performed to find the best candidate label to be combined with the original label. The performance evaluation measure used to select the best option is the F1 measure (Equation 5.6), given that both operations optimize this measure. The high computational costs involving the validation procedure to find the best match between pairs of labels justify using MtL in this scenario. When a label is predicted with *No*, the default BR transformation is used without any operation for that label.

In both MtL tasks, the set of instances P consists of the binary transformed datasets using the BR strategy (Equation 5.1). Each meta-instance is related to an individual label from a multi-label dataset, such that $|P| = 749$.

Considering that some labels come from the same multi-label dataset, they share the same predictive attributes. Therefore, only meta-features related to the target and the whole dataset (predictive attributes and target) are used to characterize the meta-instances. Table 29 lists the selected meta-features used in both MtL tasks, more details about them, as well as their formulation are available in Rivolli *et al.* (2019). The summarization functions used to sum up the multi-valued measures are *kursotis*, *max*, *min*, *median*, *min*, *sd* and *skewness*.

Table 29 – List of meta-features selected to characterize the meta-instances. The column Sum. defines if the measures must be summarized.

Group	Name	Sum.	Description
Simple	freqClass	Yes	Frequencies of the class values.
Statistical	canCor	Yes	Canonical correlations between the predictive attributes and the class.
Information-theoretic	classEnt	No	Class entropy.
	eqNumAttr	No	Equivalent number of attributes.
	jointEnt	Yes	Joint Entropy of attributes and classes.
	mutInf	Yes	Mutual information of attributes and classes.
	nsRatio	No	Noisiness of attributes.
Model-based	leaves	No	Number of leaves.
	leavesBranch	Yes	Size of branches.
	leavesCorrob	Yes	Leaf corroboration.
	leavesHomo	Yes	Homogeneity.
	leavesPerClass	Yes	Leaves per class.
	nodes	No	Number of nodes.
	nodesPerAttr	No	Ratio of the number of nodes per attributes.
	nodesPerInst	No	Ratio of the number of nodes per instances.
	nodesPerLevel	Yes	Number of nodes per level.
	nodesRepeated	Yes	Repeated nodes.
	treeDepth	Yes	Tree depth.
	treeImbalance	Yes	Tree imbalance.
	treeShape	Yes	Tree shape.
	varImportance	Yes	Variable importance.
Landmarking	bestNode	Yes	Best Decision Node's performance.
	eliteNN	Yes	Elite Nearest Neighbor's performance.
	naiveBayes	Yes	Naive Bayes' performance.
	oneNN	Yes	One Nearest Neighbor's performance.
	worstNode	Yes	Worst Decision Node's performance.
Others	classConc	Yes	Class concentration coefficient.
	gravity	No	Center of gravity.

Concerning landmarking, three other performance criteria were considered, apart from accuracy: F1, precision and recall, in which each landmarking resulted in 28 meta-features.[4] Thus, in the first MtL task, the set of meta-instances is characterized by $|F| = 252$ meta-features.

[4] 7 summarization function $\times$ 4 performance measure.

For the second task, the previous set of meta-features captures only information about the label base, in which no information related to the labels that can be used in the operations are extracted. In order to represent them, for each label, the other labels are used as predictive attributes generating an alternative dataset. The characterization measures from Table 29 are also extracted for this new dataset, which will result in $|F| = 399$ meta-features.[5]

Finally, the C5.0 decision tree induction (C5) and the Random Forest (RF) (BREIMAN, 2001) algorithms are used to induce the meta-learner from the meta-base. Moreover, random and majority predictors are adopted as baselines. Given that the meta-instances that come from the same MLC dataset share some similarities, the procedure adopted for both MtL tasks is the leave-one-out for MLC dataset. For both tasks, the performance of the meta-learner is evaluated with the simple accuracy measure, given by

$$Acc = \frac{TP + TN}{n}.$$

5.3.4 Pipeline, tools and setup

The whole pipeline of the experiments carried out in this study is related to the following steps:

1. **Characterizing datasets:** Each binary dataset is characterized using the meta-features defined in Table 29. For the second MtL task, an additional characterization is performed using the set of labels as predictive attributes for each label.

2. **Evaluating MLC Performance:** Using the iterative algorithm for stratifying MLC data (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011), 5x2-fold cross-validation with paired folds is performed for distinct strategies using C5 and RF base algorithms.

   BR: For the first MtL task, the recall of each label is assessed to define if the base algorithm will predict the respective label for all instances wrongly.

   LE and LR: Both operations require using a validation procedure to define if the labels can be optimized given an evaluation measure. Using only the training data, 5x2-fold cross-validation is applied, and for each label, the most feasible combination is used with the test set. If the F1 measure can be optimized, the respective operation is considered positive in the second MtL task.

   Other strategies: The DBR, ECC, RAkEL[6] and REMEDIAL are also evaluated using the same paired partitions. They are used in the comparison with the MtL results.

3. **Inducing the meta-models:** For each base algorithm and MtL task, a meta-model is induced using the RF algorithm. The leave-one-out by the MLC dataset is the procedure

[5] In this case, only accuracy was used to characterize the landmarkings.

[6] The RAkEL's hyperparameters are fixed in $m = 2q$ and $k = 3$ according to that suggested in the original paper (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

adopted, whereby the training data are comprised of the labels from 19 MLC datasets and the test with the labels from the other MLC dataset.

4. **Using the meta-models:** Finally, using the recommendation of the meta-models, the label prediction problem is investigated and the LE and LR recommended by the meta-learner are compared against the other strategies and baselines. Moreover, the importance of the meta-features used by the meta-models is investigated.

All the experiments are carried out in the R environment. The packages `mldr` (CHARTE; CHARTE, 2015a) and `utiml` (RIVOLLI; CARVALHO, 2018) provided the code of the multi-label resources used in the experiments. The implementation of the base algorithms come from the packages `C50` and `randomForest` for C5 and RF, respectively. The MFE tool (ALCOBACA *et al.*, 2019) contains the implementation of the selected characterization measures. Default hyperparameter values are used according to their respective packages.

To assess the statistical relevance of the results, the Bayesian hierarchical correlated t-test (BENAVOLI *et al.*, 2017) is used to compare the results of two different strategies over multiple datasets. The test results in probabilities concerning which one is better (left and right), for a particular evaluation measure. It also defines a region of equivalence (rope) that indicates the probability that the difference in performance of the classifiers is insignificant. Benavoli *et al.* (2017) suggests the interval $[-0.01, 0.01]$, which consists of a difference of 1% for a measure whose range is $[0, 1]$. This interval is used for the three macro-averaged evaluation measures.

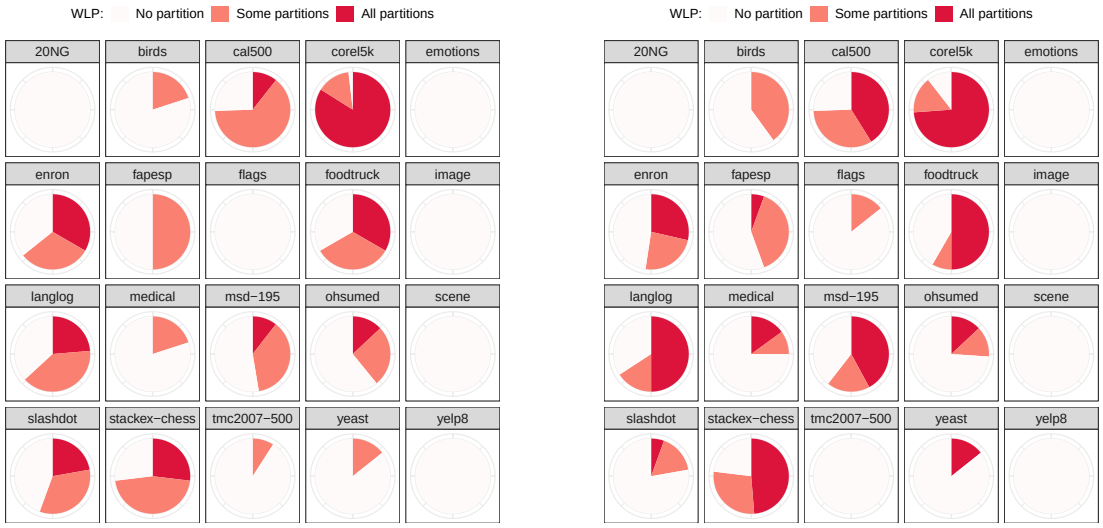
5.4 Results

This section presents the results obtained from the experimental evaluation previously described. First, the MtL results are reported concerning the task to recognize the wrong label prediction in the BR strategy. Next, the use of the MtL is analyzed to predict the label operations LE and LR. Then, the impact of the MtL recommendations in the original tasks is presented. Finally, some strategies are empirically compared to each other considering the MtL results.

5.4.1 Wrong Label Prediction

The goal of the first MtL task is to identify the cases in which a label is not correctly predicted for any instance. It can happen occasionally (only in some partitions of k-fold cross-validation) or always (for all folds). Figure 19 presents the proportion of labels wrongly predicted for all instances by BR using C5 and RF base algorithms. Only in 5 datasets (`20NG`, `emotions`, `image`, `scene` and `yelp8`) the WLP did not occur in both base algorithms.

Therefore, a meta-learner is used to identify possible labels with problems. The meta-features used are defined in Table 29. The meta-target is a binary value identifying if or not the respective label is wrongly predicted for all instances in at least one partition. Thus, a meta-base



(a) BR_{C5} (b) BR_{RF}

Figure 19 – Proportion of labels wrongly predicted for all instances in at least one partition and in all partitions.

is created for each base algorithm and a meta-model is induced from this data. Table 30 presents predictive accuracy according to the experimental procedure. In addition to the meta-learner performance (MtL), the majority (Maj) and random (Rand) predictions are used as baselines. The bold markup highlights the best result for each meta-base.

Table 30 – Accuracy of the label problem prediction task of a Majority (Maj), Random (Rand) and the RF meta-learning (MtL) predictors.

Dataset	BR_{C5}			BR_{RF}		
	Maj	Rand	MtL	Maj	Rand	MtL
20NG	0.000	0.470	**1.000**	0.000	0.490	**1.000**
birds	0.200	0.486	**0.860**	0.400	0.526	**0.666**
cal500	**0.744**	0.509	0.726	0.744	0.470	**0.775**
corel5k	**0.981**	0.480	0.918	0.894	0.512	**0.943**
emotions	0.000	0.583	**1.000**	0.000	0.400	**1.000**
enron	0.642	0.490	**0.888**	0.523	0.561	**0.883**
fapesp	0.500	0.388	**0.644**	0.444	**0.561**	0.516
flags	0.000	0.528	**0.957**	0.142	0.514	**1.000**
foodtruck	0.666	0.541	**0.991**	0.583	0.591	**0.916**
image	0.000	0.420	**1.000**	0.000	0.540	**1.000**
langlog	0.631	0.505	**0.710**	0.657	0.502	**0.760**
medical	0.200	0.415	**0.785**	0.250	0.515	**0.765**
msd-195	0.473	0.497	**0.815**	0.605	0.465	**0.892**
ohsumed	0.391	0.526	**0.839**	0.260	0.530	**0.908**
scene	0.000	0.550	**1.000**	0.000	0.483	**1.000**
slashdot	0.555	0.488	**0.833**	0.222	0.533	**0.916**
stackex-chess	0.730	0.487	**0.808**	0.769	0.515	**0.894**
tmc2007-500	0.090	0.445	**0.927**	0.000	0.536	**1.000**
yeast	0.142	0.485	**1.000**	0.142	0.421	**1.000**
yelp8	0.000	0.537	**1.000**	0.000	0.475	**1.000**
Average	0.347	0.491	**0.885**	0.332	0.507	**0.892**

In comparison to the baselines, the meta-learner achieved a good overall performance. For both meta-bases, almost 90% of labels were correctly predicted whereas in the best case, the baselines can predict 50% of them. Analyzing the performance of the meta-learner considering the MLC datasets, only in 3 of them, the baselines were not outperformed. Otherwise, for many of them, the meta-learner achieved 100% of accuracy.

In order to explain the predictions, a decision tree model is used to learn from the C5 meta-base, the induced tree representation is presented in Figure 20. For more generalization, the tree was pruned so that each leaf represents at least 30 instances from the training data.

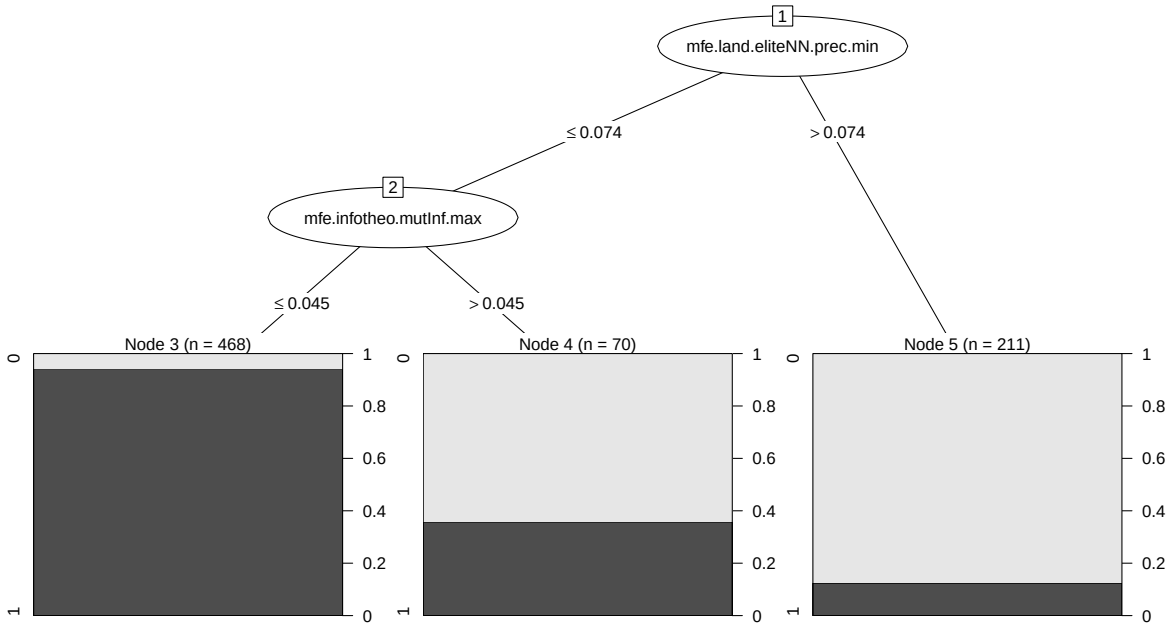


Figure 20 – Decision tree model representation used to explain the meta-learner predictions from the C5 meta-base.

Only three rules and two meta-features classified $\approx 90\%$ of the training data correctly. The rules are considerably simple and obvious, e.g. if the minimum precision of the *eliteNN* landmarker is greater than 0.074 (a low value), then the label will be correctly predicted. Otherwise, if the *eliteNN*'s minimum precision is low and the maximum mutual information between the attributes and the label is also low, then this label will not be correctly predicted. Using these rules to predict the labels from the RF meta-base, 85% of them were correctly predicted.

The most important meta-features according to the meta-models used are landmarkers related to the minimum, median and maximum performance of *eliteNN* and *oneNN*. The exceptions are the mutual information (*mutInf*) and the canonical correlations (*canCor*) that are from information-theoretical and statistical groups, respectively. They are also among the most important meta-features used.

5.4.2 Label Operation

Label operation is performed by combining pairs of labels. For each label, all other labels are assessed using a validation procedure aiming to identify matches that are able to optimize an evaluation measure. In order to decrease the number of labels evaluated in this procedure, a MtL system is used to predict the labels that can be operated. Given that some labels cannot be enhanced through the label operations, by identifying such labels the computational cost of the validation procedure is reduced.

Tables 31 and 32 present the accuracy of the MtL system to predict the labels that can be optimized using the label expansion and label reduction, respectively. The bold markup highlights the best result for each dataset.

Table 31 – Accuracy of the Majority (Maj), Random (Rand) and RF meta-learning (MtL) predictors to the LE prediction task.

Dataset	LE_{C5}			LE_{RF}		
	Maj	Rand	MtL	Maj	Rand	MtL
20NG	0.000	0.430	**0.940**	0.000	0.580	**0.995**
birds	0.333	0.473	**0.533**	0.467	0.426	**0.526**
cal500	0.064	0.495	**0.936**	0.021	0.514	**0.963**
corel5k	0.362	**0.494**	0.389	0.422	0.500	**0.648**
emotions	**0.833**	0.466	0.766	0.333	0.583	**0.816**
enron	**0.690**	0.490	0.535	**0.714**	0.533	0.447
fapesp	0.500	**0.527**	0.500	0.222	0.466	**0.572**
flags	**0.857**	0.514	**0.857**	0.571	**0.599**	0.542
foodtruck	**0.917**	0.508	**0.917**	0.917	0.466	**1.000**
image	0.200	0.440	**0.760**	0.200	0.560	**0.800**
langlog	0.368	**0.476**	0.468	0.237	**0.505**	0.342
medical	0.150	0.475	**0.850**	0.300	**0.525**	0.510
msd-195	**0.737**	0.492	0.710	**0.579**	0.486	**0.579**
ohsumed	0.522	0.469	**0.656**	**0.652**	0.543	0.456
scene	0.167	0.516	**0.667**	0.000	0.533	**0.983**
slashdot	0.389	0.466	**0.722**	0.333	0.466	**0.566**
stackex-chess	0.577	0.517	**0.714**	**0.641**	0.460	0.412
tmc2007-500	0.227	0.522	**0.763**	0.045	0.472	**0.881**
yeast	**0.929**	0.471	**0.929**	**0.857**	0.514	0.728
yelp8	0.250	**0.375**	**0.375**	0.250	0.525	**0.750**
Average	0.453	0.481	**0.699**	0.388	0.513	**0.676**

The overall performance of the MtL outperformed the baselines for all scenarios. Between LE and LR, a subtle better result is observed for the former task, and between C5 and RF, the C5 is better. Thus, the lowest overall accuracy is observed for LR_{RF}. Looking at each dataset separately, the meta-learner was outperformed by the baselines in some cases. Considering that the number of cases is lower than half of the datasets evaluated, regardless of the scenarios, MtL still seems to be advantageous. Furthermore, the accuracy obtained by the MtL is lower than 50% only in a few cases, .

To understand this result better, Table 33 presents the percentage of labels correctly and

Table 32 – Accuracy of the Majority (Maj), Random (Rand) and RF meta-learning (MtL) predictors to the LR prediction task.

Dataset	LR_{C5}			LR_{RF}		
	Maj	Rand	MtL	Maj	Rand	MtL
20NG	**0.800**	0.515	0.650	**0.950**	0.440	0.310
birds	0.400	0.460	**0.586**	0.533	**0.546**	0.519
cal500	0.078	0.492	**0.921**	0.085	0.541	**0.836**
corel5k	0.128	0.489	**0.816**	0.064	0.494	**0.890**
emotions	0.167	0.550	**0.899**	0.500	**0.566**	0.333
enron	0.381	**0.511**	0.428	0.476	0.500	**0.671**
fapesp	0.444	0.539	**0.594**	**0.611**	0.466	0.555
flags	0.000	0.471	**1.000**	0.429	0.528	**0.728**
foodtruck	0.083	0.475	**0.917**	0.167	0.458	**0.917**
image	**0.800**	0.540	0.600	0.600	0.540	**0.620**
langlog	**0.842**	0.478	0.463	0.763	0.513	**0.797**
medical	**0.600**	0.410	0.445	0.450	0.465	**0.515**
msd-195	0.368	0.555	**0.737**	**0.816**	0.502	0.711
ohsumed	0.261	0.539	**0.673**	0.391	0.517	**0.609**
scene	**0.833**	0.550	0.650	**1.000**	0.533	0.783
slashdot	0.611	0.505	**0.650**	**0.667**	0.522	0.438
stackex-chess	0.577	0.505	**0.666**	**0.756**	0.516	0.755
tmc2007-500	0.364	0.418	**0.777**	0.364	0.440	**0.482**
yeast	0.143	0.478	**0.786**	0.286	0.550	**0.850**
yelp8	**0.500**	0.437	**0.500**	0.250	0.537	**0.675**
Average	0.419	0.496	**0.688**	0.507	0.509	**0.649**

wrongly predicted, according to the confusion matrix. Moreover, precision and recall values are presented. In practice, the false negative (FN) column indicates the proportion of labels that are ignored in the validation procedure but should be considered. The false positive (FP) column indicates the proportion of labels that is wrongly considered in the validation. On the other hand, the true negative (TN) and the true positive (TP) comprise the averaged accuracy presented in the previous results and are correctly considered and ignored in the validation procedure, respectively. Precision summarizes the proportion of the labels correctly used in the validation procedure over the total of labels considered. Recall summarizes the amount of labels considered in the validation procedure in relation to all labels that should be considered.

Good precision and recall indicate that the MtL system can reduce the computational costs of the optimization procedure and obtain good results using operations. According to the results, MtL is an alternative to reduce the number of labels considered in the optimization process. The recall results show that in 3 scenarios, around 70% of the labels that should be optimized were predicted correctly. In turn, the precision obtained by the meta-learner, regardless of the scenario, can be seen as a confidence degree in the recommendations, whereby the proportion of labels unnecessarily optimized is between 20% and 40%, varying for each scenario.

Concerning the meta-features, Figure 21 presents the 30 most relevant meta-features according to the RF variable importance. The prefix *specific* indicates the meta-features that characterize the set of labels, instead of the original dataset.

Table 33 – Confusion matrix values, precision and recall results of the meta-learner system and the baselines.

Task	Meta-learner	FN	FP	TN	TP	Precision	Recall
LE_{C5}	Majority	0.445	0.176	0.012	0.367	0.676	0.452
	Random	0.226	0.260	0.231	0.283	0.521	0.556
	MtL	0.270	0.095	0.187	0.449	0.826	0.625
LE_{RF}	Majority	0.254	0.439	0.270	0.037	0.078	0.128
	Random	0.279	0.262	0.244	0.215	0.451	0.435
	MtL	0.155	0.105	0.368	0.371	0.779	0.706
LR_{C5}	Majority	0.446	0.184	0.004	0.366	0.665	0.451
	Random	0.218	0.303	0.232	0.247	0.449	0.532
	MtL	0.170	0.164	0.280	0.386	0.701	0.695
LR_{RF}	Majority	0.272	0.375	0.334	0.019	0.047	0.064
	Random	0.310	0.176	0.296	0.218	0.553	0.413
	MtL	0.111	0.132	0.495	0.262	0.664	0.703

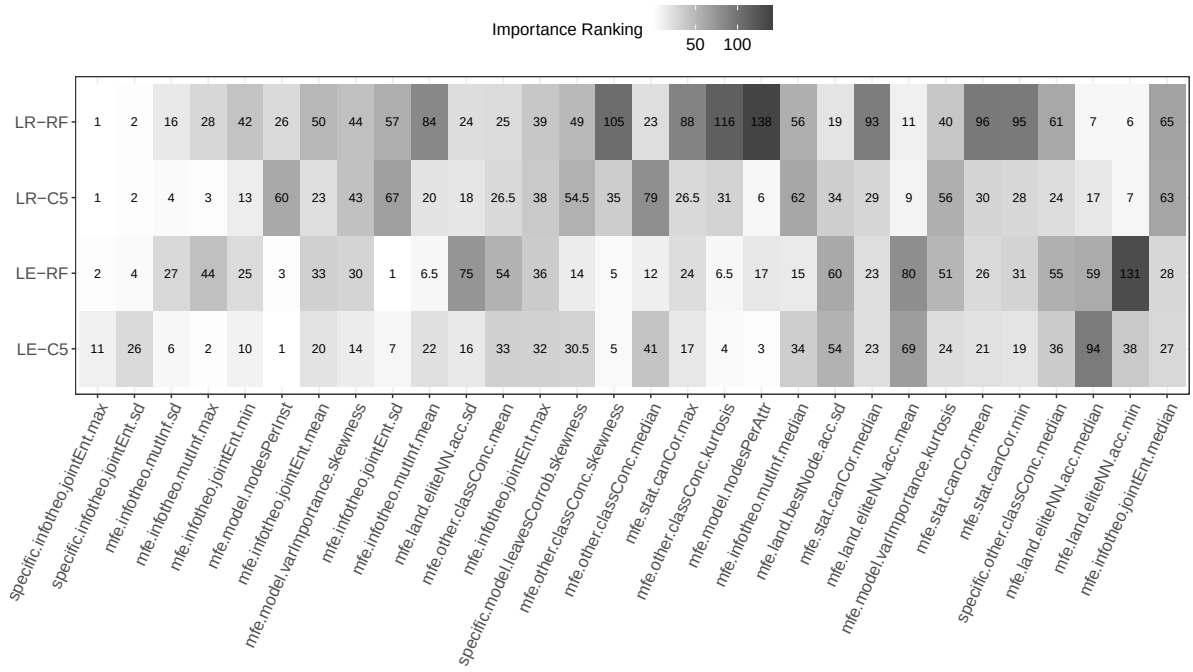


Figure 21 – Most relevant meta-features according to the RF variable importance.

Most of the meta-features selected are related to capturing the shared information among the labels and describing how informative the predictive attributes are concerning the target. The two most relevant meta-features are the joint entropy computed using the label set as dataset and summarized with the max and the standard deviation. From this subset of meta-features, 6 of them are the specific ones, showing that they contributed to the learning process. The information-theoretical group is represented by the most number of meta-features, mainly the joint entropy and the mutual information, summarized in different ways. The mean, median and standard deviation were the most relevant summarization functions present in the list.

Between the different tasks, the set of relevant features presented a noticeable variation among them. The most divergent task is the LR_{RF}, which can be observed by the high ranking of some selected meta-features. In this task, the meta-learner obtained the lowest accuracy and precision compared to the other tasks, which are facts that may be related.

5.4.3 Base-level analysis

In order to investigate the effectiveness of the MtL recommendations concerning the label operation, the results are analyzed in the MLC context. Beginning with the analysis of the cost reduction obtained from the MtL system, Table 34 presents the number of candidate labels to be optimized in each scenario. In this case, BR does not optimize any label whereas the default operations LE/LR try to optimize 100% of them. The LE1 and LR1 are the results of the first MtL task, which identifies the labels that cannot be correctly predicted for any instance. The operations are performed only for the labels identified with problems. The LE2 and LR2 are relative to the second MtL task, in which each operation is directly recommended.

Table 34 – Number of optimized labels in each scenario.

Dataset	BR C5/RF	LE/LR C5/RF	LE1/LR1 C5	LE2 C5	LR2 C5	LE1/LR1 RF	LE2 RF	LR2 RF
20NG	0	20	0	1	10	0	0	9
birds	0	15	1	4	15	1	0	13
cal500	0	141	69	141	140	75	137	106
corel5k	0	218	197	201	55	190	121	30
emotions	0	6	0	3	6	0	1	3
enron	0	42	23	21	11	21	21	15
fapesp	0	18	2	0	12	12	6	15
flags	0	7	1	7	7	1	3	4
foodtruck	0	12	8	12	12	8	11	11
image	0	5	0	2	3	0	0	0
langlog	0	38	35	34	22	34	37	7
medical	0	20	8	1	5	9	5	3
msd-195	0	38	25	38	30	23	38	16
ohsumed	0	23	12	4	12	5	2	11
scene	0	6	0	1	1	0	0	1
slashdot	0	18	7	6	4	5	6	8
stackex-chess	0	78	70	49	20	59	22	15
tmc2007-500	0	22	0	1	10	0	2	6
yeast	0	14	2	14	13	2	8	11
yelp8	0	8	0	2	8	0	0	5
Proportion	0%	100%	61%	72%	53%	59%	56%	39%

The number of recommended labels to be optimized varied according to the operation and base algorithm. For instance, 72% of the $LE2_{C5}$ is recommended whereas for the $LR2_{RF}$ only 39% of labels are recommended. It consists of a reduction of the number of assessed labels during the validation procedure, mainly for the RF base algorithm.

Contextualizing the previous results with the MLC performance, Figure 22 presents the average ranking of each strategy according to different base algorithms and evaluation measures. For instance, even $LR2_{RF}$, using only 39% of the labels, achieved the second-best ranking for *macro-F1* and *macro-precision* measures. Although the results vary according to the base algorithm and evaluation measure, it can be observed that for *macro-F1* and *macro-recall*, the MtL strategies achieved a better ranking than BR. For *macro-precision*, the reduction of the number of labels was useful, considering that the validation procedure can indicate wrong pairs of labels.

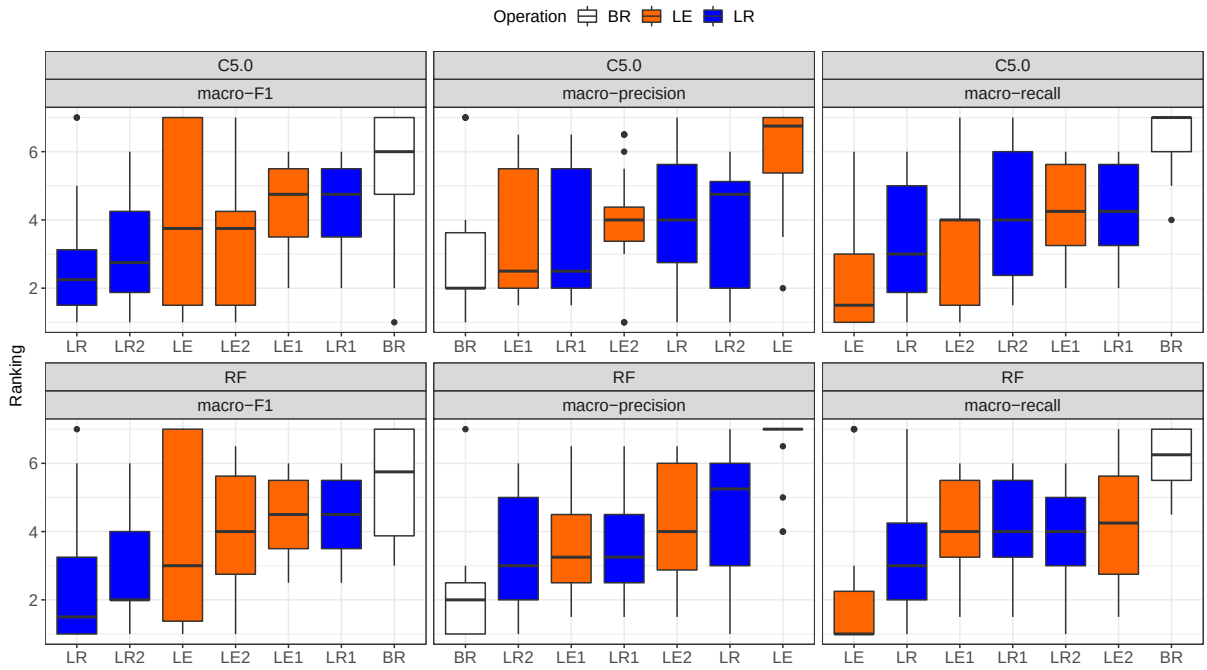


Figure 22 – Strategies' ranking according to different base algorithms and evaluation measures.

When compared statistically (Table 35), the MtL strategies are better than BR for *macro-F1* and mainly for *macro-recall*, given the high probabilities obtained. For *macro-precision*, BR statistically outperforms all operations in most of the cases. In comparison to the default operations LE and LR, the MtL strategies were statistically similar for the optimized *macro-F1* measure, in most of the cases, better for *macro-precision* and worst for *macro-recall*. The MtL recommendations provided a better trade-off between precision and recall measures than the original operations.

Finally, the effect of the operations concerning the label prediction problems CLP and WLP was analysed. Figures 23 and 24 present, respectively, the CLP and WLP results in a heatmap. The strategies (x axis) are sorted, from left to right, according to their average result. Only the datasets (y axis) with the respective problem are reported in the plot.

The CLP occurred in a small number of datasets. Mainly, for the C5 base algorithm, the operations increased the CLP, since they privilege the WLP. Among the variations, LE1 and LR1

Table 35 – Statistical result between the comparison of the meta-learner's recommendation.

Measure	Pair	C5			RF		
		left	rope	right	left	rope	right
macro-F1	BR x LE1	0.01	0.31	**0.67**	0.04	0.34	**0.61**
	BR x LE2	0.01	0.10	**0.88**	0.06	0.12	**0.81**
	BR x LR1	0.02	0.32	**0.65**	0.04	0.42	**0.53**
	BR x LR2	0.00	0.05	**0.93**	0.01	0.04	**0.93**
	LE x LE1	0.43	0.48	0.08	0.41	0.24	0.33
	LE x LE2	0.07	**0.82**	0.10	0.12	0.38	0.49
	LR x LR1	**0.68**	0.29	0.02	**0.90**	0.08	0.01
	LR x LR2	0.02	**0.97**	0.00	0.24	**0.75**	0.00
macro-precision	BR x LE1	0.16	**0.76**	0.06	**0.86**	0.10	0.02
	BR x LE2	**0.65**	0.30	0.03	**0.95**	0.02	0.02
	BR x LR1	0.14	**0.76**	0.08	**0.84**	0.13	0.02
	BR x LR2	**0.73**	0.21	0.05	**0.89**	0.05	0.04
	LE x LE1	0.00	0.00	**0.99**	0.00	0.00	**0.99**
	LE x LE2	0.01	0.05	**0.93**	0.00	0.00	**0.99**
	LR x LR1	0.04	0.17	**0.77**	0.16	0.04	**0.78**
	LR x LR2	0.00	**0.87**	0.11	0.01	0.25	**0.72**
macro-recall	BR x LE1	0.01	0.00	**0.97**	0.02	0.01	**0.95**
	BR x LE2	0.01	0.00	**0.98**	0.05	0.00	**0.93**
	BR x LR1	0.02	0.00	**0.96**	0.02	0.01	**0.96**
	BR x LR2	0.00	0.00	**0.98**	0.02	0.00	**0.97**
	LE x LE1	**0.98**	0.00	0.00	**0.98**	0.00	0.01
	LE x LE2	**0.90**	0.07	0.01	**0.98**	0.00	0.00
	LR x LR1	**0.82**	0.07	0.10	**0.80**	0.04	0.15
	LR x LR2	0.40	**0.58**	0.00	**0.88**	0.11	0.00

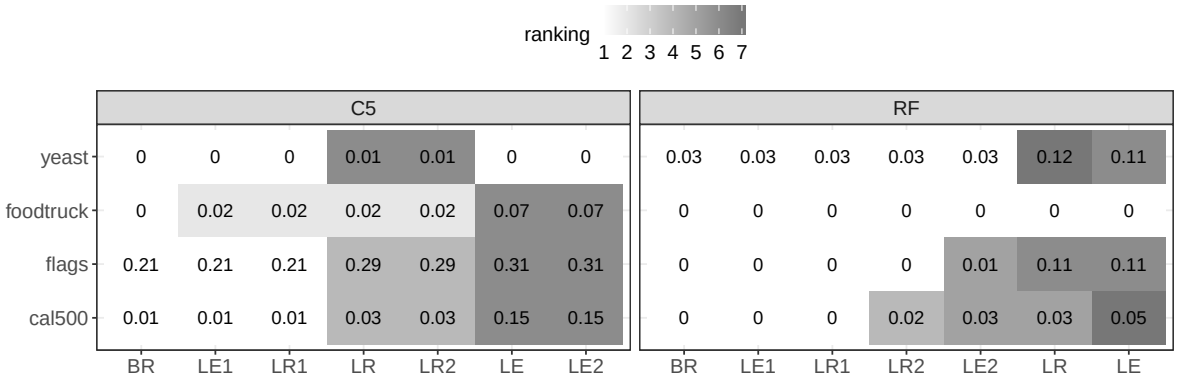


Figure 23 – CLP results considering different strategies and base algorithms.

showed the best results between the operations.

For WLP, the averaged ranking between the base algorithms is more similar. The LE obtained the best ranking, followed by LE1 and LR1, respectively. In this case, the number of labels used in the validation procedure can be correlated with the overall ranking position, except for LE2 using C5.0. Compared to BR, the WLP was reduced for almost all cases regardless of the operation.

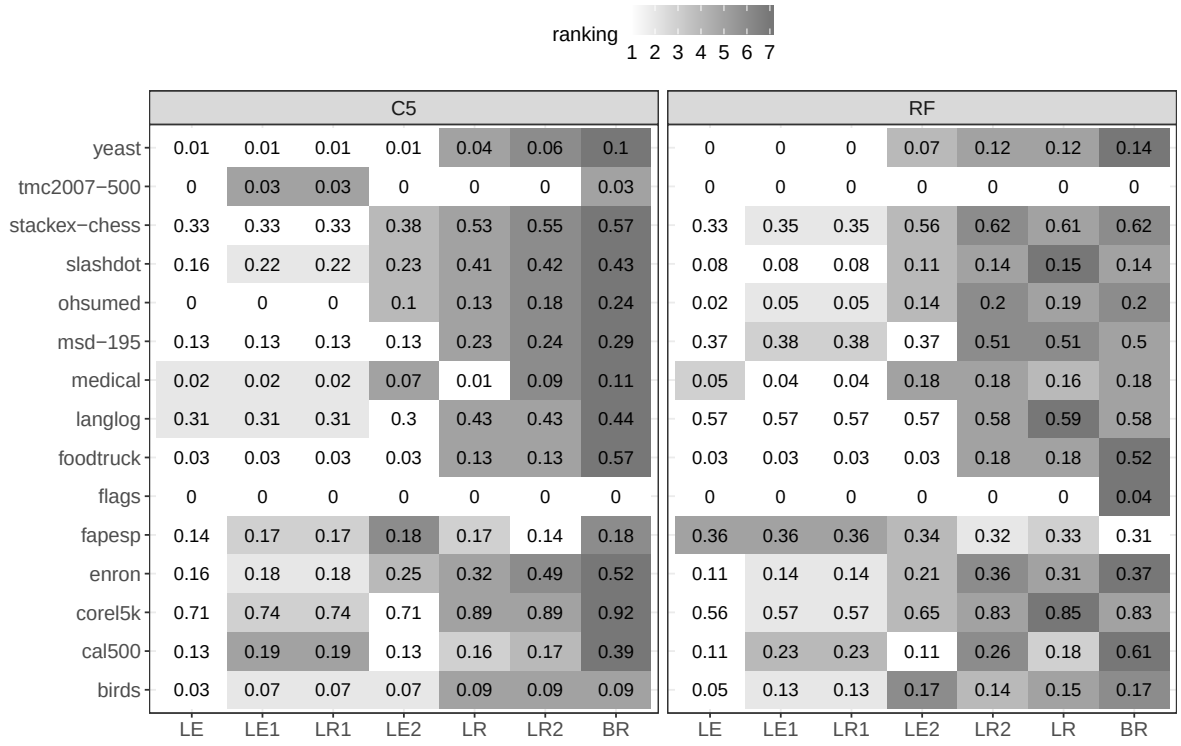


Figure 24 – WLP results considering different strategies and base algorithms.

In summary, the performance of the meta-learner in the meta-level showed to be satisfactory considering the performance of the label operations in the base-level. In addition to reducing the number of optimized labels, following the recommendation of the meta-learners also reduced the label problems and enhanced the optimized measure compared to BR. Although the performance of the operations assessing all labels is superior to the MtL operations, for most of the cases, the results presented showed to be competitive. Thus, the reduction of the number of assessed labels in the validation procedure, achieved by using MtL, is well justified.

5.4.4 Comparative among other strategies

In this section, the MtL operations are compared against other MLC strategies. For the sake of clarity, we considered only the LE2 and LR2 operations to be compared since they considerably reduced the number of optimized labels. The complete results are presented in Appendix E, Tables 56 to 60. Additionally, Tables 36, 37 and 38 present the Bayesian statistical results for the *macro-F1*, *macro-precision* and *macro-recall*, respectively. The strategies in the row improve the strategies in the columns with a probability greater than or equal to 95% for the respective base algorithm.

For *macro-F1*, ECC statistically improved all other strategies; the exceptions are the LE2 and LR2 using RF. The LE2 and LR2 obtained similar results by improving the same strategies, DBR for C5, RAkEL for RF and REMEDIAL for both base algorithms. In turn, RAkEL and

REMEDIAL did not present good results for *macro-F1* as they were statistically outperformed for all other strategies.

Table 36 – Bayesian statistical results for the *macro-F1* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%.

	BR	DBR	ECC	LE2	LR2	RAkEL	REMEDIAL
BR	-					RF	C5,RF
DBR		-				RF	C5,RF
ECC	C5,RF	C5,RF	-	C5	C5	C5,RF	C5,RF
LE2		C5		-		RF	C5,RF
LR2		C5			-	RF	C5,RF
RAkEL						-	C5
REMEDIAL							-

Table 37 – Bayesian statistical results for the *macro-precision* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%.

	BR	DBR	ECC	LE2	LR2	RAkEL	REMEDIAL
BR	-		RF	RF			C5
DBR		-	RF	RF	RF		C5
ECC	C5	C5	-	C5	C5		C5
LE2				-			C5
LR2					-		C5
RAkEL	C5	C5		C5	C5	-	C5
REMEDIAL				RF			-

Table 38 – Bayesian statistical results for the *macro-recall* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%.

	BR	DBR	ECC	LE2	LR2	RAkEL	REMEDIAL
BR	-					RF	C5,RF
DBR		-				RF	C5,RF
ECC	C5,RF	C5,RF	-			C5,RF	C5,RF
LE2	C5	C5,RF		-		C5,RF	C5,RF
LR2	C5,RF	C5,RF			-	C5,RF	C5,RF
RAkEL						-	C5
REMEDIAL							-

For *macro-precision*, ECC and RAkEL using C5.0 outperformed the other strategies. As previously observed, both LE2 and LR2 did not achieve good results for this measure, such that they were outperformed for the other strategies. On the other hand, LE2 and LR2 presented good results for *macro-recall*, statistically outperforming all other strategies. It confirms that these strategies tend to privilege recall in relation to precision.

Despite the fact that ECC obtained the best results and consequently improved the most number of strategies concerning the evaluation measures considered, LE2 obtained the greatest overall reduction of WLP, as illustrated in Figure 25.

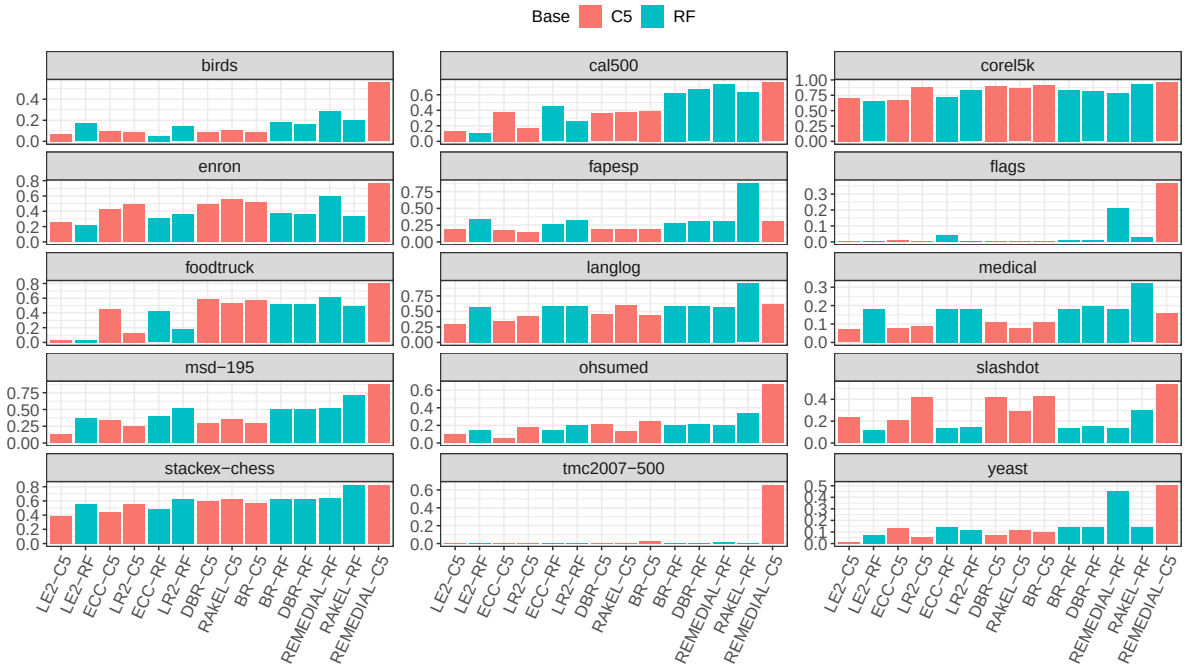


Figure 25 – Comparative WLP results for distinct datasets.

Regardless of the base algorithm, the label operations showed the best results for WLP. For some datasets, such as `foodtruck` and `cal500` the differences with the other strategies can be clearly seen. On the other hand, RAkEL and REMEDIAL showed to be less effective in reducing the WLP. Concerning the base algorithms, in general, strategies using C5 showed better results than the ones using RF.

In summary, ECC and the label operations LE2 and LR2 showed the best results for *macro-F1*, *macro-recall* and WLP. ECC uses an internal threshold calibration which can explain the good performance observed. In turn, the label operations combine the labels to optimize an evaluation measure and consequently, they reduce the number of labels that are completely mispredicted. Moreover, label operations could be combined with the threshold calibration, which would be potentially good for reducing the investigated label prediction problem.

It is worth emphasizing that using MtL to guide operations can reduce, in many cases by half, the number of labels evaluated in the validation procedure. The reduction did not degrade the performance of the operations, such that the label operation showed to be competitive when compared to other strategies.

5.5 Conclusion

This work investigated the use of MtL to reduce the complexity of the label operations and to tackle the WLP, diminishing the number of labels that are completely wrongly predicted. Thus, two different MtL tasks were empirically evaluated. The first aimed to identify the problematic

labels, which were easily detected using a few landmarking meta-features. The second focused on identifying the labels to be operated, which was a harder task when compared to the former.

Regardless of the task, the results obtained at a base level were able to consistently reduce the number of labels operated without impairing the operations' performance significantly. When compared to other strategies, the recommendations of the MtL for LE and LR achieved good results for the *macro-F1* and *macro-recall* evaluation measures. Moreover, they were still able to reduce the WLP problems for many datasets.

To reduce the complexity of the label operations even more, future works should focus on identifying the pairs of labels to be operated. Thus, the complexity of the operations would become very close to the BR strategy, with the advantage that they would optimize an evaluation measure. Replacing the validation procedure for an MtL approach could reduce the number of wrong choices obtained by it. This is a potential alternative to enhance the operations and mitigate the label prediction problems even more.

CHAPTER

6

CONCLUSION

Motivated by a growing number of applications involving multi-labeled data, many strategies have been proposed to explore the particularities of the labels and their relationship (GALINDO; VENTURA, 2014). These strategies explore different aspects, such as label correlation (CHERMAN; METZ; MONARD, 2012), dimensionality reduction (TSOUMAKAS; KATAKIS; VLAHAVAS, 2008) and class imbalance (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

By following the goal of investigating the MLC strategies focusing on the predictive performance of individual labels, some research questions were raised and consequently investigated in this thesis. In this path, MtL was investigated as a procedure to automatize and optimize some steps of the proposed strategies. Three hypotheses guided the development of the work:

1. *The base algorithm has a stronger influence than the transformation strategy on the predictive performance of the MLC models.*

2. *The right combination of labels during the transformation process leads to an improvement in the MLC performance and this can mitigate the label prediction problem.*

3. *MtL can reduce the complexity of label operations and improve their predictive performance.*

Concerning hypothesis 1, the experimental results from Chapter 2 show that, for the binary transformation strategies, the base algorithm has a stronger impact than the strategies in most of the datasets analysed. When several strategies and base algorithms had their predictions compared and clustered by similarity, the groups contained different strategies with the same base algorithm. Furthermore, the rankings of the best combinations for different evaluation measures were more segmented by the base algorithm than the strategies (See Appendix A). Hence, it can be concluded that the selection of the base algorithm for these strategies has more

impact on the results than the choice of the strategy. Following this conclusion, more than one base algorithm was used in the empirical studies reported in the other chapters.

The investigation of hypothesis 2 is reported in Chapter 4, in which the label expansion and reduction operations are proposed. Three evaluation measures were optimized through the label operations. In the experiments using these operations, the number of labels that were never correctly predicted was reduced. Suitable pairs of labels were found by assessing all possible combinations through a validation procedure. Although this procedure still needs further work, the hypothesis was validated for both operations.

Finally, Chapter 5 showed that MtL is a useful mechanism to reduce the complexity of the operations. However it was not possible to increase the predictive performance of the operations. The experimental results from this chapter partially answers hypothesis 3, given that the validation procedure was optimized only in terms of the number of labels assessed.

The rest of this chapter is organized as follows. Section 6.1 summarizes the main contributions from this thesis. Section 6.2 enumerates the papers written and submitted during this research. Section 6.3 discusses the main limitations of this research. Finally, Section 6.4 indicates possible directions for future work according to what was developed and the experience carried out during this PhD research.

6.1 Main contributions

The main contributions from this PhD thesis are related to MLC and MtL. In summary, they include:

1. Demonstration of the impact of the base algorithm to the binary transformation strategies. In many cases, by selecting a robust base algorithm, the simplest transformation strategy can be used in order to attend a given performance criterion. The main implication of this finding is the methodology used in the MLC empirical studies, since multiples and varied base algorithms must be considered when transformation strategies are involved. This contribution is detailed in Chapter 2.

2. Identification and formalization of the label prediction problem in MLC tasks. The incidence of never correctly predicted labels is completely ignored in empirical studies involving multi-labeled data. It is a new error measure that can be used to evaluate MLC models, since two strategies can have similar predictive performance (e.g. 70% of accuracy), but one of them can, for instance, correctly predict more labels. This contribution is stated in Chapter 2. Solutions for this problem are reported in Chapter 4 and 5, along with the papers reporting these contributions Rivolli, Parker and Carvalho (2017) and Rivolli, Soares and Carvalho (2018a).

3. Development of the label operation transformation for MLC. Two operations, expansion and reduction, were formalized and empirically assessed, showing that both of them can mitigate the problem of labels that are never correctly predicted. Furthermore, these operations can be used as procedures to optimize different evaluation measures, including ranking-based measures, such as AUC. This contribution is detailed in Chapter 4 and 5 and in the paper Rivolli, Soares and Carvalho (2018b).

4. Investigation of an MtL approach to identify the labels that cannot be correctly predicted, as well as those that can be enhanced using label operations. As results from this study, the cost of the validation procedure adopted to find the right pairs of labels to be operated can be reduced and by combining distinct operations the predictive performance can be increased. This contribution is detailed in Chapter 5.

5. Analysis of the current meta-features state-of-art and proposing a new taxonomy to organize meta-features according their approach. This study resulted in a survey of characterization measures for MtL with an emphasis on the reproducibility of MtL empirical studies. This comprises Chapter 3 of this thesis.

6. Publication of a new MLC benchmark, the `foodtruck` dataset. Available at Cometa repository (CHARTE *et al.*, 2018), the dataset can be used in the recommendation of food truck cuisines considering some personal information and preferences. The dataset is described and investigated in the papers Rivolli, Parker and Carvalho (2017) and Rivolli, Soares and Carvalho (2018a).

Moreover, two tools were developed and published under open source license at the Comprehensive R Archive Network (CRAN) repository. They fill gaps of libraries in the R data-scientist community concerning the MLC classification and dataset characterization for MtL. The developed tools are:

1. The `utiml` implements most of the state-of-art MLC strategies and pre-processing techniques for multi-label learning (RIVOLLI; CARVALHO, 2018). Available at <https://CRAN.R-project.org/package=utiml>, it provides a set of multi-label procedures, such as sampling methods, classification strategies, threshold functions, pre-processing techniques and evaluation metrics. All the experiments reported in this thesis were performed using the `utiml` tool.

2. The `MFE` is a tool for extracting meta-features from datasets (ALCOBACA *et al.*, 2019). Available at <https://CRAN.R-project.org/package=mfe>, it offers a flexible and standalone implementation of meta-features and summarization functions described in Chapter 3. All the datasets' characterization performed in this thesis were made using the `MFE` tool.

6.2 Publications

The development of this research resulted in some conference and journal papers that are directly and indirectly related to the main contributions of this thesis. They are presented in chronological order:

6.2.1 Conference papers

- RIVOLLI, A.; CARVALHO, A. C. P. L. F. **O uso seletivo de classificadores binários na solução de problemas multirrótulos**. In: XII Encontro Nacional de Inteligência Artificial e Computacional, 2015, Natal. p. 270-277.
- RIVOLLI, A.; PARKER, L. C.; CARVALHO, A. C. P. L. F. **Food Truck Recommendation Using Multi-label Classification**. In: 18th EPIA Conference on Artificial Intelligence, 2017, Porto. 2017. v. 10423. p. 585-596.
- RIVOLLI, A.; SOARES, C.; CARVALHO, A. C. P. L. F. **Label Expansion for Multi-Label Classification**. In: Brazilian Conference on Intelligent Systems (BRACIS), 2018, São Paulo, SP. p. 414-419.
- SILVA, P.; RIVOLLI, A.; ROCHA, P.; CORREIA, F.; SOARES, C. **Machine Learning for Drugs Prescription**. In: Intelligent Data Engineering and Automated Learning - IDEAL, 2018, Madrid, Spain. v. 11314. p. 548-555.

6.2.2 Journal papers

- RIVOLLI, A.; CARVALHO, A. C. P. L. F. **The utiml Package: Multi-label Classification in R**. R Journal, v. 10, p. 24-37, 2018.
- RIVOLLI, A.; SOARES, C.; CARVALHO, A. C. P. L. F. **Enhancing multilabel classification for food truck recommendation**. EXPERT SYSTEMS, v. 35, p. 1-19, 2018.
- RIVOLLI, A.; READ, J.; SOARES, C.; PFAHRINGER, B.; CARVALHO, A. C. P. L. F. **An empirical analysis of binary transformation strategies for multi-label learning**. Machine Learning, 2018, *under minor revision*.
- GARCIA, L. P. F.; RIVOLLI, A.; ALCOBACA, E.; LORENA, A. C.; CARVALHO, A. C. P. L. F. **Boosting Meta-Learning with Simulated Data Complexity Measures**. Intelligent Data Analysis, 2019, *accepted*.
- ALCOBACA, E.; SIQUEIRA, F.; RIVOLLI, A.; GARCIA, L. P. F.; OLIVA, J. T.; CARVALHO, A. C. P. L. F. **MFE: Towards reproducible meta-feature extraction**. Journal of Machine Learning Research, 2019, *accepted*.

- RIVOLLI, A.; GARCIA, L. P. F.; SOARES, C.; VANSCHOREN, J.; CARVALHO, A. C. P. L. F. **Characterizing classification datasets: a study of meta-features for meta-learning**. Information Sciences, 2019, *submitted*.

- RIVOLLI, A.; SOARES, C.; PFAHRINGER, B.; CARVALHO, A. C. P. L. F. **Label Operations for Multi-label Optimization**. Data Mining and Knowledge Discovery, 2020, *submitted*.

- RIVOLLI, A.; SOARES, C.; PFAHRINGER, B.; CARVALHO, A. C. P. L. F. **Recommending label operations for multi-label classification**. Neurocomputing, 2020, *submitted*.

6.3 Limitations

Empirical studies in MLC usually employ a small number of datasets, as they may demand a large amount of processing time, affected by the relationship and the number of labels. In this thesis, 20 MLC datasets were considered, which is subtly higher than the number of datasets used in most of the studies published in this area. However, considering the number of strategies and base algorithms used, this number is still small because it makes it difficult to find patterns and obtain reliable generalizations.

In addition to this methodological aspect, other research limitations were assumed with the development of this work. Only binary transformation strategies were considered in the study of the importance of the base algorithm to the transformation strategies. It leads to a more specific assumption since it is not possible to assume the same results for other transformations, such as pairwise and multi-class transformation. However, when considering a group of more diversified strategies, some of those that were examined would be removed due to the size of the experiment. Therefore, it was defined that, initially, this study would cover a whole family of strategies. Future studies will explore other families and be integrated with this study.

Concerning the label operation, the validation procedure adopted to find the pairs of labels to be operated was the main limitation of the proposed approach. Potentially, both expansion and reduction operations can produce better results providing that the right labels are combined. Moreover, the computational costs involved to find suitable combinations for large datasets is very high. Heuristics and rules to avoid the exhaustive search in the validation procedure might lead to better choices at a lower cost. However, one of the impediments here is the small number of MLC datasets available.

The solution adopted in order to remedy this problem was MtL. However, the MtL task could be designed in different ways. For instance, the recommendation system could be projected to answer whether one label could improve another given an operation. This would modify the whole validation procedure. However, a simpler recommendation system was adopted, in which given a label, the task is to predict whether it can be optimized or not. Given this decision,

it is still necessary to run the validation procedure for the labels positively identified by the recommender system.

Besides, all the meta-bases generated and used for inducing the meta-learners have an unexplored particularity. Considering that each meta-instance represents a label, many of them have more similarities with each other than with others. This results in dependence between the meta-instances so that the methodology adopted in the experiments was the leave-one-out for the MLC dataset. Nevertheless, the MtL can explore such conditions to take some advantages and improve the learning task.

6.4 Future Work

The limitations described in the previous section inspired some future studies identified for this thesis. At the end of each chapter, specific future work directions are drawn and discussed. Here, the main issues are highlighted summarizing the next steps of this work.

The impact of the base algorithm over pairwise and multi-class transformation strategies can be further investigated. This includes the analysis of their similarity, analogous to what was made with the binary transformation strategies.

Concerning the label operation, other operations, beyond label expansion and reduction, can be investigated exploring, for instance, association rules among labels. Furthermore, the mechanism to find the best pairs of labels to operate should be tuned. A more strict validation procedure can potentially reduce the number of mistakes, whereas it can lead to a gain in predictive performance. On the other hand, the use of MtL to recommend a suitable pair of labels can solve this trade-off in a better way, therefore it is also suggested as future work.

In terms of MtL, the investigation of meta-instances with a high relationship of dependency between themselves was shown to be a promising path to follow. How to explore this characteristic in favor of a performance gain in the predictive models is the question that arises from this scenario.

Finally, the problem of labels that were never correctly predicted is not completely solved. For many datasets, the label operations were able to reduce the problem observed, but there is still room for improvement in this matter. Considering that many labels never predicted are associated with a small number of instances, the use of techniques to deal with the presence of imbalanced data seems to be a reasonable alternative to be further investigated.

REFERENCES

ABDELMESSIH, S. D.; SHAFAIT, F.; REIF, M.; GOLDSTEIN, M. Landmarking for meta-learning using RapidMiner. In: **RapidMiner Community Meeting and Conference (RCOMM)**. [S.l.: s.n.], 2010. p. 1–6. Citations on pages 90 and 92.

ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. **Learning From Data**. California, USA: AMLBook, 2012. Citation on page 25.

AGGARWAL, C. C. **Data Mining - The Textbook**. [S.l.]: Springer, 2015. ISBN 978-3-319-14141-1. Citations on pages 84 and 85.

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large databases. In: BOCCA, J. B.; JARKE, M.; ZANIOLO, C. (Ed.). **VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile**. [S.l.]: Morgan Kaufmann, 1994. p. 487–499. Citation on page 118.

AL-OTAIBI, R.; FLACH, P.; KULL, M. Multi-label Classification: A Comparative Study on Threshold Selection Methods. In: **First International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD 2014**. Nancy, France: [s.n.], 2014. Citation on page 101.

ALALI, A.; KUBAT, M. Prudent: A pruned and confident stacking approach for multi-label classification. **IEEE Trans. Knowl. Data Eng.**, v. 27, n. 9, p. 2480–2493, 2015. Citations on pages 36, 39, and 45.

ALCOBACA, E.; SIQUEIRA, F.; RIVOLLI, A.; GARCIA, L. P. F.; OLIVA, J. T.; CARVALHO, A. C. P. L. F. de. Mfe: Towards reproducible meta-feature extraction. **The Journal of Machine Learning Research**, in press, 2019. Citations on pages 134 and 149.

ALI, S.; SMITH, K. A. On learning algorithm selection for classification. **Appl. Soft Comput.**, v. 6, n. 2, p. 119–138, 2006. Citations on pages 66, 69, 84, 92, 94, 127, 187, and 193.

ALI, S.; SMITH-MILES, K. A. A meta-learning approach to automatic kernel selection for support vector machines. **Neurocomputing**, v. 70, n. 1-3, p. 173–186, 2006. Citations on pages 92, 94, 187, and 188.

ANTENREITER, M.; ORTNER, R.; AUER, P. Combining Classifiers for Improved Multilabel Image Classification. In: **Proceedings of the 1st workshop on learning from multilabel data**. Bled, Slovenia: [s.n.], 2009. p. 16–27. Citation on page 25.

BALTE, A.; PISE, N.; KULKARNI, P. Meta-learning with landmarking : A Survey. **International Journal of Computer Applications**, v. 105, n. 8, p. 47 – 51, 2014. Citation on page 92.

BARELLA, V. H.; GARCIA, L. P. F.; SOUTO, M. C. P. de; LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Data complexity measures for imbalanced classification tasks. In: **2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018**. [S.l.]: IEEE, 2018. p. 1–8. Citation on page 80.

BENAVOLI, A.; CORANI, G.; DEMSAR, J.; ZAFFALON, M. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. **J. Mach. Learn. Res.**, v. 18, p. 77:1–77:36, 2017. Available: <http://jmlr.org/papers/v18/16-305.html>. Citations on pages 33, 47, 106, and 134.

BENSUSAN, H.; GIRAUD-CARRIER, C. G. Discovering task neighbourhoods through landmark learning performances. In: ZIGHED, D. A.; KOMOROWSKI, H. J.; ZYTKOW, J. M. (Ed.). **Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD 2000, Lyon, France, September 13-16, 2000, Proceedings**. [S.l.]: Springer, 2000. (Lecture Notes in Computer Science, v. 1910), p. 325–330. Citations on pages 64 and 92.

BENSUSAN, H.; GIRAUD-CARRIER, C. G.; KENNEDY, C. J. A higher-order approach to meta-learning. In: CUSSENS, J.; FRISCH, A. M. (Ed.). **Inductive Logic Programming, 10th International Conference, ILP 2000, Work-in-progress reports, London, UK, July 2000, Proceedings**. [S.l.]: CEUR-WS.org, 2000. (CEUR Workshop Proceedings, v. 35). Citations on pages 67, 68, 74, 128, 190, and 191.

BENSUSAN, H.; KALOUSIS, A. Estimating the predictive accuracy of a classifier. In: RAEDT, L. D.; FLACH, P. A. (Ed.). **Machine Learning: EMCL 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings**. [S.l.]: Springer, 2001. (Lecture Notes in Computer Science, v. 2167), p. 25–36. Citations on pages 28, 64, and 128.

BERNARDINI, F. C.; BENITO, E.; MEZA, M. Cardinality and density measures and their influence to multi-label learning methods. **Journal of the Brazilian Society on Computational Intelligence**, v. 12, n. 1, p. 53–71, 2014. Citation on page 41.

BILALLI, B.; ABELLÓ, A.; ALUJA-BANET, T. On the predictive power of meta-features in openml. **Applied Mathematics and Computer Science**, v. 27, n. 4, p. 697–712, 2017. Citations on pages 28, 64, 84, 89, 90, and 128.

BILALLI, B.; ABELLÓ, A.; ALUJA-BANET, T.; WREMBEL, R. Intelligent assistance for data pre-processing. **Computer Standards & Interfaces**, v. 57, p. 101–109, 2018. Citations on pages 82 and 127.

BOUTELL, M. R.; LUO, J.; SHEN, X.; BROWN, C. M. Learning multi-label scene classification. **Pattern Recognition**, v. 37, n. 9, p. 1757–1771, 2004. Citations on pages 25, 27, 36, 41, 98, and 124.

BRAZDIL, P.; GAMA, J.; HENERY, B. Characterizing the applicability of classification algorithms using meta-level learning. In: BERGADANO, F.; RAEDT, L. D. (Ed.). **Machine Learning: ECML-94, European Conference on Machine Learning, Catania, Italy, April 6-8, 1994, Proceedings**. [S.l.]: Springer, 1994. (Lecture Notes in Computer Science, v. 784), p. 83–102. Citations on pages 70 and 87.

BRAZDIL, P.; GIRAUD-CARRIER, C. G.; SOARES, C.; VILALTA, R. **Metalearning - Applications to Data Mining**. [S.l.]: Springer, 2009. (Cognitive Technologies). ISBN 978-3-540-73262-4. Citations on pages 27, 64, 67, 68, 89, 122, 127, and 129.

BRAZDIL, P.; SOARES, C.; COSTA, J. P. da. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. **Machine Learning**, v. 50, n. 3, p. 251–277, 2003. Citation on page 87.

BRAZDIL, P.; VILALTA, R.; GIRAUD-CARRIER, C. G.; SOARES, C. Metalearning. In: SAMMUT, C.; WEBB, G. I. (Ed.). **Encyclopedia of Machine Learning and Data Mining**. [S.l.]: Springer, 2017. p. 818–823. Citations on pages 104 and 118.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001. Citations on pages 46 and 133.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and Regression Trees**. [S.l.]: Wadsworth, 1984. ISBN 0-534-98053-8. Citation on page 94.

BRIGGS, F.; HUANG, Y.; RAICH, R.; EFTAXIAS, K.; LEI, Z.; CUKIERSKI, W.; HADLEY, S. F.; HADLEY, A.; BETTS, M.; FERN, X. Z.; IRVINE, J.; NEAL, L.; THOMAS, A.; FODOR, G.; TSOUMAKAS, G.; NG, H. W.; NGUYEN, T. N. T.; HUTTUNEN, H.; RUUSUVUORI, P.; MANNINEN, T.; DIMENT, A.; VIRTANEN, T.; MARZAT, J.; DEFRETIN, J.; CALLENDER, D.; HURLBURT, C.; LARREY, K.; MILAKOV, M. The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In: **IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2013, Southampton, United Kingdom, September 22-25, 2013**. [S.l.]: IEEE, 2013. p. 1–8. Citations on pages 25, 32, 41, and 122.

BURTON, S. H.; MORRIS, R. G.; GIRAUD-CARRIER, C. G.; WEST, J. H.; THACKERAY, R. Mining useful association rules from questionnaire data. **Intell. Data Anal.**, v. 18, n. 3, p. 479–494, 2014. Citation on page 81.

CARVALHO, A. C. P. de Leon Ferreira de; FREITAS, A. A. A tutorial on multi-label classification techniques. In: ABRAHAM, A.; HASSANIEN, A. E.; SNÁSEL, V. (Ed.). **Foundations of Computational Intelligence - Volume 5: Function Approximation and Classification**. [S.l.]: Springer, 2009, (Studies in Computational Intelligence, v. 205). p. 177–195. Citations on pages 25, 26, 32, 34, 99, 100, 122, 123, and 124.

CASTIELLO, C.; CASTELLANO, G.; FANELLI, A. M. Meta-data: Characterization of input features for meta-learning. In: TORRA, V.; NARUKAWA, Y.; MIYAMOTO, S. (Ed.). **Modeling Decisions for Artificial Intelligence, Second International Conference, MDAI 2005, Tsukuba, Japan, July 25-27, 2005, Proceedings**. [S.l.]: Springer, 2005. (Lecture Notes in Computer Science, v. 3558), p. 457–468. Citations on pages 27, 66, 67, 69, 70, 72, 73, 82, 84, 86, 87, 94, 128, 186, 187, and 189.

CHANG, C.; LIN, C. LIBSVM: A library for support vector machines. **ACM TIST**, v. 2, n. 3, p. 27:1–27:27, 2011. Citation on page 46.

CHARTE, F.; CHARTE, D. Working with Multilabel Datasets in R: The mldr Package. **The R Journal**, v. 7, n. 2, p. 149–162, 2015. Citations on pages 107 and 134.

CHARTE, F.; CHARTE, F. D. Working with Multilabel Datasets in R: The mldr Package. **The R Journal**, v. 7, n. 2, p. 149–162, 2015. Citation on page 45.

CHARTE, F.; RIVAS, A. J. R.; JESÚS, M. J. del; HERRERA, F. Concurrence among imbalanced labels and its influence on multilabel resampling algorithms. In: POLYCARPOU, M. M.; CARVALHO, A. C. P. de Leon Ferreira de; PAN, J.; WOZNIAK, M.; QUINTIÁN, H.; CORCHADO, E. (Ed.). **Hybrid Artificial Intelligence Systems - 9th International Conference, HAIS 2014, Salamanca, Spain, June 11-13, 2014. Proceedings**. [S.l.]: Springer, 2014. (Lecture Notes in Computer Science, v. 8480), p. 110–121. Citation on page 120.

CHARTE, F.; RIVERA, A. J.; CHARTE, D.; JESÚS, M. J. del; HERRERA, F. Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets R package and the cometa data repository. **Neurocomputing**, v. 289, p. 68–85, 2018. Citations on pages 41, 104, 130, and 149.

CHARTE, F.; RIVERA, A. J.; JESÚS, M. J. del; HERRERA, F. QUINTA: A question tagging assistant to improve the answering ratio in electronic forums. In: **IEEE EUROCON 2015 - International Conference on Computer as a Tool, Salamanca, Spain, September 8-11, 2015**. [S.l.]: IEEE, 2015. p. 1–6. Citations on pages 25 and 40.

____. Resampling multilabel datasets by decoupling highly imbalanced labels. In: ONIEVA, E.; SANTOS, I.; OSABA, E.; QUINTIÁN, H.; CORCHADO, E. (Ed.). **Hybrid Artificial Intelligent Systems - 10th International Conference, HAIS 2015, Bilbao, Spain, June 22-24, 2015, Proceedings**. [S.l.]: Springer, 2015. (Lecture Notes in Computer Science, v. 9121), p. 489–501. Citations on pages 27 and 125.

____. Remedial-hwr: Tackling multilabel imbalance through label decoupling and data resampling hybridization. **Neurocomputing**, v. 326-327, p. 110–122, 2019. Citation on page 120.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: KRISHNAPURAM, B.; SHAH, M.; SMOLA, A. J.; AGGARWAL, C. C.; SHEN, D.; RASTOGI, R. (Ed.). **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016**. [S.l.]: ACM, 2016. p. 785–794. Citation on page 46.

CHERMAN, E. A.; METZ, J.; MONARD, M. C. Incorporating label dependency into the binary relevance framework for multi-label classification. **Expert Syst. Appl.**, v. 39, n. 2, p. 1647–1655, 2012. Citations on pages 32, 33, 36, 38, 45, 101, and 147.

CHERMAN, E. A.; SPOLAÔR, N.; VALVERDE-REBAZA, J. C.; MONARD, M. C. Lazy multi-label learning algorithms based on mutuality strategies. **Journal of Intelligent and Robotic Systems**, v. 80, n. Supplement-1, p. 261–276, 2015. Citation on page 40.

CLARE, A.; KING, R. D. Knowledge discovery in multi-label phenotype data. In: RAEDT, L. D.; SIEBES, A. (Ed.). **Principles of Data Mining and Knowledge Discovery, 5th European Conference, PKDD 2001, Freiburg, Germany, September 3-5, 2001, Proceedings**. [S.l.]: Springer, 2001. (Lecture Notes in Computer Science, v. 2168), p. 42–53. Citation on page 26.

DAYRELL, C.; JR., A. C.; LIMA, G.; JR., D. M.; COPESTAKE, A. A.; FELTRIM, V. D.; TAGNIN, S. E. O.; ALUÍSIO, S. M. Rhetorical move detection in english abstracts: Multi-label sentence classifiers and their annotated corpora. In: CALZOLARI, N.; CHOUKRI, K.; DECLERCK, T.; DOGAN, M. U.; MAEGAARD, B.; MARIANI, J.; ODIJK, J.; PIPERIDIS, S. (Ed.). **Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012**. [S.l.]: European Language Resources Association (ELRA), 2012. p. 1604–1609. Citation on page 25.

DÍEZ, J.; LUACES, O.; COZ, J. J. del; BAHAMONDE, A. Optimizing different loss functions in multilabel classifications. **Progress in AI**, v. 3, n. 2, p. 107–118, 2015. Citation on page 105.

DIPLARIS, S.; TSOUMAKAS, G.; MITKAS, P. A.; VLAHAVAS, I. P. Protein classification with multiple algorithms. In: BOZANIS, P.; HOUSTIS, E. N. (Ed.). **Advances in Informatics, 10th Panhellenic Conference on Informatics, PCI 2005, Volos, Greece, November 11-13, 2005, Proceedings**. [S.l.]: Springer, 2005. (Lecture Notes in Computer Science, v. 3746), p. 448–456. Citation on page 26.

DUWAIRI, R. M.; KASSAWNEH, A. A framework for predicting proteins 3d structures. In: **The 6th ACS/IEEE International Conference on Computer Systems and Applications, AICCSA 2008, Doha, Qatar, March 31 - April 4, 2008**. [S.l.]: IEEE Computer Society, 2008. p. 37–44. Citation on page 26.

DUYGULU, P.; BARNARD, K.; FREITAS, J. F. G. de; FORSYTH, D. A. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: HEYDEN, A.; SPARR, G.; NIELSEN, M.; JOHANSEN, P. (Ed.). **Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part IV**. [S.l.]: Springer, 2002. (Lecture Notes in Computer Science, v. 2353), p. 97–112. Citations on pages 25, 32, 41, and 122.

ELISSEEFF, A.; WESTON, J. A kernel method for multi-labeled classification. In: **Proceedings of the Neural Information Processing Systems**. Vancouver, Canada: MIT Press, 2001. p. 681–687. Citations on pages 26, 32, 41, and 122.

ENGELS, R.; THEUSINGER, C. Using a data metric for preprocessing advice for data mining applications. In: PRADE, H. (Ed.). **13th European Conference on Artificial Intelligence, Brighton, UK, August 23-28 1998, Proceedings.** [S.l.]: John Wiley and Sons, 1998. p. 430–434. Citations on pages 68, 73, 185, 186, 187, 188, and 189.

FAN, R.; LIN, C. **A study on threshold selection for multi-label classification**. [S.l.], 2007. 1–23 p. Citations on pages 98, 103, 119, and 127.

FAYYAD, U. M.; IRANI, K. B. Multi-interval discretization of continuous-valued attributes for classification learning. In: BAJCSY, R. (Ed.). **Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993**. [S.l.]: Morgan Kaufmann, 1993. p. 1022–1029. Citation on page 85.

FERRARI, D. G.; CASTRO, L. N. de. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. **Inf. Sci.**, v. 301, p. 181–194, 2015. Citation on page 193.

FEURER, M.; SPRINGENBERG, J. T.; HUTTER, F. Using meta-learning to initialize bayesian optimization of hyperparameters. In: VANSCHOREN, J.; BRAZDIL, P.; SOARES, C.; KOTTHOFF, L. (Ed.). **Proceedings of the International Workshop on Meta-learning and Algorithm Selection co-located with 21st European Conference on Artificial Intelligence, MetaSel@ECAI 2014, Prague, Czech Republic, August 19, 2014**. [S.l.]: CEUR-WS.org, 2014. (CEUR Workshop Proceedings, v. 1201), p. 3–10. Citations on pages 82, 185, 186, and 195.

FILCHENKOV, A.; PENDRYAK, A. Datasets meta-feature description for recommending feature selection algorithm. In: **Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)**. [S.l.: s.n.], 2015. p. 11 – 18. Citations on pages 64, 68, 74, 82, 87, 89, 90, 92, and 190.

FÜRNKRANZ, J.; PETRAK, J. An evaluation of landmarking variants. In: **1st ECML/PKDD International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM)**. [S.l.: s.n.], 2001. p. 57 – 68. Citations on pages 64, 68, and 76.

GALINDO, E. L. G.; VENTURA, S. Multi-label learning: a review of the state of the art and ongoing research. **Wiley Interdiscip. Rev. Data Min. Knowl. Discov.**, v. 4, n. 6, p. 411–444, 2014. Citations on pages 25, 32, 98, 122, and 147.

GARCIA, L. P. F.; CARVALHO, A. C. P. L. F. de; LORENA, A. C. Noise detection in the meta-learning level. **Neurocomputing**, v. 176, p. 14–25, 2016. Citations on pages 78, 82, 84, 89, 90, and 92.

GARCIA, L. P. F.; LORENA, A. C.; CARVALHO, A. C. P. L. F. de. A study on class noise detection and elimination. In: LORENA, A. C.; THOMAZ, C. E.; POZO, A. T. R. (Ed.). **2012 Brazilian Symposium on Neural Networks, Curitiba, Paraná, Brazil, October 20-25, 2012**. [S.l.]: IEEE Computer Society, 2012. p. 13–18. Citation on page 28.

GARCIA, L. P. F.; LORENA, A. C.; MATWIN, S.; CARVALHO, A. C. P. de Leon Ferreira de. Ensembles of label noise filters: a ranking approach. **Data Min. Knowl. Discov.**, v. 30, n. 5, p. 1192–1216, 2016. Citation on page 127.

GARCIA, L. P. F.; LORENA, A. C.; SOUTO, M. C. P. de; HO, T. K. Classifier recommendation using data complexity measures. In: **24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018**. [S.l.]: IEEE Computer Society, 2018. p. 874–879. Citations on pages 76 and 78.

GELMAN, A.; HILL, J. **Data analysis using regression and multilevel/hierarchical models**. New York: Cambridge University Press, 2007. Citation on page 46.

GIBAJA, E.; VENTURA, S. A tutorial on multilabel learning. **ACM Comput. Surv.**, v. 47, n. 3, p. 52:1–52:38, 2015. Available: <https://doi.org/10.1145/2716262>. Citations on pages 34, 35, and 42.

GODBOLE, S.; SARAWAGI, S. Discriminative methods for multi-labeled classification. In: DAI, H.; SRIKANT, R.; ZHANG, C. (Ed.). **Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004, Proceedings**. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3056), p. 22–30. Citations on pages 36 and 38.

GONÇALVES, E. C.; PLASTINO, A.; FREITAS, A. A. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: **25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2013, Herndon, VA, USA, November 4-6, 2013**. IEEE Computer Society, 2013. p. 469–476. Available: <https://doi.org/10.1109/ICTAI.2013.76>. Citation on page 41.

GONÇALVES, T.; QUARESMA, P. A preliminary approach to the multilabel classification problem of portuguese juridical documents. In: MOURA-PIRES, F.; ABREU, S. (Ed.). **Progress in Artificial Intelligence, 11th Protuguese Conference on Artificial Intelligence, EPIA 2003, Beja, Portugal, December 4-7, 2003, Proceedings**. [S.l.]: Springer, 2003. (Lecture Notes in Computer Science, v. 2902), p. 435–444. Citation on page 25.

HALL, M. A.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The WEKA data mining software: an update. **SIGKDD Explorations**, v. 11, n. 1, p. 10–18, 2009. Citation on page 92.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques, Second Edition**. [S.l.]: Elsevier, 2006. (The Morgan Kaufmann series in data management systems). ISBN 978-1-55860-901-3. Citations on pages 85, 86, and 88.

HANDL, J.; KNOWLES, J. D.; KELL, D. B. Computational cluster validation in post-genomic data analysis. **Bioinformatics**, v. 21, n. 15, p. 3201–3212, 2005. Citations on pages 77 and 78.

HO, T. K.; BASU, M. Complexity measures of supervised classification problems. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 24, n. 3, p. 289–300, 2002. Citation on page 78.

HOTELLING, H. Analysis of a complex of statistical variables with principal components. **Journal of Educational Psychology**, v. 24, p. 417–441, 1933. Citation on page 90.

HUANG, S.; PENG, W.; LI, J.; LEE, D. Sentiment and topic analysis on social media: a multi-task multi-label classification approach. In: DAVIS, H. C.; HALPIN, H.; PENTLAND, A.; BERNSTEIN, M.; ADAMIC, L. A. (Ed.). **Web Science 2013 (co-located with ECRC), WebSci '13, Paris, France, May 2-4, 2013**. [S.l.]: ACM, 2013. p. 172–181. Citation on page 25.

HUTSON, M. Artificial intelligence faces reproducibility crisis. **Science**, v. 359, n. 6377, p. 725–726, 2018. Citation on page 64.

HUTTER, F.; KOTTHOFF, L.; VANSCHOREN, J. (Ed.). **Automated Machine Learning - Methods, Systems, Challenges**. [S.l.]: Springer, 2019. (The Springer Series on Challenges in Machine Learning). ISBN 978-3-030-05317-8. Citation on page 127.

JACKSON, P.; MOULINIER, I. **Natural Language Processing for Online Applications: Text Retrieval, Extraction & Categorization**. John Benjamins, 2002. ISBN 90-272-4989-X. Available: <http://members.aol.com/JacksonPE/music1/nlp4olap.htm>. Citations on pages 43, 105, and 131.

JAIN, A. K.; DUBES, R. C. **Algorithms for Clustering Data**. [S.l.]: Prentice-Hall, 1988. Citation on page 49.

JIN, R.; BREITBART, Y.; MUOH, C. Data discretization unification. **Knowl. Inf. Syst.**, v. 19, n. 1, p. 1–29, 2009. Citation on page 85.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: NEDELLEC, C.; ROUVEIROL, C. (Ed.). **Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Proceedings**. [S.l.]: Springer, 1998. (Lecture Notes in Computer Science, v. 1398), p. 137–142. Citations on pages 25 and 40.

JOANES, D. N.; GILL, C. A. Comparing measures of sample skewness and kurtosis. **Journal of the Royal Statistical Society**, v. 47, n. 1, p. 183 – 189, 1998. Citation on page 72.

KALOUSIS, A. **Algorithm Selection via Meta-Learning**. Phd Thesis (PhD Thesis) — Faculty of Science of the University of Geneva, 2002. Citation on page 186.

KALOUSIS, A.; HILARIO, M. Feature selection for meta-learning. In: CHEUNG, D. W.; WILLIAMS, G. J.; LI, Q. (Ed.). **Knowledge Discovery and Data Mining - PAKDD 2001, 5th Pacific-Asia Conference, Hong Kong, China, April 16-18, 2001, Proceedings**. [S.l.]: Springer, 2001. (Lecture Notes in Computer Science, v. 2035), p. 222–233. Citation on page 89.

____. Model selection via meta-learning: A comparative study. **International Journal on Artificial Intelligence Tools**, v. 10, n. 4, p. 525–554, 2001. Citations on pages 70, 80, 87, and 194.

KALOUSIS, A.; THEOHARIS, T. NOEMON: design, implementation and performance results of an intelligent assistant for classifier selection. **Intell. Data Anal.**, v. 3, n. 5, p. 319–337, 1999. Citations on pages 66, 82, 84, 87, and 185.

KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. Multilabel Text Classification for Automated Tag Suggestion. In: **Proceedings of the ECML/PKDD-08 Workshop on Discovery Challenge**. Antwerp, Belgium: [s.n.], 2008. p. 1–9. Citation on page 25.

KAWAI, K.; TAKAHASHI, Y. Identification of the Dual Action Antihypertensive Drugs Using TFS-Based Support Vector Machines. **Chem-Bio Informatics Journal**, v. 9, p. 41–51, 2009. Citation on page 26.

KLIMT, B.; YANG, Y. The enron corpus: A new dataset for email classification research. In: BOULICAUT, J.; ESPOSITO, F.; GIANNOTTI, F.; PEDRESCHI, D. (Ed.). **Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004, Proceedings**. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3201), p. 217–226. Citations on pages 32, 40, and 122.

KOPF, C.; IGLEZAKIS, I. Combination of task description strategies and case base properties for meta-learning. In: **2nd ECML/PKDD International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM)**. [S.l.: s.n.], 2002. p. 65 – 76. Citations on pages 81, 84, 90, 92, 188, 194, and 195.

KOPF, C.; TAYLOR, C.; KELLER, J. Meta-Analysis: From data characterisation for meta-learning to meta-regression. In: **PKDD Workshop on Data Mining, Decision Support,Meta-Learning and Inductive Logic Programming**. [S.l.: s.n.], 2000. p. 15 – 26. Citations on pages 67 and 188.

KUBA, P.; BRAZDIL, P.; SOARES, C.; WOZNICA, A. Exploiting sampling and meta-learning for parameter setting for support vector machines. In: **8th IBERAMIA Workshop on Learning and Data Mining**. [S.l.: s.n.], 2002. p. 209 – 216. Citations on pages 69, 82, 87, and 185.

LANG, K. Newsweeder: Learning to filter netnews. In: PRIEDITIS, A.; RUSSELL, S. J. (Ed.). **Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995**. [S.l.]: Morgan Kaufmann, 1995. p. 331–339. Citation on page 40.

LEE, J. W.; GIRAUD-CARRIER, C. G. Predicting algorithm accuracy with a small set of effective meta-features. In: WANI, M. A.; CHEN, X.; CASASENT, D.; KURGAN, L. A.; HU, T.; HAFEEZ, K. (Ed.). **Seventh International Conference on Machine Learning and Applications, ICMLA 2008, San Diego, California, USA, 11-13 December 2008**. [S.l.]: IEEE Computer Society, 2008. p. 808–812. Citation on page 89.

LEITE, R.; BRAZDIL, P. Predicting relative performance of classifiers from samples. In: RAEDT, L. D.; WROBEL, S. (Ed.). **Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005**. [S.l.]: ACM, 2005. (ACM International Conference Proceeding Series, v. 119), p. 497–503. Citations on pages 76 and 127.

LEMKE, C.; BUDKA, M.; GABRYS, B. Metalearning: a survey of trends and technologies. **Artif. Intell. Rev.**, v. 44, n. 1, p. 117–130, 2015. Citations on pages 88 and 89.

LER, D.; TENG, H.; HE, Y.; GIDIJALA, R. Algorithm selection for classification problems via cluster-based meta-features. In: ABE, N.; LIU, H.; PU, C.; HU, X.; AHMED, N.; QIAO, M.; SONG, Y.; KOSSMANN, D.; LIU, B.; LEE, K.; TANG, J.; HE, J.; SALTZ, J. S. (Ed.). **IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018**. [S.l.]: IEEE, 2018. p. 4952–4960. Citations on pages 77 and 193.

LEWIS, D. D.; SCHAPIRE, R. E.; CALLAN, J. P.; PAPKA, R. Training algorithms for linear text classifiers. In: FREI, H.; HARMAN, D.; SCHÄUBLE, P.; WILKINSON, R. (Ed.). **Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum)**. [S.l.]: ACM, 1996. p. 298–306. Citation on page 25.

LEWIS, D. D.; YANG, Y.; ROSE, T. G.; LI, F. RCV1: A new benchmark collection for text categorization research. **J. Mach. Learn. Res.**, v. 5, p. 361–397, 2004. Citation on page 25.

LI, S.; HUANG, L.; WANG, R.; ZHOU, G. Sentence-level emotion classification with label and context dependence. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers**. [S.l.]: The Association for Computer Linguistics, 2015. p. 1045–1053. Citation on page 25.

LI, Y.; ZHANG, M. Enhancing binary relevance for multi-label learning with controlled label correlations exploitation. In: PHAM, D. N.; PARK, S. (Ed.). **PRICAI 2014: Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia, December 1-5, 2014. Proceedings**. [S.l.]: Springer, 2014. (Lecture Notes in Computer Science, v. 8862), p. 91–103. Citation on page 45.

LINDNER, G.; STUDER, R. AST: support for algorithm selection with a CBR approach. In: ZYTKOW, J. M.; RAUCH, J. (Ed.). **Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings**. [S.l.]: Springer, 1999. (Lecture Notes in Computer Science, v. 1704), p. 418–423. Citations on pages 68, 69, 70, 83, 92, 186, 187, and 189.

LIU, S. M.; CHEN, J. An empirical study of empty prediction of multi-label classification. **Expert Syst. Appl.**, v. 42, n. 13, p. 5567–5579, 2015. Citation on page 45.

LO, H.; WANG, J.; WANG, H.; LIN, S. Cost-sensitive stacking for audio tag annotation and retrieval. In: **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic**. [S.l.]: IEEE, 2011. p. 2308–2311. Citation on page 25.

LOH, W.-Y. Fifty years of classification and regression trees. **International Statistical Review**, v. 82, n. 3, p. 329 – 348, 2014. Citation on page 74.

LORENA, A. C.; GARCIA, L. P. F.; LEHMANN, J.; SOUTO, M. C. P. de; HO, T. K. How complex is your classification problem?: A survey on measuring classification complexity. **ACM Comput. Surv.**, v. 52, n. 5, p. 107:1–107:34, 2019. Citations on pages 78 and 194.

LUACES, O.; DÍEZ, J.; BARRANQUERO, J.; COZ, J. J. del; BAHAMONDE, A. Binary relevance efficacy for multilabel classification. **Progress in AI**, v. 1, n. 4, p. 303–313, 2012. Citations on pages 36, 40, 45, 98, 104, and 129.

LUENGO, J.; HERRERA, F. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. **Knowl. Inf. Syst.**, v. 42, n. 1, p. 147–180, 2015. Citation on page 78.

MADJAROV, G.; KOCEV, D.; GJORGJEVIKJ, D.; DZEROSKI, S. An extensive experimental comparison of methods for multi-label learning. **Pattern Recognition**, v. 45, n. 9, p. 3084–3104, 2012. Citations on pages 27, 32, 33, 45, 46, and 59.

MANTOVANI, R. G.; ROSSI, A. L. D.; ALCOBAÇA, E.; VANSCHOREN, J.; CARVALHO, A. C. P. L. F. de. A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. **Inf. Sci.**, v. 501, p. 193–221, 2019. Citation on page 127.

MANTOVANI, R. G.; ROSSI, A. L. D.; VANSCHOREN, J.; BISCHL, B.; CARVALHO, A. C. P. L. F. To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. In: **2015 International Joint Conference on Neural Networks**. [S.l.]: IEEE, 2015. p. 1–8. Citations on pages 46 and 127.

MARKATOPOULOU, F.; MEZARIS, V.; KOMPATSIARIS, I. A comparative study on the use of multi-label classification techniques for concept-based video indexing and annotation. In: GURRIN, C.; HOPFGARTNER, F.; HÜRST, W.; JOHANSEN, H. D.; LEE, H.; O'CONNOR, N. E. (Ed.). **MultiMedia Modeling - 20th Anniversary International Conference, MMM 2014, Dublin, Ireland, January 6-10, 2014, Proceedings, Part I**. [S.l.]: Springer, 2014. (Lecture Notes in Computer Science, v. 8325), p. 1–12. Citation on page 26.

MATHWORKS. **Statistics toolbox: for use with MATLAB: user's guide**. 2001. Citation on page 92.

MENCíA, E. L.; FÜRNKRANZ, J. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In: DAELEMANS, W.; GOETHALS, B.; MORIK, K. (Ed.). **Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II**. [S.l.]: Springer, 2008. (Lecture Notes in Computer Science, v. 5212), p. 50–65. Citation on page 25.

MENC'iA, E. L.; JANSSEN, F. Learning rules for multi-label classification: a stacking and a separate-and-conquer approach. **Machine Learning**, v. 105, n. 1, p. 77–126, 2016. Citation on page 101.

METZ, J.; ABREU, L. F. D. de; CHERMAN, E. A.; MONARD, M. C. On the estimation of predictive evaluation measure baselines for multi-label learning. In: PAVÓN, J.; DUQUE-MÉNDEZ, N. D.; FUENTES-FERNÁNDEZ, R. (Ed.). **Advances in Artificial Intelligence - IBERAMIA 2012 - 13th Ibero-American Conference on AI, Cartagena de Indias, Colombia, November 13-16, 2012. Proceedings**. [S.l.]: Springer, 2012. (Lecture Notes in Computer Science, v. 7637), p. 189–198. Citations on pages 44, 45, 47, and 48.

MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C. **Machine Learning, Neural and Statistical Classification**. [S.l.]: Ellis Horwood, 1994. ISBN 0-13-106360-X. Citations on pages 69, 70, 73, 84, 185, 186, 187, 188, 189, and 190.

MIERSWA, I.; WURST, M.; KLINKENBERG, R.; SCHOLZ, M.; EULER, T. YALE: rapid prototyping for complex data mining tasks. In: ELIASSI-RAD, T.; UNGAR, L. H.; CRAVEN, M.; GUNOPULOS, D. (Ed.). **Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006**. [S.l.]: ACM, 2006. p. 935–940. Citation on page 92.

MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw-Hill, 1997. (McGraw Hill series in computer science). ISBN 978-0-07-042807-2. Citations on pages 25 and 64.

MONTAÑÉS, E.; SENGE, R.; BARRANQUERO, J.; QUEVEDO, J. R.; COZ, J. J. del; HÜLLERMEIER, E. Dependent binary relevance models for multi-label classification. **Pattern Recognition**, v. 47, n. 3, p. 1494–1508, 2014. Citations on pages 27, 32, 36, 38, 45, 59, 98, 101, 120, and 124.

MONTEJO-RÁEZ, A.; LÓPEZ, L. A. U.; STEINBERGER, R. Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In: GONZÁLEZ, J. L. V.; MARTÍNEZ-BARCO, P.; MUÑOZ, R.; SAIZ-NOEDA, M. (Ed.). **Advances in Natural Language Processing, 4th International Conference, EsTAL 2004, Alicante, Spain, October 20-22, 2004, Proceedings**. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3230), p. 1–12. Citation on page 40.

MORAIS, G.; PRATI, R. C. Complex network measures for data set characterization. In: **Brazilian Conference on Intelligent Systems, BRACIS 2013, Fortaleza, CE, Brazil, 19-24 October, 2013**. [S.l.]: IEEE Computer Society, 2013. p. 12–18. Citations on pages 76 and 79.

MOYANO, J. M.; GALINDO, E. L. G.; CIOS, K. J.; VENTURA, S. Review of ensembles of multi-label classifiers: Models, experimental study and prospects. **Information Fusion**, v. 44, p. 33–45, 2018. Citations on pages 33 and 59.

MUÑOZ, M. A.; VILLANOVA, L.; BAATAR, D.; SMITH-MILES, K. Instance spaces for machine learning classification. **Machine Learning**, v. 107, n. 1, p. 109–147, 2018. Citations on pages 68, 81, 89, and 90.

NASCIMENTO, A. C. A.; PRUDÊNCIO, R. B. C.; SOUTO, M. C. P. de; COSTA, I. G. Mining rules for the automatic selection process of clustering methods applied to cancer gene expression data. In: ALIPPI, C.; POLYCARPOU, M. M.; PANAYIOTOU, C. G.; ELLINAS, G. (Ed.). **Artificial Neural Networks - ICANN 2009, 19th International Conference, Limassol, Cyprus, September 14-17, 2009, Proceedings, Part II**. [S.l.]: Springer, 2009. (Lecture Notes in Computer Science, v. 5769), p. 20–29. Citation on page 193.

NGUYEN, P.; WANG, J.; HILARIO, M.; KALOUSIS, A. Learning heterogeneous similarity measures for hybrid-recommendations in meta-mining. In: ZAKI, M. J.; SIEBES, A.; YU, J. X.; GOETHALS, B.; WEBB, G. I.; WU, X. (Ed.). **12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012**. [S.l.]: IEEE Computer Society, 2012. p. 1026–1031. Citation on page 74.

PAPAGIANNOPOULOU, C.; TSOUMAKAS, G.; TSAMARDINOS, I. Discovering and exploiting deterministic label relationships in multi-label learning. In: CAO, L.; ZHANG, C.; JOACHIMS, T.; WEBB, G. I.; MARGINEANTU, D. D.; WILLIAMS, G. (Ed.). **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015**. [S.l.]: ACM, 2015. p. 915–924. Citation on page 101.

PENG, Y.; FLACH, P. A.; BRAZDIL, P.; SOARES, C. Decision tree-based data characterization for meta-learning. In: **2nd ECML/PKDD International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM)**. [S.l.: s.n.], 2002. p. 111 – 122. Citations on pages 82, 87, 190, and 191.

PENG, Y.; FLACH, P. A.; SOARES, C.; BRAZDIL, P. Improved dataset characterisation for meta-learning. In: LANGE, S.; SATOH, K.; SMITH, C. H. (Ed.). **Discovery Science, 5th International Conference, DS 2002, Lübeck, Germany, November 24-26, 2002, Proceedings**. [S.l.]: Springer, 2002. (Lecture Notes in Computer Science, v. 2534), p. 141–152. Citations on pages 64, 68, 69, 89, 90, and 128.

PEREIRA, R. B.; PLASTINO, A.; ZADROZNY, B.; MERSCHMANN, L. H. C. Correlation analysis of performance measures for multi-label classification. **Inf. Process. Manage.**, v. 54, n. 3, p. 359–369, 2018. Citation on page 42.

PESTIAN, J. P.; BREW, C.; MATYKIEWICZ, P.; HOVERMALE, D. J.; JOHNSON, N.; COHEN, K. B.; DUCH, W. A shared task involving multi-label classification of clinical free text. In: COHEN, K. B.; DEMNER-FUSHMAN, D.; FRIEDMAN, C.; HIRSCHMAN, L.; PESTIAN, J. (Ed.). **Biological, translational, and clinical language processing, BioNLP@ACL 2007, Prague, Czech Republic, June 29, 2007**. [S.l.]: Association for Computational Linguistics, 2007. p. 97–104. Citations on pages 25, 32, 40, and 122.

PFAHRINGER, B.; BENSUSAN, H.; GIRAUD-CARRIER, C. G. Meta-learning by landmarking various learning algorithms. In: LANGLEY, P. (Ed.). **Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000**. [S.l.]: Morgan Kaufmann, 2000. p. 743–750. Citations on pages 64, 67, and 92.

PILLAI, I.; FUMERA, G.; ROLI, F. Threshold optimisation for multi-label classifiers. **Pattern Recognition**, v. 46, n. 7, p. 2055–2065, 2013. Citation on page 100.

PIMENTEL, B. A.; CARVALHO, A. C. P. L. F. de. A new data characterization for selecting clustering algorithms using meta-learning. **Inf. Sci.**, v. 477, p. 203–219, 2019. Citations on pages 68, 76, 78, 90, and 193.

PINTO, F.; SOARES, C.; MENDES-MOREIRA, J. CHADE: metalearning with classifier chains for dynamic combination of classifiers. In: FRASCONI, P.; LANDWEHR, N.; MANCO, G.; VREEKEN, J. (Ed.). **Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I**. [S.l.]: Springer, 2016. (Lecture Notes in Computer Science, v. 9851), p. 410–425. Citation on page 26.

____. Towards automatic generation of metafeatures. In: BAILEY, J.; KHAN, L.; WASHIO, T.; DOBBIE, G.; HUANG, J. Z.; WANG, R. (Ed.). **Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part I**. [S.l.]: Springer, 2016. (Lecture Notes in Computer Science, v. 9651), p. 215–226. Citations on pages 64, 82, 83, 87, 88, 89, and 92.

QI, G.; HUA, X.; RUI, Y.; TANG, J.; MEI, T.; ZHANG, H. Correlative multi-label video annotation. In: LIENHART, R.; PRASAD, A. R.; HANJALIC, A.; CHOI, S.; BAILEY, B. P.; SEBE, N. (Ed.). **Proceedings of the 15th International Conference on Multimedia 2007, Augsburg, Germany, September 24-29, 2007**. [S.l.]: ACM, 2007. p. 17–26. Citation on page 26.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. [S.l.]: Morgan Kaufmann, 1993. ISBN 1-55860-238-0. Citation on page 46.

RAUBER, T. W.; MELLO, L. H. S.; ROCHA, V. F.; LUCHI, D.; VAREJÃO, F. M. Recursive dependent binary relevance model for multi-label classification. In: BAZZAN, A. L. C.; PICHARA, K. (Ed.). **Advances in Artificial Intelligence - IBERAMIA 2014 - 14th Ibero-American Conference on AI, Santiago de Chile, Chile, November 24-27, 2014, Proceedings**. [S.l.]: Springer, 2014. (Lecture Notes in Computer Science, v. 8864), p. 206–217. Citations on pages 36, 38, and 45.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. In: BUNTINE, W. L.; GROBELNIK, M.; MLADENIC, D.; SHAWE-TAYLOR, J. (Ed.). **Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II**. [S.l.]: Springer, 2009. (Lecture Notes in Computer Science, v. 5782), p. 254–269. Citations on pages 37, 38, 40, and 125.

____. Classifier chains for multi-label classification. **Machine Learning**, v. 85, n. 3, p. 333–359, 2011. Citations on pages 27, 32, 36, 37, 38, 39, 40, 45, 59, 101, 120, and 125.

READ, J.; ZLIOBAITE, I.; HOLLMÉN, J. Labeling sensing data for mobility modeling. **Inf. Syst.**, v. 57, p. 207–222, 2016. Citation on page 26.

REIF, M. A comprehensive dataset for evaluating approaches of various meta-learning tasks. In: CARMONA, P. L.; SÁNCHEZ, J. S.; FRED, A. L. N. (Ed.). **ICPRAM 2012 - Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods, Volume 1, Vilamoura, Algarve, Portugal, 6-8 February, 2012**. [S.l.]: SciTePress, 2012. p. 273–276. Citations on pages 67, 69, and 128.

REIF, M.; SHAFAIT, F.; DENGEL, A. Prediction of classifier training time including parameter optimization. In: BACH, J.; EDELKAMP, S. (Ed.). **KI 2011: Advances in Artificial Intelligence, 34th Annual German Conference on AI, Berlin, Germany, October 4-7,2011. Proceedings**. [S.l.]: Springer, 2011. (Lecture Notes in Computer Science, v. 7006), p. 260–271. Citations on pages 64, 68, 81, 90, 194, and 195.

____. Meta^2-Features: Providing meta-learners more information. In: **35th German Conference on Artificial Intelligence (KI)**. [S.l.: s.n.], 2012. p. 74 – 77. Citations on pages 66, 82, and 87.

REIF, M.; SHAFAIT, F.; GOLDSTEIN, M.; BREUEL, T. M.; DENGEL, A. Automatic classifier selection for non-experts. **Pattern Anal. Appl.**, v. 17, n. 1, p. 83–96, 2014. Citations on pages 64, 67, 68, 69, 84, 89, 90, 92, and 128.

RIVOLLI, A.; CARVALHO, A. C. P. L. F. de. The utiml Package: Multi-label Classification in R. **The R Journal**, v. 10, n. 2, p. 24–37, 2018. Citations on pages 45, 107, 134, and 149.

RIVOLLI, A.; GARCIA, L. P. F.; SOARES, C.; VANSCHOREN, J.; CARVALHO, A. C. P. L. F. de. Characterizing classification datasets: a study of meta-features for meta-learning. **CoRR**, abs/1808.10406, 2019. Available: <http://arxiv.org/abs/1808.10406>. Citations on pages 122, 128, and 132.

RIVOLLI, A.; PARKER, L. C.; CARVALHO, A. C. P. L. F. de. Food truck recommendation using multi-label classification. In: OLIVEIRA, E. C.; GAMA, J.; VALE, Z. A.; CARDOSO,

H. L. (Ed.). **Progress in Artificial Intelligence - 18th EPIA Conference on Artificial Intelligence, EPIA 2017, Porto, Portugal, September 5-8, 2017, Proceedings**. [S.l.]: Springer, 2017. (Lecture Notes in Computer Science, v. 10423), p. 585–596. Citations on pages 26, 148, and 149.

RIVOLLI, A.; SOARES, C.; CARVALHO, A. C. P. L. F. de. Enhancing multilabel classification for food truck recommendation. **Expert Systems**, v. 35, n. 4, 2018. Citations on pages 33, 43, 45, 98, 100, 122, 131, 148, and 149.

____. Label expansion for multi-label classification. In: **7th Brazilian Conference on Intelligent Systems, BRACIS 2018, São Paulo, Brazil, October 22-25, 2018**. [S.l.]: IEEE Computer Society, 2018. p. 414–419. Citations on pages 98, 99, and 149.

RODGERS, J. L.; NICEWANDER, W. A. Thirteen ways to look at the correlation coefficient. **The American Statistician**, v. 42, n. 1, p. 59 – 66, 1988. Citation on page 71.

RODOVALHO, R. M.; BERNARDINI, F. C. Using artificial datasets to analyze how cardinality and density influence multi-label learning. In: **2014 Brazilian Conference on Intelligent Systems, BRACIS 2014, Sao Paulo, Brazil, October 18-22, 2014**. [S.l.]: IEEE Computer Society, 2014. p. 19–24. Citation on page 102.

ROKACH, L. A survey of clustering algorithms. In: MAIMON, O.; ROKACH, L. (Ed.). **Data Mining and Knowledge Discovery Handbook, 2nd ed.** [S.l.]: Springer, 2010. p. 269–298. Citation on page 118.

ROUSSEEUW, P. J.; HUBERT, M. Robust statistics for outlier detection. **Wiley Interdiscip. Rev. Data Min. Knowl. Discov.**, v. 1, n. 1, p. 73–79, 2011. Available: <https://doi.org/10.1002/widm.2>. Citations on pages 94 and 188.

ROYSTON, P. Remark AS R94: A remark on algorithm AS 181: The W-test for normality. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, v. 44, n. 4, p. 547 – 551, 1995. Citations on pages 94 and 188.

SÁ, A. G. C. de; FREITAS, A. A.; PAPPA, G. L. Automated selection and configuration of multi-label classification algorithms with grammar-based genetic programming. In: AUGER, A.; FONSECA, C. M.; LOURENÇO, N.; MACHADO, P.; PAQUETE, L.; WHITLEY, D. (Ed.). **Parallel Problem Solving from Nature - PPSN XV - 15th International Conference, Coimbra, Portugal, September 8-12, 2018, Proceedings, Part II**. [S.l.]: Springer, 2018. (Lecture Notes in Computer Science, v. 11102), p. 308–320. Citation on page 33.

SÁ, A. G. C. de; PAPPA, G. L.; FREITAS, A. A. Towards a method for automatically selecting and configuring multi-label classification algorithms. In: BOSMAN, P. A. N. (Ed.). **Genetic and Evolutionary Computation Conference, Berlin, Germany, July 15-19, 2017, Companion Material Proceedings**. [S.l.]: ACM, 2017. p. 1125–1132. Citations on pages 33 and 60.

SALAMA, M. A.; HASSANIEN, A. E.; REVETT, K. Employment of neural network and rough set in meta-learning. **Memetic Computing**, v. 5, n. 3, p. 165–177, 2013. Citations on pages 87, 89, 90, 94, 188, and 195.

SCHAPIRE, R. E.; SINGER, Y. Improved boosting algorithms using confidence-rated predictions. **Machine Learning**, v. 37, n. 3, p. 297–336, 1999. Citation on page 42.

____. Boostexter: A boosting-based system for text categorization. **Machine Learning**, v. 39, n. 2/3, p. 135–168, 2000. Citation on page 25.

SECHIDIS, K.; TSOUMAKAS, G.; VLAHAVAS, I. P. On the stratification of multi-label data. In: GUNOPULOS, D.; HOFMANN, T.; MALERBA, D.; VAZIRGIANNIS, M. (Ed.). **Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III**. [S.l.]: Springer, 2011. (Lecture Notes in Computer Science, v. 6913), p. 145–158. Citations on pages 47, 106, and 133.

SEGRERA, S.; LUCAS, J. P.; GARCÍA, M. N. M. Information-theoretic measures for meta-learning. In: CORCHADO, E.; ABRAHAM, A.; PEDRYCZ, W. (Ed.). **Hybrid Artificial Intelligence Systems, Third International Workshop, HAIS 2008, Burgos, Spain, September 24-26, 2008. Proceedings**. [S.l.]: Springer, 2008. (Lecture Notes in Computer Science, v. 5271), p. 458–465. Citations on pages 68, 72, and 128.

SENGE, R.; COZ, J. J. del; HÜLLERMEIER, E. Rectifying classifier chains for multi-label classification. In: HENRICH, A.; SPERKER, H. (Ed.). **LWA 2013. Lernen, Wissen & Adaptivität, Workshop Proceedings Bamberg, 7.-9. October 2013**. [S.l.]: Universitätsbibliothek Bamberg, 2013. p. 151–158. Citations on pages 36, 37, and 45.

SMITH, K.; WOO, F.; CIESIELSKI, V.; IBRAHIM, R. Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks. In: **Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems**. [S.l.: s.n.], 2001. p. 357 – 362. Citations on pages 73 and 84.

SMITH, M. R.; MARTINEZ, T. R.; GIRAUD-CARRIER, C. G. An instance level analysis of data complexity. **Machine Learning**, v. 95, n. 2, p. 225–256, 2014. Citation on page 78.

SMITH-MILES, K. Cross-disciplinary perspectives on meta-learning for algorithm selection. **ACM Comput. Surv.**, v. 41, n. 1, p. 6:1–6:25, 2008. Citations on pages 27, 67, 68, 92, and 128.

SNOEK, C.; WORRING, M.; GEMERT, J. C. van; GEUSEBROEK, J.; SMEULDERS, A. W. M. The challenge problem for automated detection of 101 semantic concepts in multimedia. In: NAHRSTEDT, K.; TURK, M.; RUI, Y.; KLAS, W.; MAYER-PATEL, K. (Ed.). **Proceedings of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, USA, October 23-27, 2006**. [S.l.]: ACM, 2006. p. 421–430. Citations on pages 26 and 41.

SOARES, C.; PETRAK, J.; BRAZDIL, P. Sampling-based relative landmarks: Systematically test-driving algorithms before choosing. In: BRAZDIL, P.; JORGE, A. (Ed.). **Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving, 10th Portuguese Conference on Artificial Intelligence, EPIA 2001, Porto, Portugal, December 17-20, 2001, Proceedings**. [S.l.]: Springer, 2001. (Lecture Notes in Computer Science, v. 2258), p. 88–95. Citations on pages 67, 68, and 76.

SOHN, S. Y. Meta analysis of classification algorithms for pattern recognition. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 21, n. 11, p. 1137–1144, 1999. Citations on pages 66 and 67.

SONG, Q.; WANG, G.; WANG, C. Automatic recommendation of classification algorithms based on data set characteristics. **Pattern Recognition**, v. 45, n. 7, p. 2672–2689, 2012. Citations on pages 81 and 195.

SRIVASTAVA, A. N.; ZANE-ULMAN, B. Discovering recurring anomalies in text reports regarding complex space systems. In: **IEEE Aerospace Conference**. [S.l.: s.n.], 2005. p. 3853–3862. Citation on page 40.

SUN, Q.; PFAHRINGER, B. Pairwise meta-rules for better meta-learning-based algorithm ranking. **Machine Learning**, v. 93, n. 1, p. 141–161, 2013. Citations on pages 76 and 92.

TAN, P.; STEINBACH, M. S.; KUMAR, V. **Introduction to Data Mining**. [S.l.]: Addison-Wesley, 2005. ISBN 0-321-32136-7. Citations on pages 85 and 88.

TANAKA, E. A.; NOZAWA, S. R.; MACEDO, A. A.; BARANAUSKAS, J. A. A multi-label approach using binary relevance and decision trees applied to functional genomics. **Journal of Biomedical Informatics**, v. 54, p. 85–95, 2015. Citation on page 26.

TODOROVSKI, L.; BRAZDIL, P.; SOARES, C. Report on the experiments with feature selection in meta-level learning. In: **PKDD Workshop on Data Mining, Decision Support, Meta-Learning and Inductive Logic Programming**. [S.l.: s.n.], 2000. p. 27 – 39. Citations on pages 66, 70, 82, 87, 89, 92, and 185.

TROHIDIS, K.; TSOUMAKAS, G.; KALLIRIS, G.; VLAHAVAS, I. P. Multi-label classification of music into emotions. In: BELLO, J. P.; CHEW, E.; TURNBULL, D. (Ed.). **ISMIR 2008, 9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, PA, USA, September 14-18, 2008**. [S.l.: s.n.], 2008. p. 325–330. Citation on page 25.

____. Multi-label classification of music by emotion. **EURASIP J. Audio, Speech and Music Processing**, v. 2011, p. 4, 2011. Citation on page 41.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. **IJDWM**, v. 3, n. 3, p. 1–13, 2007. Citations on pages 26, 33, and 34.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Effective and efficient multilabel classification in domains with large number of labels. In: **Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Workshop on Mining Multidimensional Data**. [S.l.: s.n.], 2008. p. 30–44. Citations on pages 32 and 147.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. P. Mining multi-label data. In: MAIMON, O.; ROKACH, L. (Ed.). **Data Mining and Knowledge Discovery Handbook, 2nd ed**. [S.l.]: Springer, 2010. p. 667–685. Citations on pages 25, 32, 34, 42, 98, 99, 105, 122, 123, and 130.

____. Random k-labelsets for multilabel classification. **IEEE Trans. Knowl. Data Eng.**, v. 23, n. 7, p. 1079–1089, 2011. Citations on pages 27, 32, 35, 125, 133, and 147.

TSOUMAKAS, G.; Loza Mencía, E.; KATAKIS, I.; PARK, S.-H.; FÜRNKRANZ, J. On the Combination of Two Decompositive Multi-Label Classification Methods. In: **Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, Workshop on Preference Learning**. [S.l.: s.n.], 2009. p. 114–129. Citations on pages 38, 39, and 45.

TURNBULL, D.; BARRINGTON, L.; TORRES, D. A.; LANCKRIET, G. R. G. Semantic annotation and retrieval of music and sound effects. **IEEE Trans. Audio, Speech & Language Processing**, v. 16, n. 2, p. 467–476, 2008. Citation on page 41.

UEDA, N.; SAITO, K. Parametric mixture models for multi-labeled text. In: BECKER, S.; THRUN, S.; OBERMAYER, K. (Ed.). **Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]**. [S.l.]: MIT Press, 2002. p. 721–728. Citation on page 25.

VANSCHOREN, J. **Understanding Machine Learning Performance with Experiment Databases**. Phd Thesis (PhD Thesis) — Leuven Univeristy, 2010. Citations on pages 68, 71, 81, and 185.

VANSCHOREN, J.; BLOCKEEL, H.; PFAHRINGER, B.; HOLMES, G. Experiment databases - A new way to share, organize and learn from experiments. **Machine Learning**, v. 87, n. 2, p. 127–158, 2012. Citations on pages 64 and 68.

VANSCHOREN, J.; RIJN, J. N. van; BISCHL, B.; TORGO, L. Openml: networked science in machine learning. **SIGKDD Explorations**, v. 15, n. 2, p. 49–60, 2013. Citations on pages 65 and 92.

VILALTA, R. Understanding accuracy performance through concept characterization and algorithm analysis. In: **ECML Workshop on Recent Advances in Meta-Learning and Future Work**. [S.l.: s.n.], 1999. p. 3–9. Citations on pages 81 and 195.

VILALTA, R.; DRISSI, Y. A characterization of difficult problems in classification. In: WANI, M. A.; ARABNIA, H. R.; CIOS, K. J.; HAFEEZ, K.; KENDALL, G. (Ed.). **Proceedings of the 2002 International Conference on Machine Learning and Applications - ICMLA 2002, June 24-27, 2002, Las Vegas, Nevada, USA**. [S.l.]: CSREA Press, 2002. p. 133–138. Citations on pages 81 and 194.

____. A perspective view and survey of meta-learning. **Artif. Intell. Rev.**, v. 18, n. 2, p. 77–95, 2002. Citation on page 27.

VUKICEVIC, M.; RADOVANOVIC, S.; DELIBASIC, B.; SUKNOVIC, M. Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures. **IJDMB**, v. 14, n. 2, p. 101–119, 2016. Citations on pages 68, 77, 192, and 193.

WANG, G.; SONG, Q.; SUN, H.; ZHANG, X.; XU, B.; ZHOU, Y. A feature subset selection algorithm automatic recommendation method. **J. Artif. Intell. Res.**, v. 47, p. 1–34, 2013. Citation on page 86.

WANG, G.; SONG, Q.; ZHU, X. An improved data characterization method and its application in classification algorithm recommendation. **Appl. Intell.**, v. 43, n. 4, p. 892–912, 2015. Citations on pages 81 and 127.

WEVER, M.; MOHR, F.; HÜLLERMEIER, E. Automated multi-label classification based on ml-plan. **CoRR**, abs/1811.04060, 2018. Available: <http://arxiv.org/abs/1811.04060>. Citation on page 33.

WEVER, M. D.; MOHR, F.; TORNEDE, A.; HüLLERMEIER, E. Automating multi-label classification extending ml-plan. In: **6th ICML Workshop on Automated Machine Learning**. [S.l.: s.n.], 2019. Citations on pages 33 and 60.

WOLPERT, D. H. Stacked generalization. **Neural Networks**, v. 5, n. 2, p. 241–259, 1992. Citations on pages 37 and 64.

WOLPERT, D. H.; MACREADY, W. G. **No free lunch theorems for search**. [S.l.], 1995. 1–38 p. Citation on page 27.

XIAO, X.; WU, Z.-C.; CHOU, K.-C. A Multi-Label Classifier for Predicting the Subcellular Localization of Gram-Negative Bacterial Proteins with Both Single and Multiple Sites. **PLoS ONE**, v. 6, n. 6, p. e20592, 2011. Citation on page 26.

YANG, Y. An evaluation of statistical approaches to text categorization. **Inf. Retr.**, v. 1, n. 1-2, p. 69–90, 1999. Citations on pages 43, 105, and 131.

____. A study on thresholding strategies for text categorization. In: CROFT, W. B.; HARPER, D. J.; KRAFT, D. H.; ZOBEL, J. (Ed.). **SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA**. [S.l.]: ACM, 2001. p. 137–145. Citations on pages 27 and 100.

YANG, Y.; LU, B. Prediction of protein subcellular multi-locations with a min-max modular support vector machine. In: WANG, J.; YI, Z.; ZURADA, J. M.; LU, B.; YIN, H. (Ed.). **Advances in Neural Networks - ISNN 2006, Third International Symposium on Neural Networks, Chengdu, China, May 28 - June 1, 2006, Proceedings, Part III**. [S.l.]: Springer, 2006. (Lecture Notes in Computer Science, v. 3973), p. 667–673. Citation on page 26.

ZHANG, M.; LI, Y.; LIU, X.; GENG, X. Binary relevance for multi-label learning: an overview. **Frontiers Comput. Sci.**, v. 12, n. 2, p. 191–202, 2018. Citations on pages 98 and 124.

ZHANG, M.; WU, L. Lift: Multi-label learning with label-specific features. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 37, n. 1, p. 107–120, 2015. Citation on page 32.

ZHANG, M.; ZHOU, Z. A review on multi-label learning algorithms. **IEEE Trans. Knowl. Data Eng.**, v. 26, n. 8, p. 1819–1837, 2014. Citations on pages 26, 34, and 42.

ZHANG, X.; SONG, Q. A Multi-Label Learning Based Kernel Automatic Recommendation Method for Support Vector Machine. **PLOS ONE**, v. 10, n. 4, p. 1–30, 2015. Citation on page 26.

ZHENG, Y.; MOBASHER, B.; BURKE, R. D. Context recommendation using multi-label classification. In: **2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Warsaw, Poland, August 11-14, 2014 - Volume I**. [S.l.]: IEEE Computer Society, 2014. p. 288–295. Citation on page 26.

ZHOU, T.; TAO, D.; WU, X. Compressed labeling on distilled labelsets for multi-label learning. **Machine Learning**, v. 88, n. 1-2, p. 69–126, 2012. Citations on pages 35, 98, and 124.

ZHOU, Z.; ZHANG, M. Multi-instance multi-label learning with application to scene classification. In: SCHÖLKOPF, B.; PLATT, J. C.; HOFMANN, T. (Ed.). **Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006**. [S.l.]: MIT Press, 2006. p. 1609–1616. Citations on pages 32, 41, and 122.

ZOU, H.-L. A New Multi-label Classifier for Identifying the Functional Types of Singleplex and Multiplex Antimicrobial Peptides. **International Journal of Peptide Research and Therapeutics**, Springer, v. 22, n. 2, p. 281–287, 2016. Citation on page 26.

ZUFFEREY, D.; HOFER, T.; HENNEBERT, J.; SCHUMACHER, M. I.; INGOLD, R.; BROMURI, S. Performance comparison of multi-label learning algorithms on clinical data for chronic diseases. **Comp. in Bio. and Med.**, v. 65, p. 34–43, 2015. Citation on page 33.

APPENDIX

A

PERFORMANCE RESULTS OF THE BEST COMBINATIONS BETWEEN STRATEGIES AND BASE ALGORITHMS

This section presents the strategy/base-algorithm's ranking over all datasets (Figures 26 to 33) and the performance value obtained for each strategy when combined with the best base algorithm (Tables 39 to 46). The median ranking is used to select the base-algorithm for each strategy.

Table 39 – Results of best strategies for the *F1* ↑ measure.

Data set	EBR_{XGB}	ECC_{XGB}	$PruDent_{RF}$	MBR_{LR}	$RDBR_{SVMt}$	DBR_{SVMt}	BR_{RF}	NS_{RF}	$BR+_{SVM}$	CC_{RF}
20NG	0.7612 (0.003)	0.7323 (0.008)	0.7781 (0.002)	0.7659 (0.003)	0.7474 (0.004)	0.7477 (0.005)	**0.7782 (0.003)**	0.7735 (0.002)	0.7663 (0.003)	0.7725 (0.003)
birds	0.5471 (0.022)	**0.5472 (0.017)**	0.5283 (0.018)	0.3837 (0.134)	0.3835 (0.111)	0.3862 (0.110)	0.5294 (0.023)	0.5263 (0.023)	0.4088 (0.119)	0.5277 (0.023)
cal500	0.4476 (0.004)	**0.4529 (0.004)**	0.3414 (0.006)	0.3414 (0.010)	0.3693 (0.009)	0.3406 (0.011)	0.3458 (0.006)	0.3395 (0.012)	0.3261 (0.009)	0.3362 (0.006)
corel5k	0.2211 (0.003)	**0.2317 (0.003)**	0.1680 (0.004)	0.1717 (0.005)	0.2160 (0.003)	0.2247 (0.005)	0.1675 (0.004)	0.1735 (0.006)	0.1696 (0.003)	0.1698 (0.003)
emotions	0.6521 (0.010)	0.6528 (0.012)	0.6454 (0.013)	0.6366 (0.021)	0.6468 (0.019)	0.6473 (0.015)	0.6425 (0.022)	0.6377 (0.019)	**0.6640 (0.013)**	0.6377 (0.019)
enron	0.6157 (0.009)	**0.6176 (0.009)**	0.5687 (0.015)	0.5291 (0.007)	0.5540 (0.012)	0.5549 (0.013)	0.5642 (0.013)	0.5646 (0.011)	0.5405 (0.009)	0.5670 (0.015)
fapesp	**0.5795 (0.018)**	0.5778 (0.023)	0.5339 (0.032)	0.5754 (0.027)	0.5617 (0.024)	0.5672 (0.033)	0.5370 (0.038)	0.5301 (0.035)	0.5717 (0.024)	0.5301 (0.035)
flags	0.7117 (0.016)	0.7198 (0.018)	0.7356 (0.013)	0.7310 (0.013)	0.7209 (0.015)	0.6918 (0.029)	**0.7377 (0.018)**	0.7364 (0.014)	0.7227 (0.007)	0.7371 (0.016)
image	**0.7000 (0.008)**	0.6961 (0.008)	0.6905 (0.007)	0.6807 (0.010)	0.6756 (0.027)	0.6843 (0.012)	0.6893 (0.007)	0.6890 (0.006)	0.6812 (0.009)	0.6890 (0.006)
langlog	**0.3181 (0.008)**	0.3179 (0.010)	0.2875 (0.015)	0.2861 (0.010)	0.2845 (0.010)	0.2779 (0.008)	0.2883 (0.010)	0.2846 (0.010)	0.2857 (0.008)	0.2874 (0.017)
mediamill	0.6436 (0.001)	0.6496 (0.001)	0.6161 (0.001)	0.6005 (0.001)	**0.6539 (0.002)**	0.6514 (0.002)	0.6204 (0.001)	0.6114 (0.001)	0.5844 (0.002)	0.6108 (0.001)
medical	**0.8481 (0.010)**	0.8461 (0.012)	0.7746 (0.010)	0.8425 (0.004)	0.8397 (0.016)	0.8426 (0.016)	0.7674 (0.013)	0.7711 (0.013)	0.8364 (0.005)	0.7754 (0.011)
msd-195	0.3233 (0.010)	**0.3293 (0.008)**	0.2404 (0.009)	0.2737 (0.007)	0.2784 (0.010)	0.2669 (0.007)	0.2353 (0.009)	0.2427 (0.008)	0.2771 (0.005)	0.2420 (0.008)
ohsumed	0.5812 (0.003)	**0.5842 (0.003)**	0.5227 (0.004)	0.5570 (0.002)	0.5524 (0.002)	0.5524 (0.003)	0.5220 (0.003)	0.5204 (0.003)	0.5562 (0.002)	0.5207 (0.003)
scene	**0.8004 (0.002)**	0.7898 (0.010)	0.7908 (0.005)	0.7904 (0.004)	0.7965 (0.005)	0.7955 (0.006)	0.7893 (0.004)	0.7883 (0.001)	0.7921 (0.005)	0.7879 (0.000)
slashdot	0.5645 (0.006)	0.5493 (0.007)	**0.5764 (0.007)**	0.5697 (0.012)	0.3688 (0.228)	0.3742 (0.228)	0.5757 (0.008)	0.5740 (0.007)	0.5668 (0.009)	0.5721 (0.005)
stackex	**0.4372 (0.011)**	0.4366 (0.011)	0.3611 (0.008)	0.3846 (0.013)	0.3296 (0.050)	0.3355 (0.039)	0.3591 (0.008)	0.3580 (0.007)	0.3434 (0.008)	0.3546 (0.009)
tmc2007	0.7381 (0.002)	**0.7407 (0.002)**	0.7341 (0.002)	0.6835 (0.002)	0.6947 (0.012)	0.7186 (0.015)	0.7340 (0.001)	0.7352 (0.002)	0.6802 (0.002)	0.7345 (0.002)
yeast	0.6561 (0.004)	**0.6569 (0.005)**	0.6023 (0.003)	0.6342 (0.003)	0.6468 (0.006)	0.6430 (0.004)	0.6082 (0.003)	0.6105 (0.005)	0.6513 (0.004)	0.6095 (0.005)
yelp8	0.7410 (0.004)	**0.7540 (0.003)**	0.6889 (0.002)	0.6039 (0.003)	0.7301 (0.003)	0.7340 (0.004)	0.6885 (0.003)	0.7087 (0.003)	0.6072 (0.005)	0.7091 (0.003)

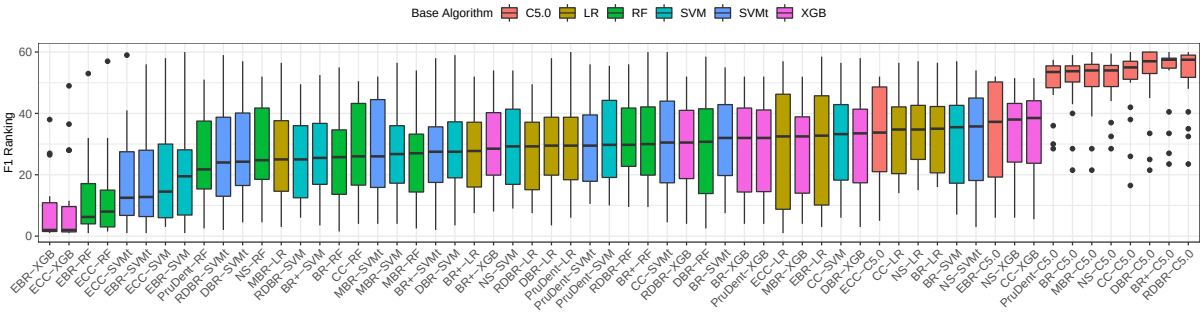


Figure 26 – Strategy/base-algorithm's rankings for the *F1* measure.

Table 40 – Results of best strategies for the *hamming-loss* ↓ measure.

Data set	$PruDent_{RF}$	BR_{RF}	MBR_{RF}	BR^{+}_{RF}	DBR_{RF}	$RDBR_{RF}$	CC_{RF}	NS_{RF}	EBR_{RF}	ECC_{RF}
20NG	**0.0228 (0.000)**	**0.0228 (0.000)**	**0.0228 (0.000)**	0.0242 (0.000)	0.0229 (0.000)	0.0246 (0.001)	0.0234 (0.000)	0.0233 (0.000)	0.0238 (0.000)	0.0243 (0.000)
birds	0.0922 (0.004)	0.0924 (0.003)	0.0927 (0.003)	**0.0921 (0.004)**	**0.0921 (0.004)**	**0.0921 (0.003)**	0.0927 (0.004)	0.0927 (0.004)	0.1094 (0.003)	0.1107 (0.006)
cal500	0.1686 (0.001)	0.1694 (0.001)	0.1694 (0.001)	0.1668 (0.001)	**0.1666 (0.001)**	0.1669 (0.001)	0.1683 (0.001)	0.1962 (0.003)	0.1932 (0.004)	0.1876 (0.003)
corel5k	**0.0167 (0.000)**	**0.0167 (0.000)**	**0.0167 (0.000)**	0.0169 (0.000)	0.0169 (0.000)	0.0169 (0.000)	**0.0167 (0.000)**	0.0181 (0.000)	0.0215 (0.000)	0.0218 (0.000)
emotions	**0.1865 (0.007)**	**0.1865 (0.009)**	0.1872 (0.008)	0.1868 (0.008)	0.1873 (0.008)	0.1867 (0.009)	0.1898 (0.009)	0.1898 (0.009)	0.1880 (0.005)	0.1889 (0.005)
enron	**0.0570 (0.002)**	0.0571 (0.002)	0.0572 (0.002)	0.0574 (0.001)	0.0573 (0.001)	0.0578 (0.001)	0.0572 (0.002)	0.0638 (0.002)	0.0605 (0.001)	0.0614 (0.001)
fapesp	0.0630 (0.004)	**0.0625 (0.005)**	**0.0625 (0.004)**	0.0634 (0.006)	0.0635 (0.005)	0.0631 (0.006)	0.0635 (0.004)	0.0635 (0.004)	0.0688 (0.005)	0.0688 (0.004)
flags	0.2366 (0.012)	0.2341 (0.012)	**0.2338 (0.016)**	0.2370 (0.014)	0.2380 (0.013)	0.2364 (0.012)	0.2350 (0.012)	0.2353 (0.012)	0.2368 (0.012)	0.2363 (0.009)
image	**0.1457 (0.004)**	0.1461 (0.003)	0.1465 (0.002)	0.1458 (0.004)	0.1458 (0.003)	0.1462 (0.003)	0.1459 (0.003)	0.1459 (0.003)	0.1551 (0.003)	0.1547 (0.003)
langlog	0.0435 (0.001)	0.0435 (0.001)	**0.0434 (0.001)**	0.0438 (0.001)	0.0439 (0.001)	0.0438 (0.001)	0.0436 (0.001)	0.0437 (0.001)	0.0503 (0.001)	0.0502 (0.001)
mediamill	0.0275 (0.000)	**0.0273 (0.000)**	**0.0273 (0.000)**	0.0280 (0.000)	0.0281 (0.000)	0.0281 (0.000)	0.0277 (0.000)	0.0277 (0.000)	0.0278 (0.000)	0.0282 (0.000)
medical	0.0263 (0.001)	0.0271 (0.001)	0.0270 (0.001)	0.0275 (0.001)	0.0275 (0.001)	0.0272 (0.001)	**0.0262 (0.001)**	0.0267 (0.001)	0.0293 (0.001)	0.0301 (0.001)
msd-195	0.0717 (0.001)	0.0716 (0.000)	0.0717 (0.001)	0.0723 (0.001)	0.0728 (0.001)	0.0721 (0.001)	**0.0715 (0.000)**	**0.0715 (0.000)**	0.0906 (0.001)	0.0929 (0.001)
ohsumed	**0.0576 (0.000)**	0.0577 (0.000)	**0.0576 (0.000)**	0.0578 (0.000)	0.0577 (0.000)	0.0577 (0.000)	0.0578 (0.000)	0.0578 (0.000)	0.0615 (0.000)	0.0616 (0.000)
scene	**0.0747 (0.001)**	0.0751 (0.001)	0.0748 (0.002)	0.0751 (0.001)	0.0751 (0.001)	0.0755 (0.000)	0.0753 (0.000)	0.0752 (0.000)	0.0782 (0.000)	0.0778 (0.000)
slashdot	0.0530 (0.001)	0.0531 (0.001)	0.0530 (0.001)	0.0547 (0.001)	**0.0528 (0.001)**	0.0545 (0.001)	0.0533 (0.001)	0.0531 (0.001)	0.0583 (0.002)	0.0591 (0.002)
stackex	**0.0257 (0.000)**	0.0258 (0.000)	**0.0257 (0.000)**	0.0259 (0.000)	0.0259 (0.000)	0.0258 (0.000)	0.0259 (0.000)	0.0258 (0.000)	0.0354 (0.001)	0.0356 (0.000)
tmc2007	0.0462 (0.000)	0.0462 (0.000)	**0.0461 (0.000)**	0.0465 (0.000)	0.0466 (0.000)	0.0466 (0.000)	0.0462 (0.000)	**0.0461 (0.000)**	0.0502 (0.000)	0.0504 (0.000)
yeast	0.1908 (0.002)	0.1902 (0.002)	**0.1901 (0.002)**	0.1921 (0.002)	0.1979 (0.002)	0.1941 (0.004)	0.1919 (0.002)	0.1918 (0.002)	0.1920 (0.002)	0.1951 (0.003)
yelp8	0.1426 (0.001)	0.1421 (0.001)	0.1421 (0.001)	0.1393 (0.001)	0.1394 (0.001)	0.1393 (0.001)	**0.1369 (0.001)**	0.1370 (0.001)	0.1616 (0.001)	0.1571 (0.007)

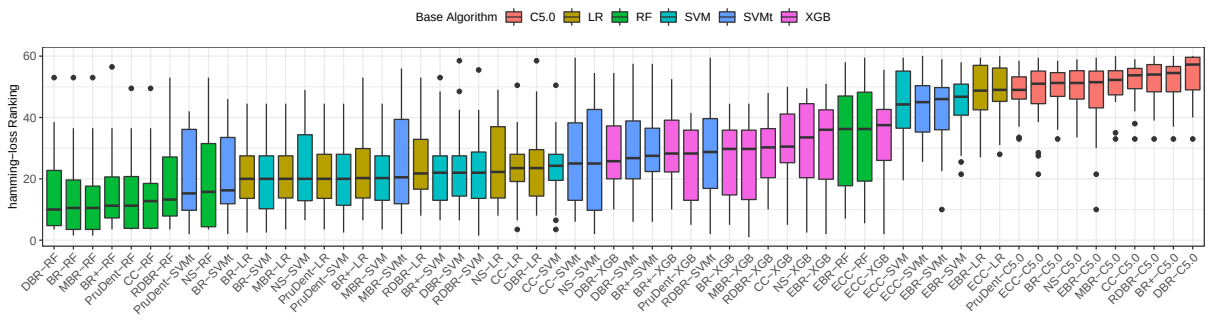


Figure 27 – Strategy/base-algorithm's rankings for the *hamming-loss* measure.

Table 41 – Results of best strategies for *macro-F1* ↑ measure.

Data set	EBR_{XGB}	ECC_{XGB}	MBR_{XGB}	BR_{XGB}	$PruDent_{XGB}$	$RDBR_{SVMt}$	DBR_{SVMt}	CC_{XGB}	NS_{XGB}	BR^{+}_{SVMt}
20NG	**0.7582**	0.7333	0.7546	0.7544	0.7544	0.7450	0.7452	0.7008	0.7409	0.7451
	(0.002)	(0.005)	(0.003)	(0.003)	(0.003)	(0.004)	(0.005)	(0.004)	(0.003)	(0.005)
birds	0.4511	**0.4512**	0.4459	0.4459	0.4459	0.2766	0.2788	0.4465	0.4307	0.2699
	(0.029)	**(0.032)**	(0.035)	(0.035)	(0.035)	(0.124)	(0.125)	(0.035)	(0.037)	(0.125)
cal500	**0.2035**	0.2031	0.1378	0.1380	0.1288	0.1688	0.1049	0.1369	0.1661	0.1185
	(0.007)	(0.008)	(0.005)	(0.005)	(0.004)	(0.011)	(0.008)	(0.005)	(0.010)	(0.006)
corel5k	0.0780	0.0751	0.0608	0.0577	0.0572	0.0917	0.0932	0.0598	0.0605	**0.0940**
	(0.002)	(0.004)	(0.003)	(0.002)	(0.003)	(0.004)	(0.006)	(0.004)	(0.004)	**(0.008)**
emotions	0.6644	**0.6694**	0.6369	0.6369	0.6514	0.6626	0.6655	0.6322	0.6324	0.6634
	(0.009)	**(0.009)**	(0.010)	(0.010)	(0.011)	(0.014)	(0.009)	(0.008)	(0.008)	(0.011)
enron	**0.2725**	0.2722	0.2523	0.2524	0.2535	0.1817	0.1820	0.2578	0.2508	0.1826
	(0.006)	(0.006)	(0.008)	(0.008)	(0.008)	(0.009)	(0.009)	(0.006)	(0.008)	(0.009)
fapesp	**0.4884**	0.4863	0.4498	0.4498	0.4513	0.4712	0.4663	0.4500	0.4462	0.4559
	(0.025)	(0.028)	(0.029)	(0.029)	(0.027)	(0.022)	(0.038)	(0.030)	(0.028)	(0.041)
flags	0.6431	0.6483	0.6449	0.6449	0.6435	0.6415	0.6293	**0.6601**	0.6531	0.6335
	(0.030)	(0.032)	(0.028)	(0.028)	(0.026)	(0.042)	(0.031)	**(0.026)**	(0.028)	(0.031)
image	**0.6923**	0.6885	0.6851	0.6851	0.6891	0.6714	0.6795	0.6666	0.6666	0.6765
	(0.008)	(0.009)	(0.008)	(0.008)	(0.009)	(0.027)	(0.012)	(0.009)	(0.009)	(0.019)
langlog	0.1854	**0.1865**	0.1616	0.1616	0.1616	0.1373	0.1342	0.1603	0.1604	0.1375
	(0.015)	**(0.016)**	(0.014)	(0.014)	(0.014)	(0.009)	(0.010)	(0.013)	(0.013)	(0.010)
mediamill	0.3277	0.2960	0.2355	0.2396	0.2387	**0.3913**	0.3865	0.2331	0.2306	0.3874
	(0.004)	(0.006)	(0.004)	(0.005)	(0.005)	**(0.003)**	(0.004)	(0.004)	(0.005)	(0.005)
medical	0.7185	0.7198	0.7253	0.7253	0.7253	0.7255	**0.7338**	0.7262	0.7107	0.7284
	(0.021)	(0.022)	(0.024)	(0.024)	(0.024)	(0.017)	**(0.017)**	(0.022)	(0.023)	(0.021)
msd-195	0.1744	**0.1779**	0.1234	0.1234	0.1241	0.1555	0.1486	0.1216	0.1220	0.1539
	(0.009)	**(0.008)**	(0.009)	(0.009)	(0.008)	(0.012)	(0.008)	(0.009)	(0.009)	(0.006)
ohsumed	0.4465	**0.4479**	0.4137	0.4139	0.4139	0.4177	0.4180	0.4129	0.4095	0.4180
	(0.003)	**(0.004)**	(0.002)	(0.002)	(0.002)	(0.004)	(0.003)	(0.004)	(0.003)	(0.004)
scene	**0.8009**	0.7945	0.7920	0.7920	0.7920	0.8005	0.7989	0.7777	0.7795	0.7980
	(0.004)	(0.013)	(0.010)	(0.010)	(0.010)	(0.006)	(0.004)	(0.001)	(0.002)	(0.005)
slashdot	**0.4258**	0.4174	0.4148	0.4148	0.4148	0.2385	0.2443	0.3893	0.4031	0.2399
	(0.008)	(0.015)	(0.008)	(0.008)	(0.008)	(0.227)	(0.230)	(0.011)	(0.007)	(0.228)
stackex	**0.2968**	0.2907	0.2580	0.2586	0.2586	0.1687	0.1778	0.2577	0.2349	0.1767
	(0.009)	(0.011)	(0.010)	(0.010)	(0.010)	(0.052)	(0.038)	(0.009)	(0.011)	(0.051)
tmc2007	0.6526	**0.6543**	0.6406	0.6323	0.6323	0.5978	0.6319	0.6286	0.6313	0.6237
	(0.011)	**(0.012)**	(0.014)	(0.013)	(0.013)	(0.008)	(0.023)	(0.011)	(0.013)	(0.028)
yeast	0.4328	0.4259	0.4019	0.4019	0.3961	**0.4589**	0.4512	0.4221	0.4223	0.4546
	(0.004)	(0.009)	(0.006)	(0.006)	(0.006)	**(0.007)**	(0.005)	(0.004)	(0.004)	(0.008)
yelp8	0.7087	**0.7141**	0.6970	0.6832	0.6846	0.6896	0.6893	0.6921	0.6982	0.6864
	(0.005)	**(0.004)**	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.005)	(0.004)

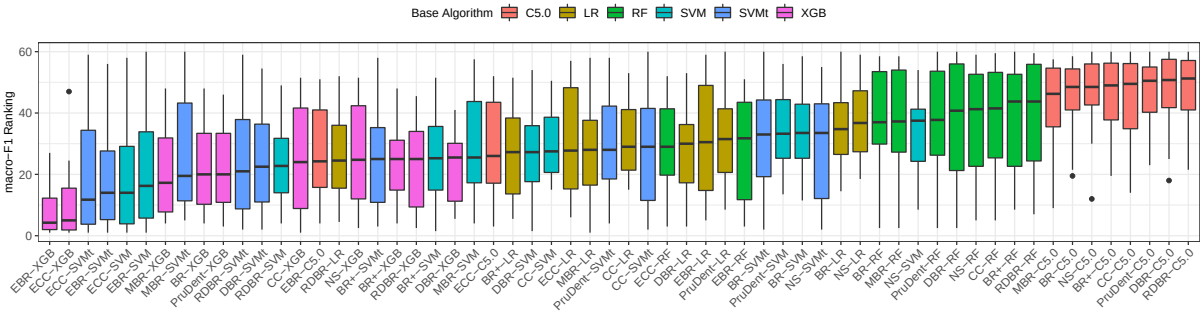


Figure 28 – Strategy/base-algorithm's rankings for the *macro-F1* measure.

Table 42 – Results of best strategies for *macro-precision* ↑ measure.

Data set	MBR_{RF}	CC_{RF}	$RDBR_{RF}$	NS_{RF}	BR^{+}_{RF}	$PruDent_{RF}$	DBR_{RF}	BR_{RF}	ECC_{XGB}	EBR_{XGB}
20NG	0.7851 (0.002)	0.7824 (0.003)	0.7871 (0.009)	0.7827 (0.002)	**0.7895 (0.008)**	0.7851 (0.003)	0.7852 (0.003)	0.7849 (0.003)	0.7236 (0.014)	0.7511 (0.003)
birds	**0.6179 (0.078)**	0.6161 (0.054)	0.6043 (0.089)	0.6167 (0.056)	0.6047 (0.087)	0.6131 (0.075)	0.6046 (0.087)	0.5916 (0.080)	0.5311 (0.060)	0.5203 (0.053)
cal500	0.1852 (0.012)	0.1745 (0.019)	0.1718 (0.021)	0.2061 (0.019)	0.1690 (0.019)	0.1797 (0.014)	0.1695 (0.015)	0.1791 (0.016)	0.2084 (0.012)	**0.2136 (0.008)**
corel5k	0.0974 (0.010)	0.0821 (0.011)	0.0897 (0.016)	0.0915 (0.011)	0.0976 (0.009)	0.0937 (0.012)	0.0929 (0.006)	0.0882 (0.009)	**0.1121 (0.012)**	0.1082 (0.007)
emotions	**0.7299 (0.011)**	0.7232 (0.013)	0.7205 (0.011)	0.7232 (0.013)	0.7209 (0.010)	0.7258 (0.010)	0.7203 (0.011)	0.7281 (0.010)	0.6843 (0.011)	0.6724 (0.015)
enron	0.3651 (0.017)	0.3697 (0.024)	**0.3769 (0.019)**	0.3524 (0.024)	0.3725 (0.018)	0.3696 (0.019)	0.3733 (0.017)	0.3625 (0.019)	0.3450 (0.017)	0.3484 (0.019)
fapesp	0.5127 (0.082)	0.4859 (0.069)	0.5057 (0.088)	0.4859 (0.069)	0.4980 (0.084)	0.4984 (0.072)	0.4922 (0.081)	0.5291 (0.087)	0.5119 (0.028)	**0.5378 (0.039)**
flags	0.7101 (0.064)	0.7065 (0.057)	0.7227 (0.062)	0.7069 (0.057)	0.7235 (0.059)	0.7117 (0.059)	**0.7238 (0.059)**	0.7145 (0.063)	0.6743 (0.031)	0.6785 (0.044)
image	0.7472 (0.005)	0.7519 (0.007)	0.7518 (0.007)	0.7520 (0.007)	**0.7533 (0.008)**	0.7486 (0.008)	0.7504 (0.008)	0.7482 (0.007)	0.6990 (0.010)	0.6987 (0.008)
langlog	0.1988 (0.018)	0.1961 (0.027)	0.1953 (0.015)	0.1918 (0.014)	0.1935 (0.015)	0.1964 (0.020)	0.1935 (0.015)	0.1964 (0.018)	0.2206 (0.031)	**0.2222 (0.024)**
mediamill	0.5550 (0.013)	0.5732 (0.019)	0.5949 (0.015)	0.5745 (0.014)	0.5944 (0.016)	0.5601 (0.011)	0.5985 (0.020)	0.5599 (0.014)	**0.6225 (0.022)**	0.5805 (0.016)
medical	0.7471 (0.028)	0.7370 (0.037)	0.7312 (0.036)	0.7405 (0.033)	0.7329 (0.035)	0.7554 (0.023)	0.7303 (0.033)	0.7468 (0.030)	**0.7624 (0.038)**	0.7490 (0.036)
msd-195	0.2572 (0.033)	0.2406 (0.028)	0.2735 (0.040)	0.2464 (0.028)	0.2756 (0.037)	0.2757 (0.033)	0.2681 (0.032)	0.2605 (0.026)	**0.2783 (0.023)**	0.2647 (0.030)
ohsumed	0.5991 (0.039)	0.5941 (0.037)	0.6007 (0.035)	**0.6028 (0.026)**	0.5866 (0.022)	0.5974 (0.025)	0.5894 (0.027)	0.5894 (0.017)	0.5538 (0.014)	0.5540 (0.012)
scene	0.8130 (0.004)	0.8124 (0.003)	0.8136 (0.003)	0.8133 (0.003)	**0.8159 (0.004)**	0.8136 (0.004)	0.8145 (0.002)	0.8129 (0.001)	0.8038 (0.014)	0.7990 (0.005)
slashdot	0.4932 (0.027)	0.5187 (0.036)	**0.5216 (0.052)**	0.5100 (0.029)	0.5184 (0.052)	0.5016 (0.028)	0.5113 (0.041)	0.5099 (0.041)	0.4540 (0.037)	0.4541 (0.016)
stackex	0.2196 (0.029)	0.2132 (0.017)	0.2211 (0.025)	0.2290 (0.023)	0.2259 (0.025)	0.2235 (0.023)	0.2277 (0.020)	0.2180 (0.033)	**0.3540 (0.019)**	0.3453 (0.018)
tmc2007	0.8738 (0.018)	0.8710 (0.018)	0.8697 (0.018)	0.8716 (0.017)	0.8696 (0.018)	0.8737 (0.016)	0.8691 (0.017)	**0.8740 (0.017)**	0.7526 (0.010)	0.7527 (0.012)
yeast	**0.7062 (0.024)**	0.6583 (0.025)	0.6760 (0.030)	0.6575 (0.025)	0.6616 (0.030)	0.6982 (0.025)	0.6565 (0.025)	0.7040 (0.027)	0.5704 (0.026)	0.5427 (0.017)
yelp8	**0.8820 (0.006)**	0.8649 (0.004)	0.8613 (0.004)	0.8673 (0.005)	0.8626 (0.004)	0.8808 (0.004)	0.8622 (0.004)	**0.8820 (0.005)**	0.7537 (0.006)	0.7429 (0.015)

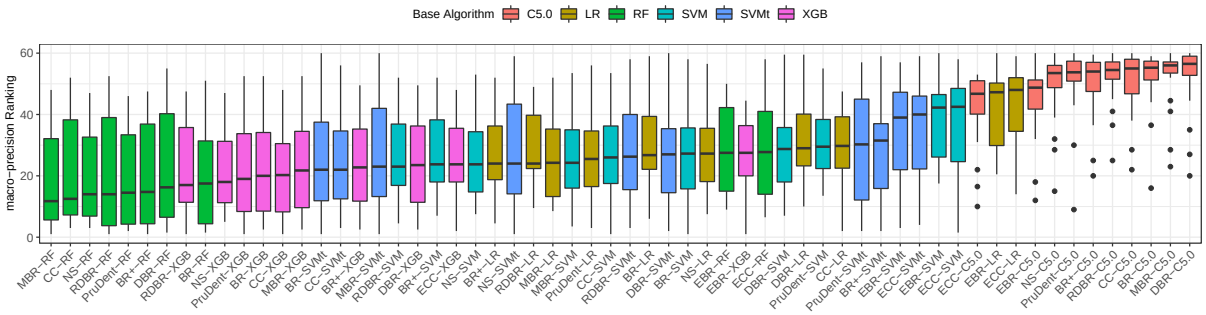


Figure 29 – Strategy/base-algorithm's rankings for the *macro-precision* measure.

Table 43 – Results of best strategies for *macro-recall* ↑ measure.

Data set	EBR_{SVM}	ECC_{XGB}	DBR_{XGB}	$RDBR_{XGB}$	MBR_{XGB}	BR_{XGB}	$BR+_{XGB}$	$PruDent_{XGB}$	CC_{XGB}	NS_{XGB}
20NG	**0.7740**	0.7634	0.7505	0.7459	0.7619	0.7615	0.7468	0.7615	0.6734	0.7364
	(0.003)	(0.016)	(0.003)	(0.002)	(0.003)	(0.003)	(0.003)	(0.003)	(0.007)	(0.004)
birds	0.1323	**0.4369**	0.3761	0.3761	0.3770	0.3770	0.3761	0.3770	0.3771	0.3605
	(0.046)	**(0.038)**	(0.027)	(0.027)	(0.026)	(0.026)	(0.027)	(0.026)	(0.026)	(0.028)
cal500	0.2186	**0.2318**	0.1089	0.1113	0.1200	0.1201	0.1085	0.1109	0.1190	0.1548
	(0.015)	**(0.009)**	(0.005)	(0.006)	(0.005)	(0.005)	(0.005)	(0.004)	(0.005)	(0.010)
corel5k	**0.0826**	0.0672	0.0394	0.0378	0.0450	0.0413	0.0400	0.0407	0.0469	0.0471
	(0.003)	(0.004)	(0.003)	(0.003)	(0.002)	(0.002)	(0.003)	(0.002)	(0.004)	(0.003)
emotions	**0.6680**	0.6655	0.6576	0.6560	0.6064	0.6064	0.6542	0.6309	0.6091	0.6089
	(0.024)	(0.018)	(0.025)	(0.025)	(0.020)	(0.020)	(0.023)	(0.021)	(0.014)	(0.013)
enron	0.2309	**0.2583**	0.2370	0.2289	0.2146	0.2145	0.2288	0.2167	0.2239	0.2220
	(0.010)	**(0.006)**	(0.005)	(0.004)	(0.007)	(0.007)	(0.006)	(0.007)	(0.007)	(0.007)
fapesp	0.4732	**0.5155**	0.4227	0.4245	0.4245	0.4245	0.4226	0.4254	0.4239	0.4163
	(0.024)	**(0.025)**	(0.024)	(0.023)	(0.027)	(0.027)	(0.025)	(0.026)	(0.026)	(0.024)
flags	0.6227	0.6440	0.6342	0.6429	0.6347	0.6347	0.6340	0.6338	**0.6505**	0.6418
	(0.028)	(0.033)	(0.023)	(0.028)	(0.031)	(0.031)	(0.024)	(0.025)	**(0.025)**	(0.027)
image	0.6777	**0.6808**	0.6476	0.6409	0.6473	0.6473	0.6426	0.6557	0.6232	0.6232
	(0.009)	**(0.011)**	(0.008)	(0.009)	(0.008)	(0.008)	(0.009)	(0.011)	(0.010)	(0.010)
langlog	0.1604	**0.1941**	0.1469	0.1476	0.1495	0.1495	0.1475	0.1495	0.1486	0.1485
	(0.006)	**(0.015)**	(0.012)	(0.011)	(0.012)	(0.012)	(0.012)	(0.012)	(0.012)	(0.012)
mediamill	**0.3256**	0.2315	0.1683	0.1648	0.1653	0.1677	0.1660	0.1666	0.1649	0.1621
	(0.004)	(0.004)	(0.003)	(0.003)	(0.002)	(0.003)	(0.002)	(0.003)	(0.003)	(0.003)
medical	0.6959	0.7236	0.7323	0.7301	0.7372	**0.7373**	0.7306	**0.7373**	0.7351	0.7011
	(0.012)	(0.021)	(0.027)	(0.029)	(0.026)	**(0.026)**	(0.026)	**(0.026)**	(0.026)	(0.026)
msd-195	**0.2102**	0.1734	0.0937	0.1000	0.0924	0.0924	0.0977	0.0941	0.0948	0.0952
	(0.004)	(0.007)	(0.007)	(0.008)	(0.006)	(0.006)	(0.007)	(0.005)	(0.006)	(0.006)
ohsumed	**0.4245**	0.4125	0.3552	0.3536	0.3490	0.3491	0.3512	0.3491	0.3502	0.3453
	(0.005)	(0.004)	(0.002)	(0.002)	(0.003)	(0.003)	(0.002)	(0.003)	(0.002)	(0.003)
scene	**0.7978**	0.7883	0.7766	0.7659	0.7895	0.7895	0.7619	0.7895	0.7633	0.7645
	(0.013)	(0.010)	(0.004)	(0.005)	(0.007)	(0.007)	(0.002)	(0.007)	(0.003)	(0.001)
slashdot	**0.4443**	0.4194	0.3862	0.3820	0.3967	0.3975	0.3792	0.3975	0.3555	0.3793
	(0.012)	(0.019)	(0.008)	(0.007)	(0.008)	(0.008)	(0.008)	(0.008)	(0.010)	(0.008)
stackex	0.2162	**0.2873**	0.2239	0.2267	0.2268	0.2274	0.2259	0.2274	0.2255	0.1967
	(0.024)	**(0.014)**	(0.011)	(0.011)	(0.013)	(0.013)	(0.012)	(0.013)	(0.011)	(0.009)
tmc2007	0.5865	**0.6048**	0.5690	0.5688	0.5737	0.5632	0.5680	0.5632	0.5622	0.5633
	(0.007)	**(0.012)**	(0.011)	(0.012)	(0.014)	(0.013)	(0.011)	(0.013)	(0.011)	(0.013)
yeast	**0.4504**	0.4258	0.3835	0.3732	0.3666	0.3666	0.3984	0.3597	0.3887	0.3902
	(0.014)	(0.009)	(0.006)	(0.010)	(0.006)	(0.006)	(0.009)	(0.005)	(0.004)	(0.004)
yelp8	0.5719	**0.7002**	0.6768	0.6583	0.6342	0.6204	0.6536	0.6234	0.6487	0.6530
	(0.015)	**(0.005)**	(0.003)	(0.004)	(0.003)	(0.003)	(0.005)	(0.002)	(0.003)	(0.004)

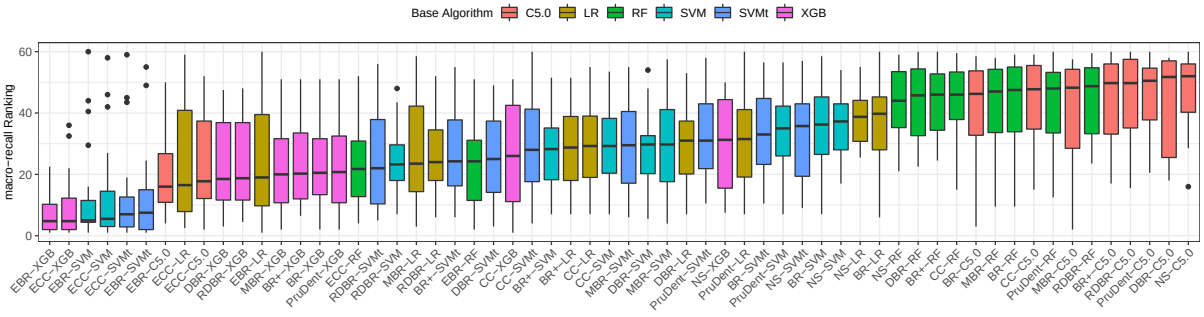


Figure 30 – Strategy/base-algorithm's rankings for the *macro-recall* measure.

Table 44 – Results of best strategies for *one-error* ↓ measure.

Data set	BR_{RF}	$PruDent_{RF}$	DBR_{RF}	MBR_{RF}	NS_{RF}	CC_{RF}	$RDBR_{RF}$	$BR+_{RF}$	EBR_{XGB}	ECC_{XGB}
20NG	0.2161 (0.003)	**0.2160 (0.002)**	0.2165 (0.002)	**0.2160 (0.002)**	0.2211 (0.002)	0.2221 (0.003)	0.2333 (0.007)	0.2297 (0.006)	0.2354 (0.003)	0.2676 (0.008)
birds	0.3270 (0.034)	0.3284 (0.026)	**0.3218 (0.031)**	0.3272 (0.031)	0.3284 (0.031)	0.3284 (0.031)	0.3230 (0.029)	0.3224 (0.030)	0.3329 (0.038)	0.3319 (0.040)
cal500	**0.1382 (0.018)**	0.1562 (0.017)	0.1418 (0.018)	0.1529 (0.021)	0.1406 (0.015)	0.1410 (0.016)	0.1422 (0.017)	0.1414 (0.017)	0.3015 (0.023)	0.2601 (0.014)
corel5k	0.6393 (0.008)	0.6379 (0.008)	0.6560 (0.007)	0.6384 (0.008)	0.6375 (0.007)	**0.6371 (0.005)**	0.6547 (0.006)	0.6553 (0.006)	0.6756 (0.012)	0.6769 (0.007)
emotions	**0.2626 (0.015)**	0.2649 (0.010)	0.2667 (0.016)	0.2659 (0.012)	0.2805 (0.014)	0.2805 (0.014)	0.2650 (0.017)	0.2667 (0.017)	0.2890 (0.016)	0.2873 (0.012)
enron	0.2128 (0.011)	**0.2117 (0.012)**	0.2121 (0.011)	0.2175 (0.010)	0.2135 (0.008)	0.2141 (0.009)	0.2199 (0.011)	0.2136 (0.010)	0.2299 (0.014)	0.2362 (0.012)
fapesp	0.3854 (0.048)	0.3894 (0.041)	0.3942 (0.052)	0.3854 (0.039)	0.3942 (0.042)	0.3942 (0.042)	0.3910 (0.052)	0.3934 (0.052)	0.3904 (0.023)	**0.3808 (0.034)**
flags	0.2009 (0.025)	**0.1979 (0.018)**	0.2009 (0.027)	0.2039 (0.029)	0.2042 (0.019)	0.2042 (0.019)	0.2071 (0.026)	0.2103 (0.018)	0.2421 (0.035)	0.2154 (0.018)
image	**0.2457 (0.007)**	0.2460 (0.009)	0.2471 (0.008)	0.2478 (0.007)	0.2469 (0.007)	0.2469 (0.007)	0.2480 (0.006)	0.2473 (0.007)	0.2525 (0.012)	0.2581 (0.010)
langlog	0.6714 (0.010)	0.6715 (0.015)	0.6779 (0.013)	0.6694 (0.013)	0.6752 (0.011)	0.6724 (0.019)	0.6775 (0.013)	0.6774 (0.013)	0.6555 (0.011)	**0.6546 (0.013)**
mediamill	**0.1015 (0.001)**	0.1057 (0.001)	0.1175 (0.001)	0.1032 (0.001)	0.1058 (0.001)	0.1059 (0.002)	0.1093 (0.003)	0.1110 (0.001)	0.1416 (0.001)	0.1605 (0.003)
medical	0.1856 (0.015)	0.1791 (0.011)	0.1909 (0.015)	0.1831 (0.012)	0.1824 (0.015)	0.1786 (0.012)	0.1877 (0.009)	0.1915 (0.013)	**0.1337 (0.011)**	0.1362 (0.012)
msd-195	0.6241 (0.011)	0.6258 (0.011)	0.6466 (0.009)	0.6258 (0.009)	0.6220 (0.009)	0.6220 (0.009)	0.6331 (0.014)	0.6361 (0.013)	**0.6043 (0.016)**	0.6127 (0.014)
ohsumed	0.3402 (0.002)	0.3397 (0.003)	0.3412 (0.003)	0.3414 (0.003)	0.3429 (0.003)	0.3422 (0.002)	0.3415 (0.002)	0.3432 (0.002)	**0.3233 (0.004)**	0.3234 (0.002)
scene	0.1903 (0.006)	0.1897 (0.006)	0.1911 (0.006)	0.1902 (0.007)	0.1924 (0.000)	0.1919 (0.000)	0.1928 (0.002)	0.1915 (0.000)	**0.1861 (0.000)**	0.1982 (0.009)
slashdot	0.3886 (0.008)	0.3887 (0.007)	**0.3873 (0.005)**	0.3885 (0.005)	0.3892 (0.008)	0.3917 (0.006)	0.4026 (0.008)	0.4045 (0.008)	0.4120 (0.008)	0.4277 (0.006)
stackex	0.4684 (0.012)	0.4657 (0.010)	0.4718 (0.008)	0.4668 (0.012)	0.4676 (0.009)	0.4718 (0.011)	0.4707 (0.009)	0.4737 (0.009)	**0.4430 (0.016)**	0.4445 (0.011)
tmc2007	**0.1482 (0.001)**	0.1489 (0.001)	0.1521 (0.001)	0.1695 (0.003)	0.1503 (0.001)	0.1512 (0.002)	0.1527 (0.002)	0.1530 (0.001)	0.1802 (0.003)	0.1763 (0.002)
yeast	0.2304 (0.005)	0.2489 (0.003)	0.2459 (0.003)	0.2467 (0.003)	0.2525 (0.003)	0.2525 (0.003)	0.2434 (0.004)	0.2455 (0.004)	**0.2249 (0.007)**	0.2307 (0.006)
yelp8	0.1659 (0.005)	0.1695 (0.004)	0.1707 (0.004)	0.1703 (0.003)	0.1622 (0.004)	0.1622 (0.005)	0.1699 (0.004)	0.1698 (0.004)	0.1616 (0.004)	**0.1588 (0.005)**

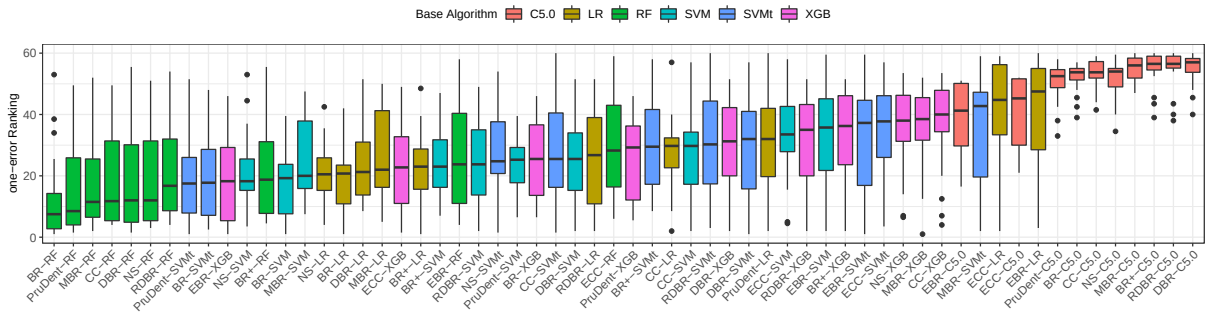


Figure 31 – Strategy/base-algorithm's rankings for the *one-error* measure.

Table 45 – Results of best strategies for *ranking-loss* ↓ measure.

Data set	DBR_{RF}	$PruDent_{RF}$	BR_{RF}	$BR+_{RF}$	CC_{RF}	NS_{RF}	MBR_{RF}	$RDBR_{RF}$	EBR_{XGB}	ECC_{XGB}
20NG	0.0378 (0.001)	0.0377 (0.001)	**0.0376 (0.001)**	0.0408 (0.001)	0.0379 (0.001)	0.0379 (0.001)	0.0389 (0.000)	0.0407 (0.001)	0.0745 (0.001)	0.0882 (0.005)
birds	0.1205 (0.011)	0.1212 (0.011)	0.1226 (0.011)	**0.1204 (0.011)**	0.1234 (0.010)	0.1234 (0.010)	0.1217 (0.012)	**0.1204 (0.010)**	0.1909 (0.009)	0.1875 (0.010)
cal500	0.2185 (0.003)	0.2287 (0.003)	**0.2180 (0.003)**	0.2187 (0.003)	0.2204 (0.003)	0.2202 (0.003)	0.2258 (0.003)	0.2210 (0.002)	0.2589 (0.002)	0.2548 (0.003)
corel5k	0.1521 (0.001)	0.1534 (0.001)	0.1532 (0.001)	**0.1514 (0.001)**	0.1516 (0.001)	0.1519 (0.001)	0.1612 (0.001)	0.1535 (0.001)	0.2389 (0.001)	0.2377 (0.002)
emotions	**0.1455 (0.006)**	0.1460 (0.006)	0.1475 (0.008)	0.1465 (0.006)	0.1490 (0.007)	0.1490 (0.007)	0.1476 (0.006)	0.1468 (0.007)	0.1689 (0.012)	0.1664 (0.009)
enron	**0.0832 (0.001)**	**0.0832 (0.001)**	0.0835 (0.001)	0.0834 (0.001)	0.0834 (0.002)	0.0833 (0.001)	**0.0832 (0.001)**	0.0834 (0.001)	0.1430 (0.003)	0.1368 (0.004)
fapesp	0.1002 (0.009)	0.1022 (0.010)	0.1028 (0.008)	**0.1000 (0.009)**	0.1026 (0.009)	0.1026 (0.009)	0.1020 (0.009)	0.1001 (0.009)	0.1462 (0.016)	0.1350 (0.013)
flags	0.1903 (0.014)	0.1876 (0.013)	0.1831 (0.013)	0.1899 (0.014)	0.1826 (0.013)	**0.1825 (0.014)**	0.1890 (0.018)	0.1900 (0.014)	0.2056 (0.015)	0.1980 (0.012)
image	**0.1313 (0.004)**	0.1323 (0.007)	0.1321 (0.006)	**0.1313 (0.004)**	0.1323 (0.005)	0.1323 (0.005)	0.1328 (0.005)	0.1315 (0.004)	0.1532 (0.007)	0.1565 (0.007)
langlog	0.1688 (0.003)	0.1681 (0.003)	0.1684 (0.003)	0.1687 (0.003)	0.1696 (0.004)	0.1695 (0.005)	**0.1664 (0.005)**	0.1685 (0.003)	0.2500 (0.008)	0.2530 (0.007)
mediamill	0.0417 (0.000)	0.0377 (0.000)	**0.0338 (0.000)**	0.0418 (0.000)	0.0400 (0.000)	0.0400 (0.000)	0.0545 (0.001)	0.0435 (0.001)	0.1634 (0.001)	0.1584 (0.002)
medical	0.0224 (0.003)	0.0227 (0.003)	0.0228 (0.003)	0.0228 (0.003)	0.0219 (0.003)	0.0218 (0.003)	**0.0217 (0.003)**	0.0222 (0.003)	0.0406 (0.007)	0.0413 (0.007)
msd-195	0.1499 (0.002)	0.1494 (0.002)	0.1520 (0.002)	0.1490 (0.002)	0.1487 (0.002)	0.1487 (0.002)	0.1522 (0.002)	**0.1469 (0.002)**	0.2695 (0.006)	0.2628 (0.005)
ohsumed	0.0849 (0.001)	0.0849 (0.001)	0.0851 (0.001)	0.0853 (0.001)	0.0853 (0.001)	0.0853 (0.001)	**0.0845 (0.001)**	0.0853 (0.001)	0.1741 (0.002)	0.1720 (0.002)
scene	0.0593 (0.003)	0.0593 (0.002)	**0.0589 (0.003)**	0.0593 (0.001)	0.0592 (0.002)	0.0598 (0.002)	0.0593 (0.003)	0.0605 (0.001)	0.0910 (0.003)	0.0862 (0.005)
slashdot	**0.1091 (0.002)**	0.1110 (0.002)	0.1116 (0.001)	0.1134 (0.001)	0.1102 (0.002)	0.1106 (0.002)	0.1127 (0.002)	0.1123 (0.001)	0.1795 (0.005)	0.1805 (0.004)
stackex	**0.1086 (0.002)**	0.1099 (0.003)	0.1107 (0.002)	0.1092 (0.002)	0.1110 (0.003)	0.1106 (0.002)	0.1130 (0.004)	0.1105 (0.002)	0.2366 (0.012)	0.2351 (0.010)
tmc2007	0.0315 (0.000)	0.0312 (0.000)	**0.0311 (0.001)**	0.0315 (0.001)	0.0315 (0.001)	0.0314 (0.000)	0.0324 (0.001)	0.0315 (0.000)	0.0764 (0.001)	0.0723 (0.001)
yeast	0.1715 (0.002)	0.1683 (0.002)	**0.1603 (0.002)**	0.1685 (0.002)	0.1682 (0.002)	0.1682 (0.002)	0.1701 (0.002)	0.1689 (0.002)	0.1678 (0.002)	0.1668 (0.002)
yelp8	0.1178 (0.002)	0.1227 (0.002)	0.1130 (0.002)	0.1178 (0.002)	0.1177 (0.002)	0.1172 (0.002)	0.1122 (0.002)	0.1217 (0.003)	0.1187 (0.002)	**0.1088 (0.002)**

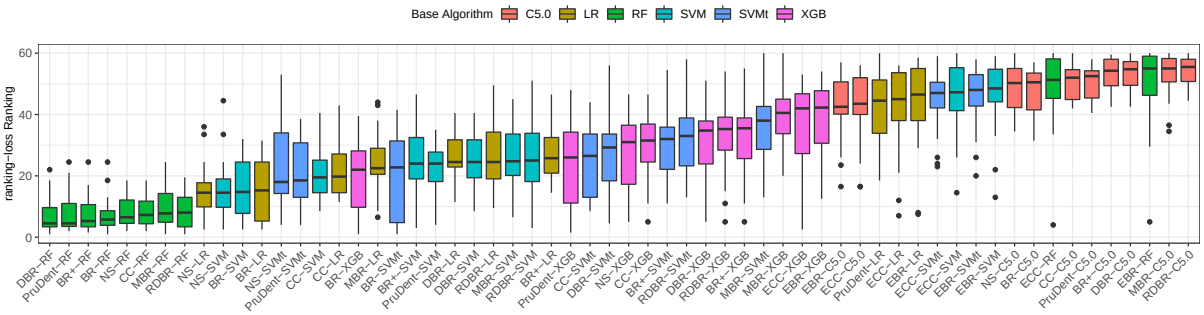


Figure 32 – Strategy/base-algorithm's rankings for the *ranking-loss* measure.

Table 46 – Results of best strategies for *subset-accuracy* ↑ measure.

Data set	$RDBR_{RF}$	$BR+_{SVMt}$	NS_{SVMt}	DBR_{RF}	CC_{SVMt}	$PruDent_{RF}$	MBR_{RF}	BR_{RF}	ECC_{XGB}	EBR_{XGB}
20NG	0.7476 (0.005)	0.6963 (0.008)	0.7164 (0.006)	**0.7630 (0.003)**	0.6945 (0.009)	0.7627 (0.002)	0.7628 (0.002)	0.7628 (0.003)	0.6543 (0.017)	0.7180 (0.003)
birds	**0.2852 (0.026)**	0.1758 (0.067)	0.1300 (0.052)	**0.2852 (0.026)**	0.1766 (0.064)	0.2804 (0.022)	0.2840 (0.026)	0.2830 (0.022)	0.2195 (0.028)	0.2196 (0.043)
cal500	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**	**0.0000 (0.000)**
corel5k	0.0045 (0.001)	0.0231 (0.002)	**0.0260 (0.004)**	0.0040 (0.001)	0.0249 (0.002)	0.0038 (0.001)	0.0044 (0.001)	0.0041 (0.001)	0.0089 (0.002)	0.0082 (0.001)
emotions	0.3253 (0.041)	**0.3346 (0.023)**	0.3045 (0.024)	0.3220 (0.038)	0.3139 (0.029)	0.3160 (0.035)	0.3079 (0.037)	0.3116 (0.037)	0.3083 (0.020)	0.2871 (0.018)
enron	0.1523 (0.016)	0.1519 (0.013)	0.1558 (0.012)	0.1495 (0.015)	0.1456 (0.012)	0.1407 (0.016)	0.1404 (0.015)	0.1388 (0.014)	**0.1563 (0.012)**	0.1465 (0.016)
fapesp	0.3993 (0.025)	0.4176 (0.035)	**0.4351 (0.030)**	0.3960 (0.024)	0.4310 (0.021)	0.3969 (0.016)	0.4049 (0.016)	0.3977 (0.016)	0.3403 (0.026)	0.3481 (0.028)
flags	**0.2506 (0.022)**	0.2207 (0.023)	0.1577 (0.049)	0.2381 (0.027)	0.1608 (0.081)	0.2144 (0.029)	0.2196 (0.026)	0.2113 (0.031)	0.2146 (0.037)	0.1835 (0.035)
image	0.5566 (0.006)	0.5485 (0.016)	0.5445 (0.016)	**0.5572 (0.009)**	0.5458 (0.015)	0.5564 (0.008)	0.5543 (0.005)	0.5552 (0.008)	0.5148 (0.012)	0.5084 (0.011)
langlog	0.2187 (0.012)	0.2130 (0.010)	0.2115 (0.011)	0.2179 (0.012)	0.2137 (0.013)	0.2199 (0.016)	**0.2240 (0.013)**	0.2222 (0.010)	0.1851 (0.005)	0.1863 (0.011)
mediamill	0.1702 (0.002)	**0.2261 (0.003)**	0.2220 (0.003)	0.1622 (0.002)	0.2212 (0.003)	0.1513 (0.001)	0.1458 (0.002)	0.1460 (0.002)	0.1483 (0.002)	0.0991 (0.002)
medical	0.6669 (0.012)	0.7449 (0.026)	0.7549 (0.017)	0.6644 (0.015)	0.7445 (0.021)	0.6749 (0.015)	0.6672 (0.015)	0.6669 (0.013)	**0.7561 (0.013)**	0.7540 (0.011)
msd-195	0.1514 (0.008)	**0.1588 (0.009)**	0.1165 (0.051)	0.1325 (0.008)	0.1542 (0.008)	0.1373 (0.009)	0.1264 (0.009)	0.1286 (0.009)	0.0693 (0.006)	0.0619 (0.005)
ohsumed	0.2997 (0.004)	0.3086 (0.003)	0.3074 (0.004)	0.2994 (0.004)	**0.3089 (0.003)**	0.2988 (0.004)	0.2990 (0.005)	0.2980 (0.004)	0.2775 (0.004)	0.2753 (0.005)
scene	0.7453 (0.001)	0.7319 (0.006)	0.7387 (0.008)	**0.7466 (0.004)**	0.7333 (0.006)	0.7454 (0.005)	0.7449 (0.005)	0.7428 (0.004)	0.7316 (0.012)	0.7266 (0.003)
slashdot	0.4813 (0.009)	0.3141 (0.191)	0.3468 (0.169)	**0.4943 (0.007)**	0.3068 (0.168)	0.4901 (0.008)	0.4903 (0.006)	0.4892 (0.008)	0.4059 (0.014)	0.4345 (0.006)
stackex	0.1324 (0.007)	0.1204 (0.019)	**0.1356 (0.012)**	0.1296 (0.007)	0.1161 (0.019)	0.1349 (0.007)	0.1336 (0.007)	0.1341 (0.007)	0.1209 (0.009)	0.1175 (0.010)
tmc2007	**0.4486 (0.003)**	0.3722 (0.038)	0.4036 (0.031)	0.4456 (0.004)	0.3992 (0.024)	0.4373 (0.004)	0.4394 (0.004)	0.4373 (0.003)	0.3782 (0.002)	0.3649 (0.004)
yeast	0.2092 (0.008)	0.2388 (0.008)	**0.2539 (0.009)**	0.1765 (0.005)	0.2264 (0.010)	0.1822 (0.010)	0.1623 (0.007)	0.1638 (0.008)	0.2295 (0.007)	0.1870 (0.008)
yelp8	0.4065 (0.004)	0.4231 (0.004)	**0.4260 (0.006)**	0.4049 (0.005)	0.4235 (0.007)	0.3756 (0.005)	0.3746 (0.003)	0.3742 (0.004)	0.4174 (0.007)	0.3752 (0.011)

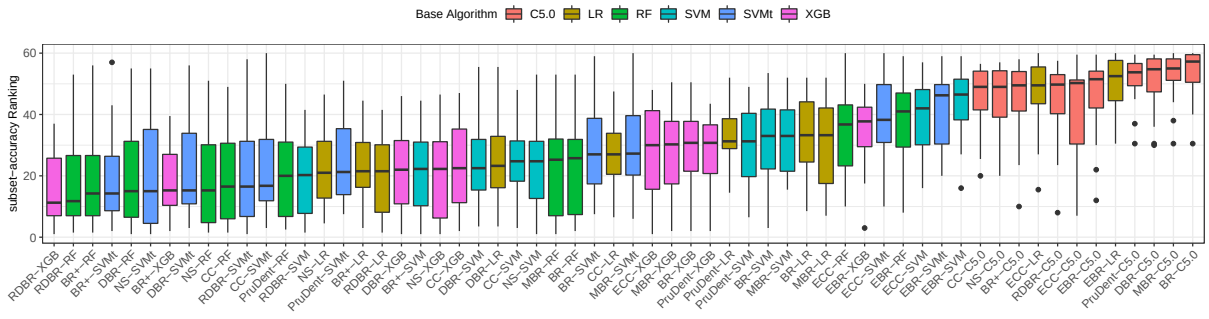


Figure 33 – Strategy/base-algorithm's rankings for the *subset-accuracy* measure.

APPENDIX

B

BAYESIAN STATISTICAL RESULTS OF THE COMPARISON BETWEEN STRATEGIES AND BASE ALGORITHMS

From the previous results, the best pairs of strategies/base-algorithms were statistically compared against the other pairs using the Bayesian statistical test. Tables 47 to 54 report the pairs that the considered strategies/base-algorithms statistically outperform with a probability greater than or equal to 95%.

Table 47 – Bayesian Statistical results for the *F1* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
EBR_{XGB}	*	*	*	*	124	12	*	*	*	*
ECC_{XGB}	*	*	*	*	124	12	*	*	*	*
$PruDent_{RF}$	1	1	1	1	1	1	1	1		
MBR_{LR}	1	1	1	1	1	1	1	1		
$RDBR_{SVMt}$	13	13	136	13	1	1	13	16	13	13
DBR_{SVMt}	1	1	136	1	1	1	1	13		
BR_{RF}	1	1	1	1	1	1	1	1		
NS_{RF}	1	1	1	1	1	1	1	1		
$BR+_{SVM}$	1	1	1	1	1	1	1	1		
CC_{RF}	1	1	1	1	1	1	1	1		

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 48 – Bayesian Statistical results for the *hamming-loss* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
$PruDent_{RF}$	16	16	16	156	*	*	16	16	16	16
BR_{RF}	16	16	16	16	*	*	16	16	16	16
MBR_{RF}	16	16	16	156	*	*	16	16	16	16
$BR+_{RF}$	1	16	16	16	*	*	1	16	1	16
DBR_{RF}	1	16	16	16	*	*	1	16	1	16
$RDBR_{RF}$	1	16	16	16	*	*	1	16	1	16
CC_{RF}	16	16	16	156	*	*	16	16	16	16
NS_{RF}	1	16	16	16	*	*	1	16	1	1
EBR_{RF}	1	1	1	1	124	12	1	1	1	1
ECC_{RF}	1	1	1	1	12	12	1	1	1	1

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 49 – Bayesian Statistical results for the *macro-F1* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
EBR_{XGB}	*	*	*	*	123	123	*	*	*	*
ECC_{XGB}	*	12346	12346	12346	123	123	*	*	*	12346
MBR_{XGB}	1234	13	123	13	13	13	13	1234	1234	13
BR_{XGB}	1234	13	1234	13	13	1	13	1234	1234	13
$PruDent_{XGB}$	1234	13	1234	123	13	1	13	1234	1234	13
$RDBR_{SVMt}$	1234	13	13	13	12	1	1234	1234	1234	13
DBR_{SVMt}	1234	13	13	13	13	1234	134	13		
CC_{XGB}	123	13	13	13	1	1	13	123	13	13
NS_{XGB}	1234	13	13	13	1	1	13	123	134	13
$BR+_{SVMt}$	1234	13	13	13	13	1234	1234	13		

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 50 – Bayesian Statistical results for the *macro-precision* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
MBR_{RF}	1	1	1	12	*	12345	1	12	14	1
CC_{RF}	1	1	1	1	12345	12345	1	1	1	1
$RDBR_{RF}$	12	124	12	124	*	12345	1	124	14	1
NS_{RF}	1	1	1	1	*	*	1	12	1	1
$BR+_{RF}$	12	1	12	12	*	12345	1	124	14	1
$PruDent_{RF}$	12	1	1	12	*	12345	1	12	14	1
DBR_{RF}	12	1	1	12	*	12345	1	12	14	1
BR_{RF}	1	1	1	1	*	12345	1	1	1	1
ECC_{XGB}	1	1	1	1	1245	124	1	1	1	1
EBR_{XGB}	1	1	1	1	124	124	1	1	1	1

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 51 – Bayesian Statistical results for the *macro-recall* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
EBR_{SVM}	*	*	*	*	123	123	*	*	*	*
ECC_{XGB}	*	*	*	*	123	123	*	*	*	*
DBR_{XGB}	1234	13	13	13	13	123	134	13		
$RDBR_{XGB}$	1234	13	13	13	13	123	134	13		
MBR_{XGB}	1234	13	13	13	13	1234	134	13		
BR_{XGB}	1234	13	13	13	13	1234	134	13		
$BR+_{XGB}$	1234	13	13	13	13	1234	134	13		
$PruDent_{XGB}$	1234	13	134	13	13	1234	1234	13		
CC_{XGB}	13	13	13	13	13	13	13	13		
NS_{XGB}	13	13	13	13	13	13	13	13		

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 52 – Bayesian Statistical results for the *one-error* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
BR_{RF}	1	1246	12456	1246	1245	1246	12456	1246	124	12456
$PruDent_{RF}$	1	1246	1246	1246	1245	124	12456	16	124	1246
DBR_{RF}	1	1246	1246	1246	124	124	146	16	12	124
MBR_{RF}	1	1246	1246	1246	124	124	1456	16	12	1246
NS_{RF}	1	1246	1246	1246	124	124	1456	16	12	1246
CC_{RF}	1	1246	1246	1246	124	124	1456	16	12	1246
$RDBR_{RF}$	1	1	146	1	12	12	16	16	12	1
$BR+_{RF}$	1	1	16	1	12	12	16	16	12	1
EBR_{XGB}	1	1	16	1	124	12	16	1	12	1
ECC_{XGB}	1	1	16	1	12	12	1	1	1	1

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 53 – Bayesian Statistical results for the *ranking-loss* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
DBR_{RF}	1	126	16	16	*	*	156	16	126	126
$PruDent_{RF}$	1	1246	16	126	*	*	156	16	126	1246
BR_{RF}	1	1246	16	1246	*	*	156	16	126	1246
$BR+_{RF}$	1	16	16	16	*	*	156	16	126	126
CC_{RF}	1	126	16	16	*	*	156	16	126	126
NS_{RF}	1	126	16	16	*	*	156	16	126	126
MBR_{RF}	1	16	16	16	*	*	156	16	12	16
$RDBR_{RF}$	1	126	16	16	*	*	156	16	126	126
EBR_{XGB}	1	1	1	1	34	3	1	1	1	1
ECC_{XGB}	1	1	1	1	34	34	1	1	1	1

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

Table 54 – Bayesian Statistical results for the *subset-accuracy* measure such that the strategies in the row improve the strategies in the columns with a probability greater than or equal to 95%. The cells' content indicates the base algorithms from the columns.

Strategy	BR	BR+	CC	DBR	EBR	ECC	MBR	NS	PruDent	RDBR
$RDBR_{RF}$	16	1	1	1	*	*	16	1	1	1
$BR+_{SVMt}$	16	1	1	1	*	*	1	1	1	1
NS_{SVMt}	1	1	1	1	*	*	1	1	1	1
DBR_{RF}	16	1	1	1	*	*	1	1	1	1
CC_{SVMt}	16	1	1	1	*	*	1	1	1	1
$PruDent_{RF}$	1	1	1	1	*	12345	1	1	1	1
MBR_{RF}	1	1	1	1	*	12345	1	1	1	1
BR_{RF}	1	1	1	1	*	12345	1	1	1	1
ECC_{XGB}	1	1	1	1	1245	1245	1	1	1	1
EBR_{XGB}	1	1	1	1	12	12	1	1	1	1

*Symbols, 1: C5.0; 2: LR; 3: RF; 4: SVM; 5: SVMt; 6: XGB; Empty: none; *: All*

APPENDIX

C

CHARACTERIZATION MEASURES FORMALIZATION

C.0.1 Simple

attrToInst Ratio of the number of attributes per the number of instances (KALOUSIS; THEOHARIS, 1999), also known as dimensionality: $\frac{d}{n}$.

catToNum Ratio of the number of categorical attributes per the number of numeric attributes (FEURER; SPRINGENBERG; HUTTER, 2014): $\frac{nrCat_{\mathbf{X}}}{nrNum_{\mathbf{X}}}$.

classToAttr Ratio of the number of classes per the number of attributes (TODOROVSKI; BRAZDIL; SOARES, 2000): $\frac{q}{d}$

instToAttr Ratio of the number of instances per the number of attributes (KUBA *et al.*, 2002): $\frac{n}{d}$.

instToClass Ratio of the number of instances per the number of classes (VANSCHOREN, 2010): $\frac{n}{q}$.

ntAttr Number of attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): d.

nrAttrMissing Number of attributes with missing values (FEURER; SPRINGENBERG; HUTTER, 2014):

$$\sum_{j=1}^{d} \mathbb{1}(\sum_{i=1}^{n} \mathbb{1}(\vec{x}_{ij} = \varnothing) > 0)$$

nrBin Number of binary attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $\sum_{i=1}^{d} \mathbb{1}(\phi_{\vec{a}_i} = 2)$. It includes numerical and categorical attributes that contain only two distinct values.

nrCat Number of categorical attributes (ENGELS; THEUSINGER, 1998): $d - nrNum_{\mathbf{X}}$.

nrClass Number of classes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): q.

nrInst Number of instances (MICHIE; SPIEGELHALTER; TAYLOR, 1994): n.

nrInstMissing Number of instances with missing values (LINDNER; STUDER, 1999):

$$\sum_{i=1}^{n} \mathbb{1}(\sum_{j=1}^{d} \mathbb{1}(\vec{x}_{ij} = \varnothing) > 0)$$

nrMissing Number of missing values (LINDNER; STUDER, 1999):

$$\sum_{i=1}^{n} \sum_{j=1}^{d} \mathbb{1}(\vec{x}_{ij} = \varnothing)$$

nrNum Number of numeric attributes (ENGELS; THEUSINGER, 1998): $\sum_{i=1}^{d} \mathbb{1}(\vec{a}_i \in \mathbb{R}^n)$.

numToCat Ratio of the number of numeric attributes per the number of categorical attributes (FEURER; SPRINGENBERG; HUTTER, 2014): $\frac{nrNum_{\mathbf{X}}}{nrCat_{\mathbf{X}}}$.

freqClass Frequencies of the classes values (LINDNER; STUDER, 1999): $\left[prop_{c_1}, \ldots, prop_{c_q}\right]$, such that

$$prop_{c_j} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(y_i = c_j). \tag{C.1}$$

C.0.2 Statistical

canCor Canonical correlations between the predictive attributes and the class (KALOUSIS, 2002): $[\rho_1, \cdots, \rho_z]$, such that $\rho_i = cor_{\vec{w}_x^{(i)}\mathbf{X}, \vec{w}_y^{(i)}\mathbf{Y}}$, where $\vec{w}_x^{(i)}$ and $\vec{w}_y^{(i)}$ maximizes ρ_i and are orthogonal to the $\vec{w}_x^{(i-1)}$ and $\vec{w}_y^{(i-1)}$, $\mathbf{Y}$ is the binarized version of $\vec{y}$ and $z \leq min\,[q,d]$ is the number of distinct $\vec{w}_x$ and $\vec{w}_y$ vectors found by using discriminant analysis. Frequently, the canonical correlation is reported in the literature as the eigenvalues of the canonical discriminant matrix, such that

$$\rho_i = \sqrt{\frac{\lambda_i}{1+\lambda_i}}. \tag{C.2}$$

cor Absolute attributes correlation (CASTIELLO; CASTELLANO; FANELLI, 2005): $\left[\, |cor_{\vec{a}_1,\vec{a}_3}|, \cdots, |cor_{\vec{a}_{d-1},\vec{a}_d}| \,\right]$, such that $cor_{x,y}$ is obtained by the use of a correlation algorithm. The most common one used is the Pearson's Correlation coefficient, given by

$$cor_{x,y} = \frac{cov_{x,y}}{sd_x sd_y}, \text{ where} \tag{C.3}$$

$$cov_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}, \text{ and} \tag{C.4}$$

$$sd_x = \sqrt[2]{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{C.5}$$

cov Attributes covariance (CASTIELLO; CASTELLANO; FANELLI, 2005): $[\ |cov_{\vec{a}_1,\vec{a}_2}|, \cdots, |cov_{\vec{a}_{d-1},\vec{a}_d}|\ ]$, where $cov_{x,y}$ is given by Equation C.4.

nrDisc Number of discriminant functions (LINDNER; STUDER, 1999): $|canCor_{\mathscr{D}}|$.

eigenvalues Eigenvalues of the covariance matrix (ALI; SMITH, 2006): $[\lambda_1, \cdots, \lambda_d]$, such that $S\vec{v} = \lambda_i\vec{v}$ for some $\vec{v} \neq 0$, where $S_{d\times d}$ is the covariance matrix of $\mathbf{X}$.

gMean Geometric mean of attributes (ALI; SMITH-MILES, 2006): $\left[gMean_{\vec{a}_1}, \cdots, gMean_{\vec{a}_d}\right]$, such that $gMean_x = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$.

hMean Harmonic mean of attributes (ALI; SMITH-MILES, 2006): $\left[hMean_{\vec{a}_1}, \cdots, hMean_{\vec{a}_d}\right]$, such that

$$hMean_x = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}}.$$

iqRange Interquartile range of attributes (ALI; SMITH-MILES, 2006): $\left[iqRange_{\vec{a}_1}, \cdots, iqRange_{\vec{a}_d}\right]$, such that $iqRange_x = Q3_x - Q1_x$, where $Q1_x$ and $Q3_x$ represent the first and third quartile values of x, respectively.

kurtosis Kurtosis of attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $\left[kurt_{\vec{a}_1}, \cdots, kurt_{\vec{a}_d}\right]$, such that

$$kurt_x = \frac{m_4}{sd_x^4} - 3,$$

where m_j represents a statistical moment, given by

$$m_j = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^j. \tag{C.6}$$

mad Median absolute deviation of attributes (ALI; SMITH, 2006): $\left[mad_{\vec{a}_1}, \cdots, mad_{\vec{a}_d}\right]$, such that $mad_x = median\,[\ |x_1 - median_x|, \cdots, |x_n - median_x|\ ]$, where

$$median_x = \begin{cases} \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } |x| \text{ is even } (|x| = 2r) \\ x_{(r+1)} & \text{otherwise } (|x| = 2r+1) \end{cases} \tag{C.7}$$

max Maximum value of attributes (ENGELS; THEUSINGER, 1998): $[max\ \vec{a}_1, \cdots, max\ \vec{a}_d]$.

mean Mean value of attributes (ENGELS; THEUSINGER, 1998): $[\ \overline{\vec{a}_1}, \cdots, \overline{\vec{a}_d}\ ]$.

median Median value of attributes (ENGELS; THEUSINGER, 1998): $\left[median_{\vec{a}_1}, \cdots, median_{\vec{a}_d}\right]$, where $median_x$ is given by Equation C.7.

min Minimum value of attributes (ENGELS; THEUSINGER, 1998): $[min\ \vec{a}_1, \cdots, min\ \vec{a}_d]$.

nrCorAttr Number of attributes pairs with high correlation (SALAMA; HASSANIEN; REVETT, 2013):

$$\frac{2}{d(d-1)} \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \mathbb{1}(|cor_{\vec{a}_i,\vec{a}_j}| \geq \tau),$$

where τ is a threshold value between 0 and 1, usually $\tau = 0.5$. This is the normalized version adapted by the authors.

nrNorm Number of attributes with normal distribution (KOPF; TAYLOR; KELLER, 2000): $\sum_{i=1}^{d} \mathbb{1}(isNormal_{\vec{a}_i})$. To check if an attribute has or does not have a normal distribution the W-Test for normality (ROYSTON, 1995) can be applied, for instance.

nrOutliers Number of attributes with outliers values (KOPF; IGLEZAKIS, 2002): $\sum_{i=1}^{d} \mathbb{1}(hasOutlier_{\vec{a}_j})$. To test if an attribute has or does not have outliers, the Tukey's boxplot algorithm (ROUSSEEUW; HUBERT, 2011) can be used, for instance.

range Range of Attributes (ALI; SMITH-MILES, 2006): $[(max\ \vec{a}_1 - min\ \vec{a}_1), \cdots, (max\ \vec{a}_d - min\ \vec{a}_d)]$.

sd Standard deviation of the attributes (ENGELS; THEUSINGER, 1998): $\left[sd_{\vec{a}_1}, \cdots, var_{\vec{a}_d}\right]$, such that sd_x is given by Equation C.5.

sdRatio Statistic test for homogeneity of covariances (MICHIE; SPIEGELHALTER; TAYLOR, 1994):

$$exp(M/d \sum_{i=1}^{q}(n_{c_i} - 1)), \text{ where } M = \gamma \sum_{i=1}^{q}(n_{c_i} - 1) log|S_i^{-1}S|;$$

$$\gamma = 1 - \frac{2d^2 + 3d - 1}{6(d+1)(q-1)} \sum_{i=1}^{q} \frac{1}{n_{c_i} - 1} - \frac{1}{n-q};$$

$$S = \frac{1}{n-q} \sum_{i=1}^{q}(n_{c_i} - 1)S_i$$

such that, n_{c_i} is the number of instances related to the class c_i, S is called pooled covariance matrix and S_i is the sample covariance matrix of the instances for the i^{th} class.

skewness Skewness of attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $\left[skewness_{\vec{a}_1}, \cdots, skewness_{\vec{a}_d}\right]$, such that

$$skewness_x = \frac{m_3}{sd_x^3},$$

where sd_x and m_3 are given by Equation C.5 and C.6, respectively.

tMean Trimmed mean of attributes (ENGELS; THEUSINGER, 1998): $\left[tMean_{\vec{a}_1}, \cdots, tMean_{\vec{a}_d}\right]$, such that

$$tMean_x = \frac{x_{(i+1)} + x_{(i+2)} + \cdots + x_{(n-i-2)} + x_{(n-i-1)}}{n-2i},$$

where $i = \lceil n\alpha \rceil$ and α is a hyperparameter, such that $0 < \alpha < 0.5$. The suggested value is $\alpha = 0.2$.

var Attributes variance (CASTIELLO; CASTELLANO; FANELLI, 2005): $\left[var_{\vec{a}_1}, \cdots, var_{\vec{a}_d}\right]$, such that

$$var_x = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}.$$

wLambda Wilks Lambda (LINDNER; STUDER, 1999):

$$\prod_{i=1}^{z} \frac{1}{1+\lambda_i},$$

where $z = nrDisc_{\mathscr{D}}$ and λ_i is defined in Equation C.2.

C.0.3 Information-Theoretic

Let H_x be the entropy of a given attribute, such that

$$H_x = -\sum_{i=1}^{\phi_x} P(x = \varphi_i^x) \log_2 P(x = \varphi_i^x),$$

and let $H_{x,y}$ be the joint entropy of a predictive attribute x and the class y, such that

$$H_{x,y} = \sum_{i=1}^{\phi_x} \sum_{j=1}^{\phi_y} \pi_{ij} \log_2 \pi_{ij},$$

where $\pi_{ij} = P(x = \varphi_i^x, y = \varphi_j^y)$. The mutual information shared between them is given by $MI_{x,y} = H_x + H_y - H_{x,y}$. Mainly from these concepts, the information-theoretic measures are computed as following:

attrEnt Attributes entropy (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $\left[H_{\vec{a}_1}, \cdots, H_{\vec{a}_d}\right]$.

classEnt Class entropy (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $H_{\vec{y}}$

eqNumAttr Equivalent number of attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994):

$$\frac{H_{\vec{y}}}{\frac{1}{d}\sum_{i=1}^{d} MI_{\vec{a}_i,\vec{y}}}$$

jointEnt Joint Entropy of attributes and classes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $\left[H_{\vec{a}_1,\vec{y}}, \cdots, H_{\vec{a}_d,\vec{y}}\right]$.

mutInf Mutual information of attributes and classes (MICHIE; SPIEGELHALTER; TAYLOR, 1994): $\left[MI_{\vec{a}_1,\vec{y}}, \cdots, MI_{\vec{a}_d,\vec{y}}\right]$.

nsRatio Noisiness of attributes (MICHIE; SPIEGELHALTER; TAYLOR, 1994) :

$$\frac{\frac{1}{d}\sum_{j=1}^{d} H_{\vec{a}_j} - \frac{1}{d}\sum_{j=1}^{d} MI_{\vec{a}_j,\vec{y}}}{\frac{1}{d}\sum_{j=1}^{d} MI_{\vec{a}_j,\vec{y}}}.$$

C.0.4 Model-Based

For DT-model meta-features, let ψ be the set of leaves, η be the set of nodes, such that $\psi \cap \eta = \emptyset$ and $\Gamma = \psi \cup \eta$ are the whole structure of the tree that represents the DT learning model. In addition, consider the following tree properties:

$attr_{\eta_i}$ Predictive attribute used in the node η_i.

$class_{\psi_i}$ Class predicted by the leaf ψ_i.

$inst_{\Gamma_i}$ Number of training instances used to define the tree element Γ_i.

$level_{\Gamma_i}$ Level of the tree element Γ_i. In other words, it is the number of nodes in the tree hierarchy necessary to reach the root of the tree, such that $level_{\Gamma_i} = 0$ iff $\Gamma_i = root_\Gamma$.

$prob_{\psi_i}$ Probability of reaching the leaf ψ_i from the root in a random walk through the tree hierarchy, such that $prob_{\psi_i} = \frac{1}{2^{level_{\psi_i}}}$.

$root_\Gamma$ Root node of a tree, such that $root_\Gamma \in \eta$.

The DT-model meta-features are the following:

leaves Number of leaves (PENG *et al.*, 2002a): $|\psi|$.

leavesBranch Size of branches (PENG *et al.*, 2002a): $\left[level_{\psi_1}, \cdots, level_{\psi_z})\right]$, where $z = |\psi|$.

leavesCorrob Leaves corroboration (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\left[\frac{inst_{\psi_1}}{n}, \cdots, \frac{inst_{\psi_z}}{n}\right]$, where $z = |\psi|$.

leavesHomo Homogeneity (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\left[\frac{z}{shape_{\psi_1}}, \cdots, \frac{z}{shape_{\psi_z}}\right]$, where $z = |\psi|$.

leavesPerClass Leaves per class (FILCHENKOV; PENDRYAK, 2015): $\left[lpc_{c_1}, \cdots, lpc_{c_q}\right]$, such that

$$lpc_{c_j} = \frac{1}{|\psi|}\sum_{i=1}^{|\psi|} \mathbb{1}(class_{\psi_i} = c_j)$$

nodes Number of nodes (PENG *et al.*, 2002a): $|\eta|$.

nodesPerAttr Ratio of the number of nodes per the number of attributes (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\frac{|\eta|}{d}$.

nodesPerInst Ratio of the number of nodes per the number of instances (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\frac{|\eta|}{n}$.

nodesPerLevel Number of nodes per level (PENG *et al.*, 2002a): $\left[npl_1, \cdots, npl_{level_w}\right]$, such that

$$w = \arg\max_{\eta_i \in \eta} level_{\eta_i}, \text{and}$$

$$npl_j = \sum_{i=1}^{|\eta|} \mathbb{1}(level_{\eta_i} = j).$$

nodesRepeated Repeated nodes (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $[nrp_1, \cdots, nrp_d]\ \forall\ nrp_j > 0$, such that

$$nrp_j = \sum_{i=1}^{|\eta|} \mathbb{1}(attr_{\eta_i} = j).$$

treeDepth Tree depth (PENG *et al.*, 2002a): $[level_{\Gamma_1}, \cdots, level_{\Gamma_w}]$, where $w = |\Gamma|$.

treeImbalance Tree imbalance (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\left[imb_{\psi_1}, \cdots, imb_{\psi_w}\right]$, such that $w = |\psi|$ and $imb_{\psi_j} = -z_{\psi_j}(log_2\ z_{\psi_j})$, where $sz_{\psi_j} = prob_{\psi_j} \sum_{i=1}^{w} \mathbb{1}(prob_{\psi_i} = prob_{\psi_j})$.

treeShape Tree shape (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\left[shape_{\psi_1}, \cdots, shape_{\psi_w}\right]$, such that $shape_{\psi_j} = -prob_{\psi_j}(log_2\ prob_{\psi_j})$ and $w = |\psi|$.

varImportance Variable importance (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000): $\left[imp_{\vec{a}_1,\vec{y}}, \cdots, imp_{\vec{a}_d,\vec{y}}\right]$, where $imp_{\vec{a}_j,\vec{y}}$ describes the homogeneity of the class $\vec{y}$ produced by some split of a given attribute $\vec{a}_j$. Each DT learning algorithm uses a specific procedure to define the importance of the variables.

C.0.5 Landmarking

Let $\mathscr{A}$, θ_A and ξ be, respectively, a learning algorithm, a learning model and an evaluation measure. All landmarking meta-features are computed in the same way, such as the model is induced using the learning algorithm and a train data:

$$\mathscr{A}(\mathbf{X}_{train}, \vec{y}_{train}) \rightarrow \theta_A$$

and the prediction of the learning model for a test data is evaluated using the given measure, such as

$$landmarking_A = \xi(\theta_A(\mathbf{X}_{test}), \vec{y}_{test}),$$

where *train* and *test* subset are defined for each fold.

The differences between the landmarking measures are given by the learning-algorithm family and the predictive attributes used to induce the model, as described below:

bestNode Decision Node: $\mathscr{A}_{\text{DT}}(\mathbf{X}_{train,battr}, \vec{y}_{train}) \rightarrow \theta_{\text{bestNode}}$, where $\mathbf{X}_{train,battr}$ is the content of the most informative attribute, which is defined using the *varImportance* result.

eliteNN Elite Nearest Neighbor: $\mathscr{A}_{\text{KNN}}(\mathbf{X}_{train,battrset}, \vec{y}_{train}, k=1) \rightarrow \theta_{\text{eliteNN}}$, where $\mathbf{X}_{train,battrset}$ contains only the subset of the most informative attributes for the train data. They are defined using the *varImportance* result.

linearDiscr linear Discriminant: $\mathscr{A}_{\text{LD}}(\mathbf{X}_{train}, \vec{y}_{train}) \rightarrow \theta_{\text{linearDiscr}}$.

naiveBayes Naive Bayes: $\mathscr{A}_{\text{NB}}(\mathbf{X}_{train}, \vec{y}_{train}) \rightarrow \theta_{\text{naiveBayes}}$.

oneNN One Nearest Neighbor: $\mathscr{A}_{\text{KNN}}(\mathbf{X}_{train}, \vec{y}_{train}, k=1) \rightarrow \theta_{\text{oneNN}}$.

randomNode Random node: $\mathscr{A}_{\text{DT}}(\mathbf{X}_{train,rattr}, \vec{y}_{train}) \rightarrow \theta_{\text{randomNode}}$, where $\mathbf{X}_{train,rattr}$ is the content of a random attribute.

worstNode Worst node: $\mathscr{A}_{\text{DT}}(\mathbf{X}_{train,wattr}, \vec{y}_{train}) \rightarrow \theta_{\text{worstNode}}$, where $\mathbf{X}_{train,wattr}$ is the content of the least informative attribute.

C.0.6 Others

The following subsections specify the non-traditional characterization measures, that include groups and standalone meta-features.

C.0.6.1 Clustering and distance-based

The clustering and distance-based measures use the result of a clustering algorithm and/or a distance measure. The k partitions obtained from the use of a clustering algorithm are denoted by $C_i \subset \mathscr{D}$, such that $\bar{\vec{x}}_{C_i}$ denotes the center of cluster i. Without loss of generality, $dist_{x,y}$ represents a distance between two instances $\vec{x}_i \in \mathscr{D}$, $\vec{x}_j \in \mathscr{D}$, regardless of the type of attributes they have.

AIC Akaike Information Criterion (VUKICEVIC *et al.*, 2016):

$$\sum_{i=1}^{k} \sum_{\vec{x}_j \in C_i} (\vec{x}_j - \bar{\vec{x}}_{C_i})^2 + 2dk.$$

BIC Bayesian Information CriterionVukicevic2016

$$\sum_{i=1}^{k} \sum_{\vec{x}_j \in C_i} (\vec{x}_j - \bar{\vec{x}}_{C_i})^2 + dk\ log_n.$$

compactness Quantify the compactness of the partitions (VUKICEVIC *et al.*, 2016): $[c_1, \cdots, c_k]$, such that

$$c_i = \sum_{\vec{x}_j \in C_i} dist_{\vec{x}_j, \bar{\vec{x}}_{C_i}}. \tag{C.8}$$

connectivity Amount of neighbouring instances that are not in the same partition (VUKICEVIC *et al.*, 2016):

$$\sum_{i=1}^{n} \mathbb{1}(\vec{x}_j \in C_i \wedge nn_{\vec{x}_i} \notin C_i),$$

where $nn_{\vec{x}_i}$ is the nearest neighbor of instance $\vec{x}_i$.

distInst Distance between all pairs of instances (FERRARI; CASTRO, 2015):

$$\left[dist_{\vec{x}_1,\vec{x}_2}, dist_{\vec{x}_1,\vec{x}_3}, \cdots, dist_{\vec{x}_{n-2},\vec{x}_n}, dist_{\vec{x}_{n-1},\vec{x}_n}\right] \tag{C.9}$$

distCorrInst Distance and correlations of all pairs of instances (PIMENTEL; CARVALHO, 2019): $[c', d']$, where

$$c' = \frac{c+1}{2}, \qquad c = \left[cor_{\vec{x}_1,\vec{x}_2}, cor_{\vec{x}_1,\vec{x}_3}, \cdots, cor_{\vec{x}_{n-2},\vec{x}_n}, cor_{\vec{x}_{n-1},\vec{x}_n}\right],$$
$$d' = \frac{d - \min(d)}{\max(d) - \min(d)},$$

such that $cor_{x,y}$ (Equation C.3) is used to compute the correlation between 2 instances and d is given by Equation C.9.

gravity Center of gravity (ALI; SMITH, 2006): $dist_{\bar{\vec{x}}_{C_m}, \bar{\vec{x}}_{C_n}}$, where $\bar{\vec{x}}_{C_m}$ and $\bar{\vec{x}}_{C_n}$ are the center points of the instances related to the majority and minority classes, respectively.

nrClusters Number of clusters (NASCIMENTO *et al.*, 2009): $|C| = k$.

purityRatio Ratio of the number of clusters with a given class (LER *et al.*, 2018): $\left[\frac{s_1}{k}, \cdots, \frac{s_q}{k}\right]$, where

$$s_i = \sum_{j=1}^{k} \mathbb{1}((\vec{x}_l, y_i) \in C_j)$$

silhouette Global silhouette index (VUKICEVIC *et al.*, 2016):

$$\frac{1}{k}\sum_{i=1}^{k}\left(\frac{1}{|C_i|}\sum_{\vec{x}_j \in C_i}\frac{b(\vec{x}_j) - a(\vec{x}_j)}{max(a(\vec{x}_j), b(\vec{x}_j))}\right), \text{where}$$
$$a(\vec{x}_j) = \frac{1}{|C_i| - 1}\sum_{\substack{\vec{x}_l \in C_i \\ j \neq l}} dist_{\vec{x}_j,\vec{x}_l}, \qquad b(\vec{x}_j) = \min_{i \neq i'}\left(\frac{1}{|C_{i'}|}\sum_{\vec{x}_l \in C_{i'}} dist_{\vec{x}_j,\vec{x}_l}\right).$$

sizeDist Proportion of instances present in each cluster (LER *et al.*, 2018): $\left[\frac{|C_1|}{n}, \cdots, \frac{|C_k|}{n}\right]$.

XB Xie-Beni index (VUKICEVIC *et al.*, 2016):

$$\frac{1}{n}\frac{\sum_{i=0}^{k} c_i}{\min\limits_{i<i'} \delta(C_i, C_{i'})}, \qquad \delta(C_i, C_{i'}) = \min_{\substack{\vec{x}_j \in C_i \\ \vec{x}_l \in C_{i'}}} dist_{\vec{x}_j,\vec{x}_l},$$

where c_i is given by Equation C.8.

C.0.6.2 Complexity Measures

The complexity measures are specified, well described and explained in Lorena *et al.* (2019).

C.0.6.3 Miscellaneous

attrConc Attributes concentration coefficient (KALOUSIS; HILARIO, 2001b): $\left[conc_{\vec{a}_1,\vec{a}_2}, conc_{\vec{a}_2,\vec{a}_1}, \cdots, conc_{\vec{a}_{d-1},\vec{a}_d}, conc_{\vec{a}_d,\vec{a}_{d-1}}\right]$, such that

$$conc_{x,y} = \frac{\sum_{i=1}^{\phi_x}\sum_{j=1}^{\phi_y}\frac{\pi_{ij}^2}{\pi_{i+}} - \sum_{j=1}^{\phi_y}\pi_{+j}^2}{1-\sum_{j=1}^{\phi_j}\pi_{+j}^2}, \text{where}$$
$$\pi_{ij} = P(x=\varphi_i^x, y=\varphi_j^y), \quad \pi_{i+} = \sum_{j=1}^{\phi_y}\pi_{ij} \quad \text{and} \quad \pi_{+j} = \sum_{i=1}^{\phi_x}\pi_{ij}. \tag{C.10}$$

classConc Class concentration coefficient (KALOUSIS; HILARIO, 2001b): $\left[conc_{\vec{a}_1,\vec{y}}, \cdots, conc_{\vec{a}_d,\vec{y}}\right]$, where $conc_{x,y}$ is given by Equation C.10.

cohesiveness Density of the example distribution (VILALTA; DRISSI, 2002a): $[v(\vec{x}_1), \cdots, v(\vec{x}_n)]$.

$$v(\vec{x}_i) = \frac{1}{|\mathscr{K}|}\sum_{(\vec{x}_j,y_j)\in\mathscr{K}_{\vec{x}_i}} 1 - \mathbb{1}(y_i = y_j),$$

where $\mathscr{K}_{\vec{x}_i}$ contains the k nearest neighbors of instance $\vec{x}_i$. The k is a user hyperparameter.

consistencyRatio Proportion of repeated instances that have different targets (KOPF; IGLEZAKIS, 2002):

$$\frac{1}{n}\sum_{i=2}^{n}\sum_{j=1}^{i-1}\mathbb{1}(dist_{\vec{x}_i,\vec{x}_j} = 0 \wedge y_i \neq y_j)$$

incoherenceRatio Ratio of instances that does not overlap with any other instances in a predefined number of attributes (KOPF; IGLEZAKIS, 2002):

$$\frac{1}{n}\sum_{i=2}^{n}\mathbb{1}\left(\sum_{j=1}^{i-1}\mathbb{1}(o(\vec{x}_i,\vec{x}_j) > \alpha) = 0\right) \qquad o(\vec{x}_i,\vec{x}_j) = \sum_{l=1}^{d}\mathbb{1}(v_{il} = v_{jl}),$$

where α is a user hyperparameter to set the number of similar attributes to define when two instances overlap.

infotheoTime The elapsed time to compute the information theretical meta-features (REIF; SHAFAIT; DENGEL, 2011).

landTime The elapsed time to compute the landmarkings meta-features (REIF; SHAFAIT; DENGEL, 2011).

modelTime The elapsed time to compute the model-based meta-features (REIF; SHAFAIT; DENGEL, 2011).

oneItemset Frequency of the predictive attributes after they are binarized (SONG; WANG; WANG, 2012):

$$\left[\frac{\sum_{i=1}^{n} v_{i1}}{n}, \cdots, \frac{\sum_{i=1}^{n} v_{id}}{n})\right].$$

propPCA Proportion of principal components that explain a specific variance of the dataset (FEURER; SPRINGENBERG; HUTTER, 2014):

$$\frac{|\Lambda| - \sum_{i=1}^{|\lambda|} \mathbb{1}\left(\sum_{j=1}^{i} \lambda_i > \alpha\right) + 1}{|\Lambda|},$$

where Λ is the set of all eigen values λ_i inversely ordered according to their variance and α is a user defined threshold indicating the amount of variance desired, e.g. 0.95.

sparsity Attributes sparsity (SALAMA; HASSANIEN; REVETT, 2013): $\left[sparsity_{\vec{a}_1}, \cdots, sparsity_{\vec{a}_d}\right]$, such that

$$sparsity_x = \frac{1}{n-1}\left(\frac{\sum_{i=1}^{\phi_x} N(x = \varphi_i^x)}{\phi_x} - 1\right),$$

where $N(x = \varphi_i^x)$ is the number of times that the i^{th} distinct value of x are present in the vector. This is the normalized version adapted by the authors.

statTime The elapsed time to compute the statistical meta-features (REIF; SHAFAIT; DENGEL, 2011).

twoItemset Frequency of predictive attributes' pairs after they are binarized (SONG; WANG; WANG, 2012): $[v(1,2), v(1,3), \cdots, v(d-2,d), v(d-1,d))]$

$$v(i,j) = \frac{1}{n}\sum_{l=1}^{n} \mathbb{1}(v_{li} \neq v_{lj}).$$

uniquenessRatio Proportion of repeated instances (KOPF; IGLEZAKIS, 2002):

$$\frac{1}{n}\sum_{i=2}^{n}\sum_{j=1}^{i-1} \mathbb{1}(dist_{\vec{x}_i,\vec{x}_j} = 0)$$

wgDist Weighted distance (VILALTA, 1999): $[v(\vec{x}_1), \cdots, v(\vec{x}_n)]$, where

$$v(\vec{x}_i) = \frac{\sum_{j=1, j\neq i}^{n} W(\vec{x}_i, \vec{x}_j) dist_{\vec{x}_i,\vec{x}_j}}{\sum_{j=1, j\neq i}^{n} W(\vec{x}_i, \vec{x}_j)}$$

$$W(\vec{x}_i, \vec{x}_j) = \frac{1}{2^{2d}} \qquad d = \frac{dist_{\vec{x}_i,\vec{x}_j}}{\sqrt{n - dist_{\vec{x}_i,\vec{x}_j}}}$$

APPENDIX

D

PERFORMANCE RESULTS OF LE, LR AND THRESHOLD CALIBRATION

Figures 34, 35 and 36 present the average results for the macro-AUC, macro-F1 and macro-precision evaluation measures, respectively. The strategies LE, LR, BR+T, LE+T and LR+T were obtained by the optimization of the individual labels using the a binary evaluation measure. The BR's performance is also reported since it represents the lack of operations. The text color (red to black) indicates, for each dataset, the performance ranking over all strategies and base algorithms. For each base algorithm, the best solution is highlighted with a gray background.

Additionally, Table 55 presents the probabilities from the Bayesian statistical test between the comparison of pairs of strategies. For each base algorithm, the probability values in the left column are relative to the respective left strategy in a row, as well as, the right column is to the right strategy. The rope columns indicate the likelihood that the two strategies will be similar.

Figure 34 – Macro-AUC result of BR, LE and LR. For each base algorithm, the best solution is highlighted with a gray background.

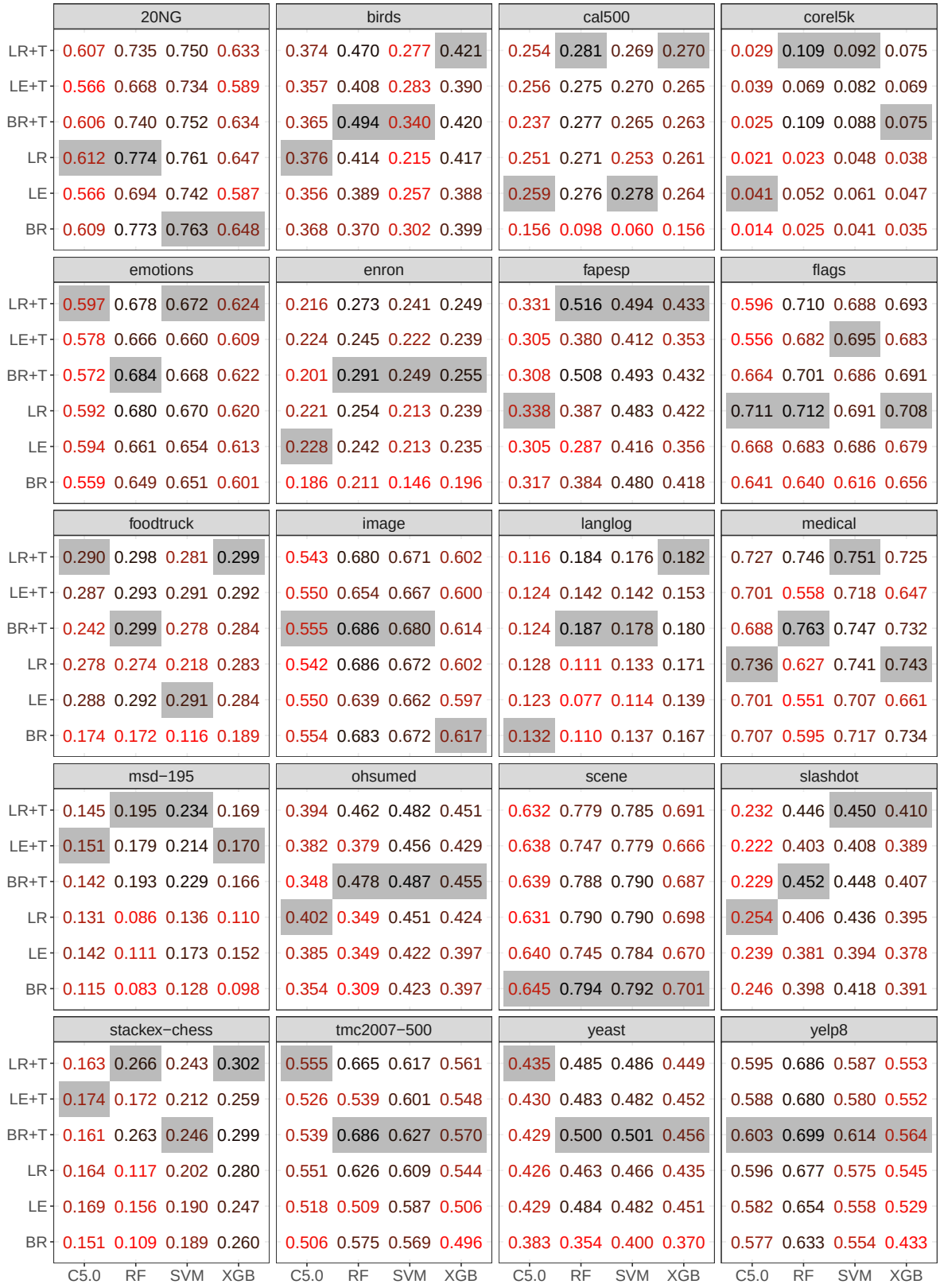


Figure 35 – Macro-F1 result of BR, LE and LR with and without threshold calibration. For each base algorithm, the best solution is highlighted with a gray background.

Figure 36 – Macro-precision result of BR, LE and LR with and without threshold calibration. For each base algorithm, the best solution is highlighted with a gray background.

Table 55 – Bayesian statistical probabilities for different pairs of strategies and evaluation measures. The highlighted values indicate probabilities higher than 50%.

left	right	C5.0			RF			SVM			XGB		
		left	rope	right	left	rope	right	left	rope	right	left	rope	right
						macro-AUC							
BR	LE	0.00	0.02	0.98	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.78	0.22
BR	LR	0.00	0.37	0.63	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
LE	LR	0.05	0.95	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.03	0.96	0.01
						macro-F1							
BR	BR+T	0.01	0.08	0.92	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
BR	LE	0.02	0.00	0.98	0.21	0.00	0.79	0.11	0.01	0.88	0.17	0.00	0.83
BR	LE+T	0.03	0.00	0.97	0.03	0.00	0.97	0.01	0.00	0.99	0.05	0.00	0.95
BR	LR	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.00	1.00
BR	LR+T	0.01	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
BR+T	LE	0.07	0.61	0.32	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
BR+T	LE+T	0.04	0.78	0.18	1.00	0.00	0.00	1.00	0.00	0.00	0.97	0.03	0.00
BR+T	LR	0.00	0.76	0.23	1.00	0.00	0.00	1.00	0.00	0.00	0.95	0.04	0.00
BR+T	LR+T	0.00	0.89	0.11	0.11	0.89	0.00	0.01	0.99	0.00	0.00	1.00	0.00
LE	LE+T	0.00	1.00	0.00	0.00	0.06	0.93	0.00	0.25	0.75	0.00	0.39	0.61
LE	LR	0.06	0.59	0.35	0.03	0.00	0.97	0.15	0.20	0.65	0.01	0.01	0.98
LE	LR+T	0.04	0.74	0.22	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
LE+T	LR	0.07	0.60	0.34	0.41	0.00	0.59	0.72	0.17	0.12	0.21	0.09	0.69
LE+T	LR+T	0.02	0.82	0.16	0.00	0.00	1.00	0.00	0.09	0.91	0.00	0.06	0.94
LR	LR+T	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.01	0.99	0.00	0.13	0.86
						macro-precision							
BR	BR+T	0.01	0.00	0.99	0.00	0.00	0.99	0.00	0.00	1.00	0.00	0.00	1.00
BR	LE	0.01	0.99	0.00	0.04	0.95	0.01	0.00	1.00	0.00	0.00	1.00	0.00
BR	LE+T	0.01	0.00	0.98	0.47	0.02	0.51	0.08	0.00	0.92	0.02	0.00	0.98
BR	LR	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
BR	LR+T	0.00	0.00	1.00	0.01	0.01	0.98	0.00	0.00	1.00	0.00	0.00	1.00
BR+T	LE	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
BR+T	LE+T	0.00	0.99	0.00	0.94	0.05	0.00	0.02	0.98	0.00	0.03	0.97	0.00
BR+T	LR	0.97	0.00	0.03	0.99	0.01	0.00	1.00	0.00	0.00	1.00	0.00	0.00
BR+T	LR+T	0.00	0.98	0.02	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00
LE	LE+T	0.01	0.01	0.99	0.00	0.04	0.96	0.00	0.00	1.00	0.00	0.00	1.00
LE	LR	0.00	1.00	0.00	0.00	0.98	0.02	0.00	1.00	0.00	0.00	0.99	0.01
LE	LR+T	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	1.00
LE+T	LR	0.95	0.00	0.05	0.37	0.11	0.52	0.95	0.01	0.04	0.92	0.00	0.08
LE+T	LR+T	0.00	0.92	0.07	0.00	0.03	0.96	0.00	0.93	0.07	0.00	0.92	0.08
LR	LR+T	0.00	0.00	0.99	0.01	0.01	0.99	0.00	0.00	1.00	0.00	0.00	1.00

APPENDIX

E

PERFORMANCE RESULTS OF OTHER STRATEGIES

Tables 56 to 60 present the predictive performance of the measures *macro-F1*, *macro-precision*, *macro-recall*, CLP and WLP, respectively. The bold markup indicates the best result for each dataset.

Table 56 – Macro-F1 results of distinct strategies and base algorithms.

Dataset	Base	BR	DBR	ECC	REMED	RAkEL	LE2	LR2
20NG	C5	.609(.00)	.601(.00)	.663(.00)	.603(.00)	.606(.00)	.604(.00)	.613(.00)
	RF	**.773(.00)**	**.773(.00)**	.762(.00)	.768(.00)	.607(.00)	.773(.00)	.773(.00)
birds	C5	.367(.02)	.370(.02)	.422(.01)	.197(.03)	.354(.03)	.364(.02)	.376(.02)
	RF	.373(.02)	.377(.03)	**.433(.01)**	.313(.01)	.340(.03)	.370(.03)	.405(.03)
cal500	C5	.156(.00)	.161(.00)	.184(.00)	.049(.00)	.164(.00)	.259(.00)	.251(.00)
	RF	.097(.00)	.086(.00)	.166(.00)	.041(.00)	.098(.00)	**.275(.00)**	.260(.00)
corel5k	C5	.014(.00)	.014(.00)	**.059(.00)**	.007(.00)	.019(.00)	.038(.00)	.020(.00)
	RF	.024(.00)	.026(.00)	.054(.00)	.029(.00)	.006(.00)	.043(.00)	.024(.00)
emotions	C5	.559(.01)	.563(.02)	.644(.01)	.422(.09)	.601(.01)	.580(.01)	.592(.01)
	RF	.652(.01)	.663(.01)	**.674(.01)**	.556(.01)	.673(.01)	.663(.00)	.671(.01)
enron	C5	.186(.00)	.189(.01)	.221(.00)	.097(.00)	.181(.00)	.210(.00)	.199(.00)
	RF	.210(.01)	.213(.00)	**.254(.00)**	.120(.01)	.220(.00)	.223(.00)	.244(.00)
fapesp	C5	.317(.04)	.316(.03)	.404(.03)	.293(.04)	.358(.02)	.317(.04)	.340(.01)
	RF	.394(.04)	.376(.04)	**.419(.04)**	.380(.04)	.019(.01)	.355(.04)	.393(.05)
flags	C5	.640(.03)	.616(.03)	.636(.03)	.402(.02)	.665(.02)	.668(.01)	**.711(.01)**
	RF	.647(.02)	.641(.02)	.635(.02)	.459(.01)	.653(.02)	.668(.02)	.709(.02)
foodtruck	C5	.174(.01)	.156(.01)	.213(.00)	.089(.01)	.181(.02)	.288(.01)	.278(.01)
	RF	.177(.01)	.148(.02)	.228(.01)	.108(.01)	.158(.01)	**.292(.01)**	.274(.01)
image	C5	.554(.01)	.517(.01)	.628(.00)	.555(.01)	.612(.01)	.555(.01)	.555(.01)
	RF	.683(.00)	.684(.00)	**.689(.00)**	.680(.00)	.688(.00)	.682(.00)	.682(.00)
langlog	C5	.132(.01)	.125(.01)	**.168(.00)**	.099(.01)	.087(.01)	.125(.01)	.129(.00)
	RF	.108(.00)	.104(.00)	.113(.00)	.113(.00)	.012(.00)	.086(.01)	.112(.00)
medical	C5	.707(.02)	.710(.02)	**.731(.01)**	.644(.02)	.716(.01)	.718(.02)	.710(.01)
	RF	.593(.01)	.585(.01)	.602(.01)	.588(.00)	.422(.01)	.563(.02)	.603(.01)
msd-195	C5	.115(.00)	.113(.00)	.137(.02)	.034(.00)	.103(.00)	**.142(.00)**	.131(.00)
	RF	.083(.00)	.079(.00)	.137(.00)	.082(.00)	.020(.00)	.111(.00)	.086(.00)
ohsumed	C5	.354(.00)	.364(.00)	**.426(.00)**	.164(.01)	.349(.00)	.362(.00)	.394(.00)
	RF	.308(.00)	.307(.00)	.352(.00)	.306(.00)	.243(.00)	.326(.00)	.329(.00)
scene	C5	.645(.00)	.614(.01)	.719(.00)	.643(.00)	.688(.00)	.643(.00)	.642(.00)
	RF	.792(.00)	.792(.00)	.791(.00)	.793(.00)	.771(.01)	.794(.00)	**.794(.00)**
slashdot	C5	.246(.01)	.244(.01)	.289(.01)	.219(.01)	.265(.01)	.250(.01)	.250(.01)
	RF	.398(.01)	.396(.01)	.400(.00)	.393(.00)	.331(.00)	.387(.01)	**.401(.01)**
stackex	C5	.151(.01)	.139(.00)	**.190(.00)**	.068(.01)	.113(.01)	.174(.01)	.157(.01)
	RF	.109(.00)	.109(.00)	.166(.00)	.106(.00)	.038(.00)	.118(.00)	.113(.00)
tmc2007	C5	.506(.00)	.517(.00)	.579(.00)	.180(.00)	.546(.00)	.512(.00)	.548(.00)
	RF	.574(.01)	.576(.01)	**.601(.01)**	.356(.01)	.576(.01)	.568(.00)	.593(.01)
yeast	C5	.383(.01)	.383(.00)	.399(.00)	.264(.00)	.400(.00)	.429(.00)	.425(.00)
	RF	.354(.00)	.364(.00)	.399(.00)	.257(.00)	.377(.00)	**.464(.00)**	.463(.00)
yelp8	C5	.577(.01)	.598(.01)	.659(.00)	.342(.00)	.646(.00)	.577(.01)	.596(.01)
	RF	.633(.00)	.655(.00)	**.667(.00)**	.453(.00)	.635(.00)	.633(.00)	.655(.00)

Table 57 – Macro-precision results of distinct strategies and base algorithms.

Dataset	Base	BR	DBR	ECC	REMED	RAkEL	LE2	LR2
20NG	C5	.661(.00)	.662(.00)	.675(.00)	.666(.00)	.775(.00)	.660(.00)	.651(.00)
	RF	.784(.00)	.785(.00)	.766(.00)	.786(.00)	**.925(.00)**	.784(.00)	.780(.00)
birds	C5	.419(.02)	.419(.03)	.507(.03)	.253(.05)	.463(.05)	.393(.02)	.387(.02)
	RF	.591(.08)	.604(.08)	.539(.06)	.492(.04)	**.709(.05)**	.590(.07)	.585(.07)
cal500	C5	.175(.00)	.192(.00)	.205(.00)	.086(.01)	.201(.01)	.185(.00)	.190(.00)
	RF	.179(.01)	.169(.01)	**.211(.01)**	.151(.01)	.170(.01)	.200(.00)	.187(.00)
corel5k	C5	.040(.00)	.047(.00)	.093(.00)	.021(.00)	.066(.01)	.053(.00)	.049(.00)
	RF	.088(.00)	.092(.00)	**.099(.00)**	.090(.01)	.059(.01)	.068(.00)	.095(.00)
emotions	C5	.567(.01)	.566(.02)	.658(.01)	.501(.15)	.637(.01)	.544(.02)	.538(.01)
	RF	.728(.01)	.720(.01)	.699(.01)	**.765(.01)**	.710(.01)	.680(.01)	.669(.02)
enron	C5	.236(.01)	.236(.01)	.281(.00)	.148(.01)	.260(.02)	.244(.00)	.222(.00)
	RF	.362(.01)	.373(.01)	.346(.01)	.286(.03)	**.381(.01)**	.300(.01)	.332(.01)
fapesp	C5	.345(.05)	.342(.05)	.446(.05)	.337(.05)	.481(.06)	.345(.05)	.365(.02)
	RF	**.529(.08)**	.492(.08)	.501(.08)	.479(.08)	.122(.05)	.465(.05)	.493(.09)
flags	C5	.680(.04)	.677(.04)	.663(.03)	.483(.07)	.690(.03)	.571(.02)	.614(.01)
	RF	.714(.06)	**.723(.05)**	.691(.05)	.610(.07)	.696(.05)	.631(.02)	.643(.02)
foodtruck	C5	.188(.01)	.199(.04)	.230(.02)	.108(.03)	.242(.05)	.215(.01)	.225(.02)
	RF	.342(.04)	.350(.04)	.271(.04)	.246(.04)	**.363(.05)**	.228(.02)	.232(.01)
image	C5	.523(.01)	.511(.01)	.645(.02)	.578(.01)	.640(.01)	.506(.02)	.502(.01)
	RF	.748(.00)	.750(.00)	.701(.00)	.757(.00)	**.758(.01)**	.747(.00)	.747(.00)
langlog	C5	.144(.01)	.145(.02)	.179(.02)	.113(.02)	.152(.03)	.124(.01)	.138(.01)
	RF	.196(.01)	.193(.01)	.190(.02)	.200(.03)	.043(.00)	.154(.03)	**.209(.01)**
medical	C5	.734(.03)	.739(.02)	**.771(.02)**	.687(.02)	.757(.03)	.740(.03)	.731(.02)
	RF	.746(.03)	.730(.03)	.726(.02)	.748(.02)	.635(.04)	.710(.04)	.746(.02)
msd-195	C5	.147(.00)	.132(.01)	.176(.01)	.038(.00)	.204(.02)	.124(.00)	.140(.01)
	RF	.260(.02)	**.268(.03)**	.251(.04)	.245(.02)	.262(.05)	.174(.01)	.256(.03)
ohsumed	C5	.467(.01)	.482(.01)	.510(.01)	.236(.02)	.531(.01)	.466(.01)	.466(.01)
	RF	**.589(.01)**	**.589(.02)**	.562(.01)	.585(.02)	.549(.03)	.584(.02)	.581(.02)
scene	C5	.614(.00)	.607(.00)	.720(.00)	.616(.00)	.752(.01)	.598(.00)	.607(.00)
	RF	.812(.00)	.814(.00)	.794(.00)	.816(.00)	**.862(.01)**	.814(.00)	.808(.00)
slashdot	C5	.343(.03)	.349(.04)	.432(.04)	.286(.02)	.492(.03)	.333(.03)	.346(.04)
	RF	.509(.04)	.511(.04)	.479(.03)	.516(.02)	**.519(.02)**	.453(.03)	.497(.04)
stackex	C5	.184(.01)	.174(.00)	**.251(.02)**	.079(.01)	.171(.01)	.186(.02)	.187(.01)
	RF	.218(.03)	.227(.02)	.211(.01)	.199(.01)	.125(.01)	.203(.01)	.228(.01)
tmc2007	C5	.634(.00)	.626(.00)	.686(.01)	.269(.02)	.676(.01)	.633(.01)	.586(.00)
	RF	**.873(.01)**	.869(.01)	.823(.01)	.868(.02)	.869(.01)	.818(.00)	.813(.01)
yeast	C5	.407(.01)	.373(.00)	.492(.02)	.323(.01)	.457(.01)	.348(.00)	.361(.00)
	RF	**.704(.02)**	.656(.02)	.619(.02)	.461(.05)	.687(.02)	.442(.00)	.437(.01)
yelp8	C5	.650(.00)	.612(.00)	.725(.00)	.452(.02)	.734(.01)	.634(.01)	.552(.01)
	RF	.881(.00)	.862(.00)	.787(.02)	**.929(.00)**	.879(.00)	.881(.00)	.755(.00)

Table 58 – Macro-recall results of distinct strategies and base algorithms.

Dataset	Base	BR	DBR	ECC	REMED	RAkEL	LE2	LR2
20NG	C5	.592(.00)	.580(.00)	.668(.00)	.581(.00)	.518(.00)	.570(.00)	.605(.00)
	RF	.770(.00)	.769(.00)	.767(.00)	.762(.00)	.479(.00)	.770(.00)	**.771(.00)**
birds	C5	.351(.02)	.355(.02)	.404(.01)	.180(.02)	.313(.03)	.374(.02)	.393(.03)
	RF	.311(.01)	.314(.02)	**.421(.02)**	.258(.01)	.257(.03)	.308(.02)	.348(.03)
cal500	C5	.156(.00)	.153(.01)	.195(.00)	.040(.00)	.157(.00)	.514(.01)	.422(.01)
	RF	.087(.00)	.080(.00)	.175(.00)	.030(.00)	.091(.00)	**.534(.01)**	.509(.03)
corel5k	C5	.012(.00)	.013(.00)	.053(.00)	.008(.00)	.012(.00)	.047(.00)	.017(.00)
	RF	.018(.00)	.020(.00)	.051(.00)	.021(.00)	.003(.00)	**.054(.00)**	.018(.00)
emotions	C5	.555(.02)	.565(.03)	.637(.02)	.382(.06)	.579(.03)	.660(.03)	.670(.03)
	RF	.614(.01)	.632(.02)	.671(.02)	.475(.01)	.658(.02)	**.686(.02)**	.684(.03)
enron	C5	.170(.00)	.177(.01)	.215(.00)	.079(.00)	.162(.00)	.267(.02)	.202(.00)
	RF	.170(.00)	.175(.00)	.231(.00)	.089(.00)	.177(.00)	**.285(.02)**	.234(.00)
fapesp	C5	.333(.03)	.334(.02)	.419(.03)	.293(.03)	.326(.04)	.333(.03)	.353(.03)
	RF	.367(.02)	.353(.03)	**.430(.03)**	.362(.02)	.011(.00)	.332(.04)	.371(.03)
flags	C5	.647(.05)	.599(.03)	.629(.04)	.380(.03)	.675(.03)	.851(.05)	**.861(.02)**
	RF	.640(.03)	.624(.02)	.628(.02)	.411(.01)	.653(.03)	.770(.03)	.810(.02)
foodtruck	C5	.171(.02)	.147(.01)	.223(.01)	.091(.01)	.169(.01)	.630(.05)	.419(.04)
	RF	.159(.01)	.132(.01)	.233(.01)	.101(.00)	.140(.01)	**.656(.09)**	.369(.02)
image	C5	.591(.01)	.526(.01)	.614(.02)	.536(.01)	.587(.01)	.630(.01)	.627(.02)
	RF	.630(.00)	.630(.00)	**.680(.00)**	.619(.00)	.631(.00)	.630(.00)	.630(.00)
langlog	C5	.136(.01)	.127(.01)	**.192(.01)**	.102(.01)	.070(.00)	.153(.01)	.134(.01)
	RF	.109(.00)	.106(.00)	.125(.00)	.113(.00)	.008(.00)	.103(.01)	.112(.00)
medical	C5	.704(.02)	.705(.02)	.730(.01)	.640(.02)	.707(.02)	**.747(.02)**	.718(.02)
	RF	.559(.01)	.550(.01)	.583(.01)	.558(.01)	.363(.02)	.546(.03)	.567(.01)
msd-195	C5	.103(.00)	.112(.00)	.140(.04)	.043(.00)	.081(.00)	**.204(.01)**	.133(.00)
	RF	.068(.00)	.066(.00)	.150(.00)	.068(.00)	.011(.00)	.105(.01)	.070(.00)
ohsumed	C5	.302(.00)	.312(.00)	**.403(.00)**	.144(.00)	.289(.00)	.326(.00)	.364(.00)
	RF	.248(.00)	.247(.00)	.311(.00)	.245(.00)	.181(.00)	.282(.00)	.279(.00)
scene	C5	.683(.00)	.627(.01)	.721(.00)	.674(.00)	.639(.01)	.715(.00)	.685(.01)
	RF	.775(.00)	.773(.00)	**.790(.00)**	.774(.00)	.704(.01)	.777(.00)	.783(.00)
slashdot	C5	.243(.01)	.238(.00)	.288(.01)	.225(.00)	.222(.01)	.258(.01)	.246(.01)
	RF	.373(.01)	.368(.00)	**.393(.01)**	.363(.00)	.283(.00)	.378(.01)	.381(.00)
stackex	C5	.143(.01)	.133(.01)	.195(.00)	.068(.00)	.099(.01)	**.203(.01)**	.152(.01)
	RF	.088(.00)	.087(.00)	.175(.00)	.087(.00)	.027(.00)	.126(.01)	.093(.00)
tmc2007	C5	.454(.00)	.467(.00)	.531(.00)	.150(.00)	.492(.00)	.465(.00)	.536(.01)
	RF	.482(.00)	.487(.01)	**.538(.01)**	.267(.01)	.485(.01)	.488(.01)	.526(.00)
yeast	C5	.376(.01)	.408(.01)	.386(.00)	.244(.01)	.392(.01)	**.646(.03)**	.543(.03)
	RF	.318(.00)	.344(.00)	.405(.00)	.218(.00)	.361(.01)	.624(.01)	.543(.01)
yelp8	C5	.542(.01)	.601(.01)	.637(.00)	.290(.01)	.605(.01)	.561(.01)	**.688(.03)**
	RF	.531(.00)	.568(.00)	.623(.01)	.339(.00)	.532(.01)	.531(.00)	.641(.00)

Table 59 – CLP results of distinct strategies and base algorithms.

Dataset	Base	BR	DBR	ECC	REMED	RAkEL	LE2	LR2
cal500	C5	.00(.00)	**.00(.00)**	.00(.00)	.00(.00)	**.00(.00)**	.15(.03)	.03(.01)
	RF	**.00(.00)**	**.00(.00)**	.01(.00)	**.00(.00)**	.00(.00)	.03(.02)	.02(.02)
flags	C5	.21(.07)	**.00(.00)**	**.00(.00)**	.12(.08)	.05(.10)	.31(.11)	.28(.09)
	RF	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	.01(.04)	**.00(.00)**
foodtruck	C5	**.00(.00)**	**.00(.00)**	**.00(.00)**	.00(.02)	**.00(.00)**	.06(.11)	.01(.03)
	RF	**.00(.00)**	**.00(.00)**	.05(.04)	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**
yeast	C5	**.00(.00)**	**.00(.00)**	**.00(.00)**	.05(.06)	**.00(.00)**	**.00(.00)**	.00(.02)
	RF	.01(.03)	.01(.03)	.14(.00)	**.00(.00)**	.05(.06)	.02(.05)	.02(.05)

Table 60 – WLP results of distinct strategies and base algorithms.

Dataset	Base	BR	DBR	ECC	REMED	RAkEL	LE2	LR2
birds	C5	.09(.04)	.08(.05)	.10(.04)	.56(.09)	.11(.06)	.06(.05)	.09(.05)
	RF	.18(.06)	.16(.07)	**.04(.05)**	.29(.05)	.20(.05)	.17(.07)	.14(.06)
cal500	C5	.39(.02)	.35(.02)	.38(.02)	.75(.04)	.37(.02)	.12(.02)	.16(.01)
	RF	.62(.04)	.67(.03)	.44(.04)	.74(.04)	.64(.02)	**.11(.02)**	.26(.02)
corel5k	C5	.91(.01)	.90(.01)	.66(.01)	.96(.00)	.86(.01)	.71(.03)	.89(.01)
	RF	.83(.01)	.80(.01)	.71(.02)	.79(.01)	.92(.01)	**.64(.02)**	.83(.01)
emotions	C5	**.00(.00)**	**.00(.00)**	**.00(.00)**	.16(.20)	**.00(.00)**	**.00(.00)**	**.00(.00)**
	RF	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**
enron	C5	.51(.03)	.49(.04)	.42(.03)	.77(.02)	.55(.01)	.24(.03)	.48(.02)
	RF	.37(.03)	.35(.02)	.30(.01)	.59(.04)	.33(.01)	**.21(.03)**	.35(.03)
fapesp	C5	.17(.10)	.17(.09)	.17(.06)	.30(.08)	.19(.08)	.17(.10)	**.13(.04)**
	RF	.26(.09)	.30(.08)	.25(.10)	.31(.08)	.87(.05)	.33(.07)	.32(.10)
flags	C5	**.00(.00)**	**.00(.00)**	.01(.04)	.37(.07)	**.00(.00)**	**.00(.00)**	**.00(.00)**
	RF	.01(.04)	.01(.04)	.04(.06)	.21(.07)	.02(.06)	**.00(.00)**	**.00(.00)**
foodtruck	C5	.57(.04)	.59(.06)	.45(.07)	.80(.05)	.53(.09)	**.03(.04)**	.13(.09)
	RF	.51(.03)	.51(.03)	.41(.06)	.62(.04)	.50(.05)	**.03(.05)**	.18(.11)
langlog	C5	.43(.02)	.46(.04)	.35(.03)	.61(.06)	.60(.04)	**.30(.03)**	.42(.02)
	RF	.58(.03)	.58(.02)	.57(.03)	.56(.03)	.94(.00)	.57(.03)	.57(.02)
medical	C5	.11(.03)	.11(.03)	.07(.04)	.16(.04)	.08(.04)	**.07(.02)**	.09(.03)
	RF	.18(.03)	.19(.03)	.17(.02)	.18(.02)	.32(.04)	.17(.03)	.17(.03)
msd-195	C5	.29(.04)	.29(.05)	.33(.07)	.87(.02)	.35(.04)	**.13(.04)**	.23(.05)
	RF	.50(.01)	.50(.01)	.39(.03)	.51(.03)	.70(.06)	.36(.04)	.51(.02)
ohsumed	C5	.24(.04)	.21(.03)	**.04(.01)**	.66(.03)	.13(.02)	.09(.02)	.17(.02)
	RF	.20(.02)	.21(.03)	.14(.02)	.20(.02)	.33(.03)	.14(.02)	.19(.03)
slashdot	C5	.42(.03)	.42(.04)	.21(.04)	.53(.04)	.28(.04)	.22(.08)	.41(.05)
	RF	.12(.04)	.15(.06)	.13(.06)	.13(.03)	.30(.03)	**.10(.04)**	.14(.06)
stackex	C5	.56(.04)	.59(.02)	.43(.03)	.82(.02)	.63(.03)	**.37(.03)**	.55(.04)
	RF	.62(.03)	.61(.02)	.48(.01)	.63(.01)	.83(.01)	.56(.02)	.62(.01)
tmc2007	C5	.02(.02)	**.00(.00)**	**.00(.00)**	.65(.03)	.00(.01)	.00(.01)	**.00(.00)**
	RF	**.00(.00)**	**.00(.00)**	**.00(.00)**	.01(.02)	**.00(.00)**	**.00(.00)**	**.00(.00)**
yeast	C5	.10(.03)	.07(.00)	.12(.03)	.50(.02)	.12(.03)	**.01(.03)**	.06(.02)
	RF	.14(.00)	.14(.00)	.14(.00)	.45(.04)	.14(.00)	.07(.00)	.12(.03)
yelp8	C5	**.00(.00)**	**.00(.00)**	**.00(.00)**	.36(.04)	**.00(.00)**	**.00(.00)**	**.00(.00)**
	RF	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**	**.00(.00)**


ICMC
USP
SÃO CARLOS