UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Imbalanced classification tasks: measuring data complexity and recommending techniques

Victor Hugo Barella

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

Victor Hugo Barella

Imbalanced classification tasks: measuring data complexity and recommending techniques

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

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

USP – São Carlos
April 2021

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

```
        Barella, Victor Hugo
B248i      Imbalanced classification tasks: measuring data
        complexity and recommending techniques / Victor
        Hugo Barella; orientador André Carlos Ponce de Leon
        Ferreira de Carvalho. -- São Carlos, 2021.
           148 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, 2021.


           1. Machine learning. 2. Imbalanced datasets. 3.
        Data complexity. 4. Meta-learning. 5. Meta-
        features. I. de Carvalho, André Carlos Ponce de
        Leon Ferreira, 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

Victor Hugo Barella

Tarefas de classificação desbalanceadas: medindo complexidade de dados e recomendando técnicas

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. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

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

USP – São Carlos
Abril de 2021

ACKNOWLEDGEMENTS

Incredible people supported me professionally and emotionally, which made this dissertation possible. I name a few of them below.

I would first like to thank my supervisor, Dr. André de Carvalho, who kindly guided me in doing this research. You care about your students, our research and promote the best environment for us to work. Thank you for being such an admirable human and supervisor.

I would like to acknowledge Drs. Marcílio de Souto and Nathalie Japkowicz for supervising me during my internships. Your hospitality was fundamental for me to feel welcomed in different countries. Your feedback pushed me to sharpen my thinking, scientific writing, and presentation skills.

I would also like to thank Drs. Luís Paulo Garcia and Ana Lorena, who collaborated with me during this research. Our fruitful discussions were essential to bring this research to the level it is.

In addition, I would like to thank my parents, Marta and Armando, for being supportive and understanding. You have never stopped growing as people, and that inspires me to keep on growing.

In São Carlos, I met incredible people and shared excellent moments with them. All of these moments transformed me into the person and researcher I am today. Here is a small sample of those people: Thomás, Dario, Amanda, Mariana, Gean, Luís Paulo, Everlândio, Clara, Giulia, Laís, Carolina, Saulo, Edesio, Moisés, José Pedro, Sthéfanny, Adriano, Davi, Lucas, Iury, Ângelo, Victor, Glauco, Eduardo, Anderson, Julio, Evandro and Alexandrina.

Moreover, I would like to thank the friends I made abroad during my internships. Although I was so far, I have never felt so at home. I thank Gregory, Teresa, Manuel, Nathan, Melanie, David, Michael, Roberto, Yohan, Caio, Bilal, Vidal, Hala, Alicia, Louis-denis, Céline, and many others.

Finally, I would like to acknowledge the funding I had during this research. I want to thank the São Paulo Research Foundation (FAPESP) (grants #2015/01382-0 and #2019/13015-2), the CAPES/COFECUB program, and CNPq. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

These acknowledgments represent many other people who have helped me remain strong and make this research possible.

“A vida em seus métodos diz calma
vai com calma você vai chegar
se existe desespero é contra calma
e sem ter calma nada você vai encontrar”
(Di Melo)

ABSTRACT

BARELLA, V. **Imbalanced classification tasks: measuring data complexity and recommending techniques**. 2021. 148 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, 2021.

Machine learning classification algorithms tend to perform poorly in datasets with class imbalance. Class imbalance is not a problem per se, but it poses adverse effects when combined with other data characteristics, such as class overlap and noise. This study aims to measure data characteristics in imbalanced datasets and recommend techniques to deal with class imbalance in a meta-learning system.

Popular data complexity measures were decomposed per class to better assess the imbalanced datasets characteristics. They were applied to controlled artificial datasets and to real datasets. These measures were correlated with several classification models' predictive performance. The measures were also evaluated before and after applying popular pre-processing techniques for imbalanced datasets. Moreover, a meta-learning system was implemented using popular meta-features along with the data complexity measures developed in this research. The results showed that decomposing the data complexity measures per class improved their ability to measure complexity in imbalanced datasets. Furthermore, according to experimental results, they were the most important meta-features in the meta-learning system.

Based on the results, data science practitioners should consider measuring the data complexity of imbalanced datasets, whether it is to interpret the data characteristics, select techniques, or develop new techniques.

Keywords: Machine learning, Imbalanced datasets, Data complexity, Meta-learning, Meta-features.

RESUMO

BARELLA, V. **Tarefas de classificação desbalanceadas: medindo complexidade de dados e recomendando técnicas**. 2021. 148 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, 2021.

Algoritmos de classificação em aprendizado de máquina tendem a desempenhar pior em dados com classes desbalanceadas. Desbalanceamento de classes não é um problema sozinho, mas provoca efeitos adversos quando combinado com outras características de dados, como sobreposição de classes e ruído. Este estudo tem por objetivo medir características de dados desbalanceados e recomendar técnicas para lidar com desbalanceamento por meio de um sistema de meta-aprendizado.

Nesta pesquisa, medidas populares de complexidade de dados foram decompostas por classe para melhor aferir as características de dados desbalanceados. Elas foram aplicadas em conjuntos de dados artificiais controlados e conjuntos reais. Essas medidas foram correlacionadas com o desempenho preditivo de diversos modelos de classificação. Elas também foram avaliadas antes e após a aplicação de famosas técnicas de pré-processamento pra dados desbalanceados. Além disso, um sistem de meta-prendizado foi implementado usando meta-atributos populares na literatura juntamente com as medidas de complexidade de dados desenvolvidas nessa pesquisa. Os resultados mostraram que decompor as medidas de complexidade por classe melhorou sua habilidade em medir complexidade em dados desbalanceados. Ademais, de acordo com os resultados dos experimentos, elas foram os meta-atributos mais relevantes para o sistema de meta-aprendizado.

Baseado nos resultados desta pesquisa, praticantes de ciência de dados devem considerar medir a complexidade de conjuntos de dados desbalanceados, seja para interpretar características de dados, selecionar técnicas ou desenvolver novas técnicas.

Palavras-chave: Aprendizado de máquina, Dados desbalanceados, Meta-aprendizado, Meta-atributos.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

Machine Learning (ML) is an area of study whose objective is programming computers to learn (MITCHELL, 1997). A classification task is a specific form of task in ML. It consists of identifying which of a set of categories a new observation belongs to based on previous observations (FLACH, 2012). The previous observations are expressed by a set of pairs (T_i, y_j), where T_i is usually a set of characteristics, and y_j is its category. The set of characteristics is called features, the categories are called classes, and the set of pairs are called the training set. In ML, a computer program follows a classification algorithm to induce a function to map each T_i to their corresponding y_j.

One factor that may hinder the adequate induction of a function is the class imbalance, which is a disproportion in the number of observations between the training set classes (FERNÁNDEZ *et al.*, 2018). Several authors reported poor performances in the classes less represented, which are called minority classes and usually are the classes of interest (YANG *et al.*, 2009; TAVALLAEE; STAKHANOVA; GHORBANI, 2010; CHEN *et al.*, 2018). Class imbalance is the object of study of this thesis. We investigated tools to assess better the nature of the problem and the data-level techniques. We also explored a recommendation system to suggest different approaches techniques to mitigate the effects of class imbalance.

This chapter is organized as follows: Section 1.1 discusses the nature of the class imbalance problem and motivates our research, Section 1.2 presents the objective and the hypotheses of this thesis, and Section 1.3 outlines the organization of this document.

1.1 The Class Imbalance Problem

Classification models induced on datasets with class imbalance tend to underperform in the minority classes. To identify this problem, better performance measures rather than accuracy are needed (JAPKOWICZ; SHAH, 2011). The accuracy problem is that it favors models with high performance on the majority classes even when they perform poorly on the minority classes. The most common performance measures in the literature of class imbalance are gmean, f-measure, and AUPRC (FERNÁNDEZ *et al.*, 2018). The right choice of the performance measure will depend on the domain and task at hand.

Besides the importance of the performance measure, another key to understanding the problem is the nature of the class imbalance problem. Although class imbalance is usually in conjunction with low performance in the minority classes, it is not a rule. The reason is that class imbalance is not a problem, per se. It combines with other data characteristics forming a particular effect that is harmful to the classification of the minority classes (BATISTA; PRATI; MONARD, 2004; JO; JAPKOWICZ, 2004; SáEZ *et al.*, 2014; KHOSHGOFTAAR; HULSE; NAPOLITANO, 2011; MALDONADO; WEBER; FAMILI, 2014). Figure 1 shows three different datasets. Blue circles represent the majority class, orange squares represent the minority class, and the black line represents a linear SVM classifier induced from the data. Each dataset represents a different data characteristic, i.e., linear separability, class overlap, and label noise.

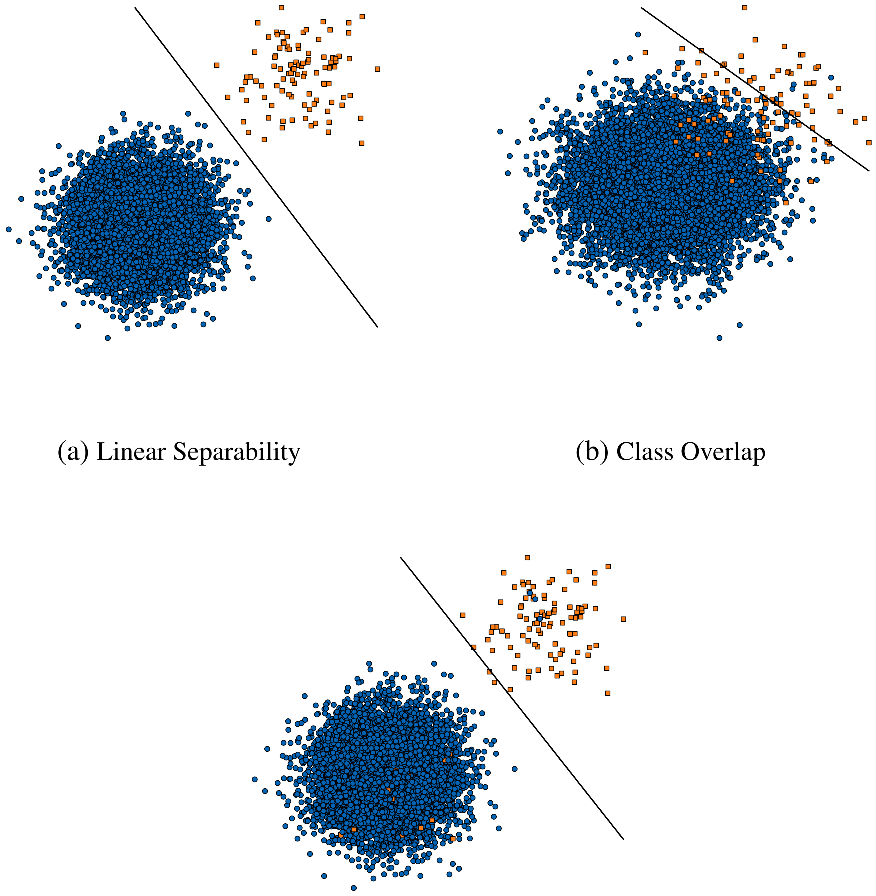
When the classes are linearly separable in the feature space (Figure 1a), a suitable linear model can be induced even when the classes are imbalanced. In the face of overlap (Figure 1b), the model tends to favor the majority class, degrading its performance in the minority class. The same holds when the data is noisy (Figure 1c).

Although all the data characteristics mentioned above degrade the performance also in balanced datasets, the problem has its particularities when the datasets are imbalanced. The most important of them is that the performance in the minority class is the one usually affected. Also, domains with class imbalance are usually more interested in identifying the minority classes than the majority classes, such as in oil spill detection in the sea (KUBAT; MATWIN *et al.*, 1997), fault diagnosis (YANG *et al.*, 2009), and medical diagnosis (MAZUROWSKI *et al.*, 2008). That means when a classification task is imbalanced, the models induced tend to perform poorly in the class of interest.

In recent years, researchers have focused on measuring such data characteristics to understand ML's problems better and propose techniques to mitigate their effects (HO; BASU, 2002; LORENA *et al.*, 2019). Measures proposed with that objective have been called data complexity measures. Although they have been used to express the complexity of imbalanced datasets (LUENGO; HERRERA, 2015; DÍEZ-PASTOR *et al.*, 2015; FERNÁNDEZ; JESUS; HERRERA, 2015), this thesis shows that they do not work adequately in such scenarios. We present a simple but effective adaptation of them, consisting of a decomposition per class. Our experiments demonstrate that our adaptation assesses better the difficulty in both artificial and real imbalanced datasets. We also investigated the relationship between data complexity and predictive performance before and after applying some popular pre-processing techniques for the class imbalance problem.

Pre-processing is a popular approach to mitigate the effects of class imbalance (BARUA

Figure 1 – Examples of data characteristics in imbalanced datasets and linear models induced from them



(a) Linear Separability

(b) Class Overlap

(c) Label Noise

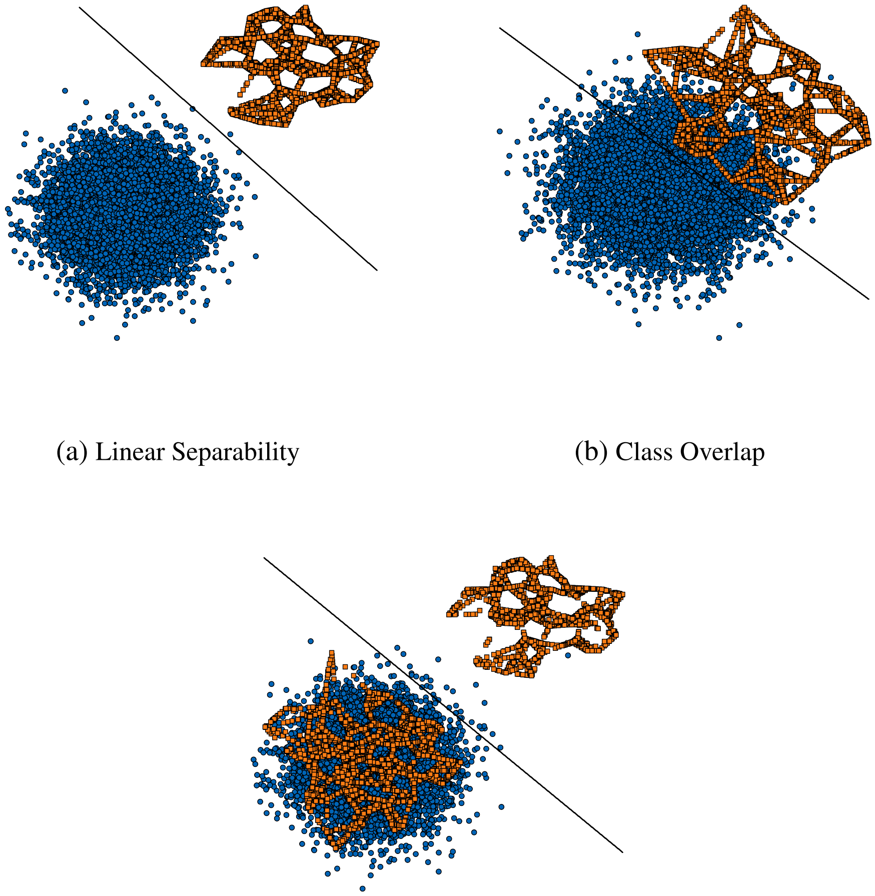
Source: Elaborated by the author.

et al., 2014; YU; NI; ZHAO, 2013; NG *et al.*, 2015). It consists of modifying the dataset to make it more balanced. Oversampling and undersampling are the main strategies. The former adds examples to the minority classes employing duplication or synthesis of examples, and the latter removes examples from the majority classes. Pre-processing techniques are used independent of the learning step, but it may remove significant instances or generate noise in the training set.

SMOTE is a popular oversampling technique that generates new examples by interpolating minority class instances (CHAWLA *et al.*, 2002). Figure 2 shows the datasets from Figure 1 and linear SVM classifiers after applying SMOTE to them. SMOTE helped the classification algorithm identify the minority class better, but it gave the minority class a non-natural shape and oversampled noise data. In a more complex scenario, SMOTE could not have improved enough the classifier.

Another possible approach is the algorithm-level approach, which consists of adapting

Figure 2 – Examples of data characteristics in imbalanced datasets after applying SMOTE and linear models induced from them



(a) Linear Separability

(b) Class Overlap

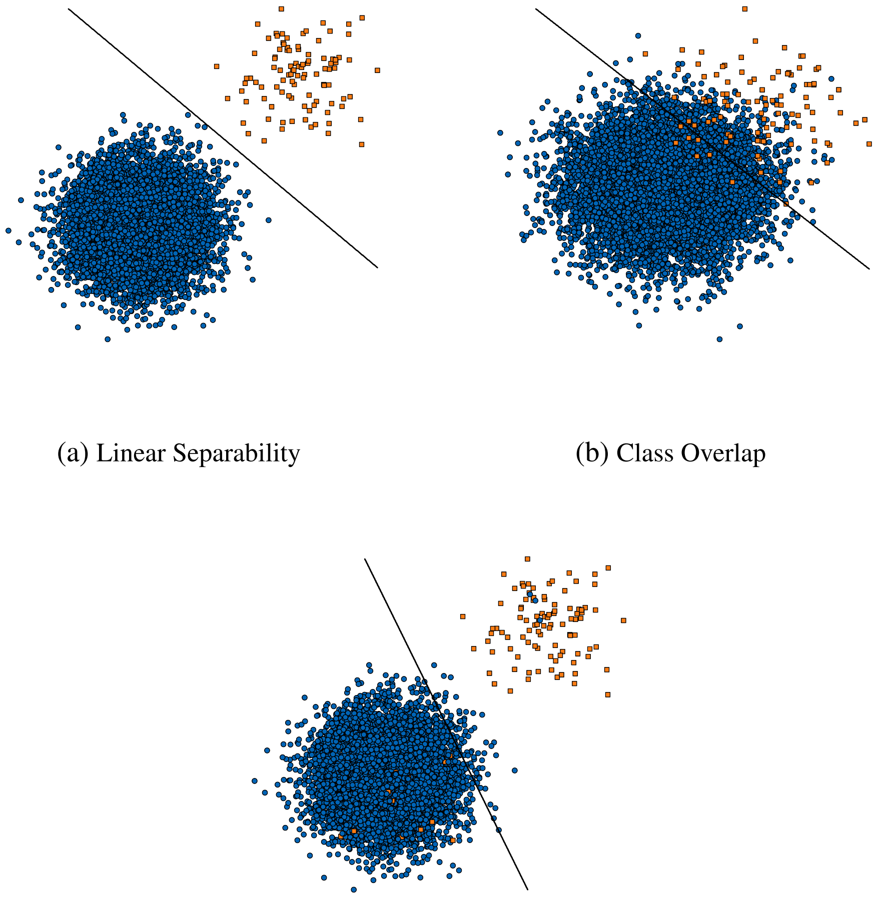
(c) Label Noise

Source: Elaborated by the author.

classification algorithms to take imbalance into account during the learning step (VEROPOULOS *et al.*, 1999; SEIFFERT *et al.*, 2009). Some adaptations include new hyper-parameters, such as different weights for each class or a cost matrix. Other approaches are based on adapting ensembles and outlier detection techniques, for example, one-class classification. Algorithm-level techniques usually do not include a pre-processing step in the pipeline, but it is dependent on the classification algorithm used.

Using class weights on SVM classifiers is an example of an algorithm-level approach (VEROPOULOS *et al.*, 1999). Figure 3 shows the datasets from Figure 1 and linear SVM classifiers with class weights induced from them. The use of class weights helped the classification algorithm better identify the minority class, but it could not have improved the classifier in a more complex scenario.

Figure 3 – Examples of data characteristics in imbalanced datasets and linear models with class weights induced from them



(a) Linear Separability

(b) Class Overlap

(c) Label Noise

Source: Elaborated by the author.

No technique works best for all datasets, which leads researchers to develop approaches to recommend a good technique depending on the dataset (MORAIS; MIRANDA; SILVA, 2016; BORSOS; LEMNARU; POTOLEA, 2018; SMOLYAKOV *et al.*, 2019; COSTA *et al.*, 2020). A technique's performance is usually related to the data characteristics of the dataset, where the decomposed DCMs can be useful. This thesis investigated the use of meta-learning (MtL) on a recommendation system to suggest pre-processing and algorithm-level techniques. We used MtL to induce models to map the data characteristics to the performance of techniques, considering the decomposed DCMs as meta-features. Our approach recommended techniques better than a baseline, and the decomposed DCMs were the most important meta-features in the system.

Although our experiments indicated the importance of the DCMs and decomposed DCMs as meta-features, their high computational cost may prevent them from being used as meta-features. Thus, we also investigated an MtL approach to estimate the decomposed DCMs. Our approach demonstrated to be useful in estimating the decomposed DCMs even when pre-processing techniques were applied.

1.2 Objective and Hypotheses

The main objective of this thesis was to assess the difficulty of imbalanced datasets through DCMs. To this end, we decomposed the DCMs per class and demonstrated their high correlation with the predictive performance considering several ML algorithms. Also, we showed that the gain in performance after applying pre-processing techniques is correlated with the reduction in data complexity. Moreover, we implemented and analyzed an MtL approach to recommend both pre-processing and algorithm-level approaches for imbalanced datasets, where the decomposed DCMs were the most important meta-features.

The main question that motivated this thesis is "How to define measures able to assess the complexity of imbalanced datasets and use these measures to recommend pre-processing and algorithm-level techniques with a good predictive performance for imbalanced datasets?"

The research question above guided the development of the following hypotheses to be tested in this thesis. The discussion for each hypothesis can be found in the chapters in parenthesis.

1. **The original DCMs do not properly assess the difficulty of imbalanced datasets.**

   (Chapters 2 and 3)

2. **Decomposing the DCMs per class improves their ability to measure data complexity in imbalanced datasets.**

   (Chapters 2 and 3)

3. **The gain observed in predictive performance after applying pre-processing techniques is related to the reduction in data complexity.**

   (Chapters 3 and 4)

4. **It is possible to recommend both pre-processing and algorithm-level techniques for imbalanced datasets automatically.**

   (Chapter 5)

5. **The decomposed DCMs are relevant characteristics to recommend techniques for imbalanced datasets properly.**

   (Chapter 5)

6. **It is possible to estimate the decomposed DCMs reducing their computational cost.**

   (Chapter 6)

1.3 Thesis Organization

This thesis is organized as a collection of papers. Thus, chapters 2 to 6 are articles written during the Ph.D. program. These chapters' order was chosen in a cohesive and progressive sequence, but they can be read in any order since they are self-contained. It is important to stress that some chapters share similar background sections due to this document's format. Chapter 7 presents the final remarks of this thesis. We describe below the following chapters with their main contributions.

- Chapter 2 - "Data Complexity Measures for imbalanced Classification Tasks"

  Using artificial datasets with controlled characteristics, we show that the DCMs do not assess the difficulty of imbalanced datasets properly. We introduce our proposed decomposition per class and show their effectiveness on those datasets.

- Chapter 3 - "Assessing the Data Complexity of Imbalanced Datasets"

  We extended the experiments from Chapter 2, considering real datasets retrieved from open access repositories. We show that the observations made on artificial datasets hold on to the real ones. Moreover, we analyzed the effects of applying popular pre-processing techniques for imbalanced datasets in the data complexity and their correlation with predictive performance gain.

- Chapter 4 - "The Influence of Sampling on Imbalanced Data Classification"

  This chapter shows how the decomposed DCMs change as we progressively increase the sampling size of popular pre-processing techniques. We compared its overall tendency with the tendency of the predictive performance of several ML algorithms.

- Chapter 5 - "Recommending Techniques for Imbalanced Datasets Using Meta-Learning and Data Complexity Measures"

  Applying MtL concepts, we implemented a recommendation system that suggests a set of techniques depending on imbalanced datasets' characteristics. We show that the system outperforms a baseline and that the decomposed DCMs are the most relevant meta-features to the system.

- Chapter 6 - "Simulating Complexity Measures on Imbalanced Datasets"

  In this chapter, we estimate the decomposed DCMs using MtL. We show that our approach reduces the computational cost of measuring them with a relatively small error rate.

- Chapter 7 - “Conclusion”

  We present the final remarks, summarizing the main contributions of this thesis, discussing the limitations, and suggesting future works.

CHAPTER

2

DATA COMPLEXITY MEASURES FOR IMBALANCED CLASSIFICATION TASKS

Authors

Victor H. Barella *University of São Paulo, São Carlos, São Paulo, Brazil*

Luís P. F. Garcia *Leipzig University, Leipzig, Germany*

Marcilio P. de Souto *University of Orleans, Orleans, France*

Ana C. Lorena *Federal University of São Paulo, São José dos Campos, São Paulo, Brazil*

André de Carvalho *University of São Paulo, São Carlos, São Paulo, Brazil*

Abstract

In imbalanced classification tasks, the training datasets may show class overlapping and classes of low density. In these scenarios, the predictions for the minority class are impaired. Although assessing the imbalance level of a training set is straightforward, it is hard to measure other aspects that may affect the predictive performance of classification algorithms in imbalanced tasks. This paper presents a set of measures designed to understand the difficulty of imbalanced classification tasks by regarding on each class individually. They are adapted from popular data complexity measures for classification problems, which are shown to perform poorly in imbalanced scenarios. Experiments on synthetic datasets with different levels of imbalance, class overlapping and density of the classes show that the proposed adaptations can better explain the difficulty of imbalanced classification tasks.

2.1 Introduction

In classification tasks, a training dataset is regarded as imbalanced when there is a disproportion between the number of examples of each class. A typical domain in which this inequality is found is the prediction of frauds in credit card transactions (BRAUSE; LANGSDORF; HEPP, 1999). Most of these transactions are legitimate, with a very low proportion of fraudulent transactions. Traditional Machine Learning (ML) classification algorithms tend to perform badly for the minority class, which in turn is usually the class of most interest. For instance, in the example of the credit card transactions, the predictions for the fraudulent transactions should be very precise, since classifying fraudulent transactions as legitimate can have a high financial cost.

Although the problem of imbalanced classes in ML was pointed out many years ago, it is still an open issue. In 1997, Kubat et al. discussed the difficulty found in some imbalanced training datasets and proposed a method to remove examples from the majority class (KUBAT; MATWIN *et al.*, 1997). Later, many methods were proposed in the literature to deal with the same problem by either preprocessing the training set (CHAWLA *et al.*, 2002; HAN; WANG; MAO, 2005; HE *et al.*, 2008) or learning adapted models which are aware of and can deal with class imbalance (SUN *et al.*, 2007; VEROPOULOS *et al.*, 1999). The state-of-art of the literature on imbalanced classification problems helps data specialists mitigate some of the challenges, but there is still a need to better understand the difficulties imposed in such sort of classification problems (HAIXIANG *et al.*, 2016).

Nonetheless, it is well known that an imbalanced dataset does not always impose problems on the minority class predictions. When the classes are linearly separable in the input feature space, it is not difficult to induce a proper classification model, even for imbalanced data (PRATI; BATISTA; MONARD, 2004). The problem arises when the classes overlap. In this case, traditional classification algorithms tend to ignore the minority class examples and focus on the majority class only. Therefore, it is worth investigating other aspects that may influence the difficulty in imbalanced classification tasks.

Ho and Basu (2002) gathered and proposed measures that help to assess the complexity of classification problems, such as class overlapping and density (HO; BASU, 2002). These measures estimate the difficulty of a classification task through information from the training dataset only. Some of the complexity measures assess the overlapping by the range of the features values, others use neighborhood information to measure the classification difficulty and there are also measures which assess the linearity of a classification task.

We show in a controlled set of experiments that the original complexity measures of Ho and Basu (2002) do not properly estimate the classification difficulty when the training datasets are imbalanced. For such, synthetic datasets with different levels of difficulty were generated. They have different imbalance ratios, levels of overlapping, density values and dimensionalities. The data complexity measures were adapted in order to take into account each class individually.

Herewith, the difficulty of classifying each one of the classes, including the minority class, can be better assessed. Indeed, the results reveal that the original data complexity measures evaluate mostly the difficulty of the majority class. They also reveal that the adapted complexity measures properly evaluate the difficulty of the classes individually.

This paper is divided into five sections. Section 2.2 discusses how the complexity measures have been used in the literature related to imbalanced classification tasks. Section 2.3 describes the original complexity measures along with a description of how they are adapted to estimate the difficulty of each class separately. Next, Section 2.4 presents the experimental setup designed in this work to assess the difficulty of imbalanced classification tasks. The experimental results are shown and discussed in Section 2.5. Section 2.6 concludes this paper with contributions, limitations and future works.

2.2 Related Works

The complexity measures from Ho and Basu (2002) have been used in various analyses of classification problems, including: supporting data pre-processing tasks, such as noise identification (GARCIA; CARVALHO; LORENA, 2015); understanding the domains of competence of different ML techniques (LUENGO; HERRERA, 2015); generating datasets spanning different complexity levels (MACIÀ; BERNADÓ-MANSILLA, 2014).

Related with imbalanced data, Luengo et al. (2011) used the complexity measures to predict whether a preprocessing technique can be successful or not (LUENGO *et al.*, 2011). They found intervals of values of some of the complexity measures in which the techniques showed improved performance. Other authors also used the complexity measures to analyze the suitability of using a specific technique in imbalanced datasets. Díez-Pastor et al. (2015) used them to predict data complexity intervals for which some diversity-enhancing techniques may improve the results of an ensemble method (DÍEZ-PASTOR *et al.*, 2015). Fernándes et al. (2015) used one complexity measure combined with other characteristics (such as imbalance) in a multi-objective approach to select attributes and instances from a dataset (FERNÁNDEZ; JESUS; HERRERA, 2015).

Anwar et al. (2014) proposed a new complexity measure for imbalanced datasets (ANWAR; JONES; GANESH, 2014). The definition of their complexity measure is similar to one of the complexity measures from Ho and Basu (2002), but using more neighbors in a nearest neighbor classifier. They also present a procedure to optimize the number of neighbors to be regarded which can be quite costly. However, the main contribution of their work is to show that the complexity measure values may be decomposed by class. They have also decomposed some of the original complexity measures to estimate classification complexity by class. But their work concluded that, despite these decompositions, these measures remain unable to properly estimate the difficulty of imbalanced classification problems. Despite very correlated to our work, we understand that the work of Anwar et al. (2014) lacks a deeper analysis of the behavior of complexity measures in imbalanced problems, which we address here.

With this paper, we intend to fill the previous gaps by: (i) adapting most of the original complexity measures from Ho and Basu (2002) to assess the difficulty of each class individually; (ii) designing artificial datasets with controlled characteristics which helps us to understand better the effects of different aspects in the difficulty of imbalanced classification problems; (iii) performing an extensive set of experiments to show how the original and adapted measures behave in imbalanced classification problems of different characteristics. Our adaptations are very simple and do not imply in additional computational costs in the calculation of the measures. Experimentally, most of the adapted complexity measures are able to successfully estimate the difficulty of imbalanced classification problems.

2.3 Data Complexity Measures and Adaptations

This section describes the original data complexity measures used in this paper and their adaptations for estimating the difficulty of each class in a dataset. The description of the original measures is based on the works (HO; BASU, 2002; LORENA; de Souto, 2015) and implemented in a revised R package (GARCIA; LORENA, 2018). The measures were separated into three main groups: feature overlapping, neighborhood-based and linear separability.

2.3.1 Feature overlapping measures

The feature overlapping measures assess the discrimination power of the input attributes.

F1: maximum Fisher's discriminant ratio

F1 computes the Fisher's discriminant ratio for each attribute, which is defined by:

$$f = \frac{(\mu_{c_1} - \mu_{c_2})^2}{\sigma_{c_1}^2 + \sigma_{c_2}^2}, \tag{2.1}$$

in which μ_{c_j} and σ_{c_j} are respectively the mean and the variance of the values of the feature in class j. F1 outputs the maximum f among all input attributes. The higher the F1 value, the simpler is the classification problem concerning feature separability. Since F1 relates two means and variances, it was not possible to adapt it and obtain a similar information by class. Therefore, F1 is maintained as originally proposed in the experiments. We opted to include F1 in the analysis because its use is reported in many papers dealing with imbalanced datasets, e.g. (LUENGO *et al.*, 2011; FERNÁNDEZ; JESUS; HERRERA, 2015; DÍEZ-PASTOR *et al.*, 2015).

F2: Volume of overlap region

F2 computes the volume of the overlapping region of the classes using the minimum and maximum values of each input attribute per class. If the attribute ranges overlap in a certain region, this region is considered ambiguous for the attribute. Next, a product of the normalized size of the ambiguous regions for all attributes is output. As an example, suppose an attribute whose values for class 1 range between 0 and 1, and the values for class 2 range between 0.75 and 1.25. The overlapping region for this attribute has size 0.25. Taking the full range of values for normalization, the final overlapping for this attribute is $\frac{0.25}{1.25} = 0.2$. F2 is null if at least one of the attributes does not have any overlapping region and is equal to 1 when the classes are completely overlapped for all attributes. However, in the original measure, a given volume of overlapping may represent a low overlapping with respect to one class, whilst the overlapping is high for another class. The proposed adaptation considers the impact of the overlapping volume per class. The difference between the original F2 and the adaptation is the division of the size of the ambiguous region of each attribute by the range of the class of interest, instead of the range of all values of the attribute. Taking the previous example, F2 for class 1 would be $\frac{0.25}{1} = 0.25$ and F2 for class 2 would be $\frac{0.25}{0.5} = 0.5$.

F3: Feature efficiency

In F3, one feature is considered efficient depending on how much examples are not in an ambiguous region. For each attribute, the number of examples out of the ambiguous region is divided by the total number of examples. F3 outputs the maximum of such values among all the input attributes, which corresponds to the best discriminative attribute. F3 ranges in the $[0,1]$ interval and higher values are expected for simpler datasets. Similar to F2, F3 has a bias towards the majority class, since the minority class may be completely inside the ambiguous region and F3 can still be close to 1. The adaptation of F3 considers one class at a time. The F3 per class divides the number of examples from the class outside the ambiguous region by the number of examples from the class only.

F4: Collective feature efficiency

F4 uses the main concept of F3 but instead of getting the maximum value from all attributes, it combines their discrimination power. First, the most discriminative attribute according to F3 is found; next, the examples correctly discriminated by that attribute are removed. The previous steps are repeated until all examples are correctly discriminated or until all attributes are removed. F4 is the proportion of examples discriminated at the end of the process. As in F3, higher F4 values are obtained for simpler problems. Our adaptation of F4 computes the number of examples correctly discriminated in each class divided by the number of examples from that class.

2.3.2 Neighborhood measures

The neighborhood measures use the concept of Nearest Neighbors (NN) to assess classification difficulty.

N1: The fraction of points on the class boundary

N1 builds a minimum spanning tree (MST) that connects all the examples from a dataset based on their distances, despite their classes. Next, it counts the number of examples that are connected to at least one example from another class. Those examples are possibly borderline and the fraction of their number over the total number of examples is the final N1 measure. N1 is bounded between 0 and 1 and values closer to 0 represent a lower complexity. The adaptation from this work consists in calculating the number of examples from the class of interest which connects with another class divided by the number of examples from that class only.

N2: The ratio of average intra/inter class NN distance

N2 compares the intraclass and interclass dispersions of the classes. For each example, its distance from the NN of the same class (intraclass) and its distance to the NN of a different class (interclass) are computed. N2 is the ratio of the intraclass distances average and the interclass distances average. Higher values are expected for problems of higher complexity. By taking the averages of all examples, N2 values are biased towards the majority class. Our adaptation takes the averages of the intra and inter class distances for examples of one class at a time. Therefore, the N2 value for a specific class will take the ratio of two averages: the average of intraclass distances between the examples from that class only; and the average of the interclass distances for the examples from that class only.

N3: Leave-one-out error rate of the 1NN classifier

N3 gives the leave-one-out training error of a nearest-neighbor classifier, which is easy to be calculated and is a good indicator of the separability of the classes. When a dataset is highly imbalanced, N3 tends to be closer to the majority class error and it may become inadequate. Therefore, our adaptation of N3 takes the NN training error per class.

N4: Nonlinearity of a 1-NN classifier

N4 uses a method which creates a new test set by interpolating two randomly selected examples from the same class multiple times. Then an NN classifier using the training set is used to predict the labels of the examples in the interpolated test set. N4 gives the error rate achieved in this procedure. A value closer to one may indicate either that the classes are overlapped or that the classes are not convex. Using the same criterion as in N3, N4 was adapted to take the error rate per class.

T1: Fraction of maximum covering spheres

T1 tries to explain the training set with hyper-spheres. Suppose that every example in the training set has a hypersphere with radius zero. If we gradually increase the radius of all hyperspheres some of them will touch a hypersphere from a different class. When that happens both hyperspheres stop growing. The method stops when there is no more growing hypersphere. The hyperspheres that are contained in another hypersphere are discarded. T1 is the ratio between the number of remaining hyperspheres and the number of examples in the dataset. A number closer to 0 indicates that there is no need for many hyperspheres to describe the training set and a number closer to 1 indicates a higher complexity and that almost the same number of hyperspheres as number of examples is needed to describe the training set. Consider a binary training dataset highly imbalanced and completely overlapped. T1 may be low for this training set since a few number of hyperspheres is needed to describe the data compared to the number of examples. But we may notice that to describe the minority class, we need almost the same number of minority examples as hyperspheres. Therefore, our adaptation of T1 consists of taking the ratio between the hyperspheres needed to describe each class and the number of examples from that class.

2.3.3 Linear Separability Measures

The linear separability measures whether the classes can be linearly separable in the attribute space.

L1: The minimized sum of error distance of a linear classifier

In L1, one linear model (e.g. a linear SVM) is built using the training dataset and calculating the distances of erroneous instances to the obtained hyperplane. L1 is the sum of these distances. L1 is equal to 0 for linearly separable problems. In our adaptation, only the distances of erroneous examples from each specific class are summed up.

L2: The training error of a linear classifier

L2 is the training error of a linear classifier. Higher values are expected for non-linear problems. Our adaptation takes the error rate per class.

L3: Nonlinearity of the linear classifier

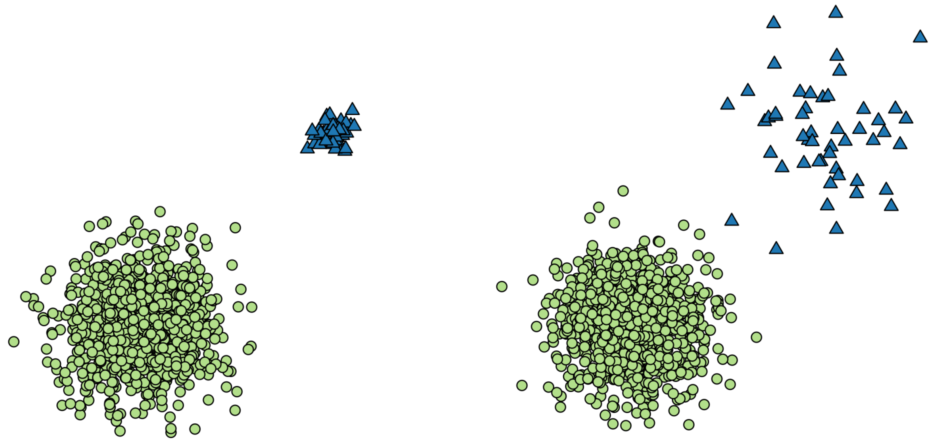
L3 is based on the same method of N4. A test set is interpolated and instead of an NN classifier, L3 uses a linear classifier to predict the labels of the examples from the test set. Our adaptation for L3 takes the error rate per class.

2.4 Experimental Setup

A set of experiments using synthetic datasets was designed to compare how the original complexity measures and the adapted complexity measures behave in imbalanced tasks. All datasets are binary classification problems generated by multivariate normal distributions with different levels of imbalance, class overlapping, dimensionality and density. The majority class has a fixed size of 1000 examples and the size of the minority class was varied from a high imbalance ratio to a balanced problem. To simulate class overlapping, the distances between the centroids of the classes distributions varied as completely overlapped to completely separated and the dimensionality was changed using different numbers of input attributes.

Regarding density, two different settings are tested: one in which both minority and majority classes have the same density; and another in which the minority class examples occupy the same hyper-volume as the examples from the majority class. To illustrate this, Figure 4a shows an example in which the two classes have the same density and Figure 4b shows one example in which the two classes occupy the same hyper-volume. Both figures are bidimensional datasets in which one majority class of 1000 examples is represented by the green circles, one minority class of 50 examples is represented by the blue triangles and the centroids of the classes have a distance fixed in four. It is clear that in Figure 4a the examples from the minority class are concentrated in a region, whilst in Figure 4b they are distributed more sparsely. Fixing the radius of the majority class in two, all datasets in which the distance between the center of the classes is four are linearly separable, including the sparse ones. But in the sparse datasets, the margin of separation between the classes becomes narrower.

Figure 4 – Examples of synthetic datasets with a majority class size of 1000, minority class size of 50 and distance between centroids of 4



(a) High Density in the Minority Class (b) Low Density in the Minority Class

Source: Barella *et al.* (2018).

To control the density level of the classes, a method is applied to make the samples distribution within the input space as dense as desired. Equation 2.2 represents the density of a class c_i, where n_{c_i} represents the number of examples in that class and $hyperVolume_{c_i}$ gives the hyper volume of the region those examples occupy. The idea is to control the distance of the furthest example of a class from the class centroid and adjusting the distances of other examples based on this value. To achieve this objective we must consider Equation 2.3, which represents the hypervolume, in which r_{c_i} is the radius of the hypersphere which contains all examples from class c_i, d is the dimension of the input space (number of attributes) and α is a multiplicative factor that varies according to d. For d equal 2, 3 and 5, α is respectively π, $\frac{4\pi}{3}$ and $\frac{8\pi}{15}$. For instance, considering $d = 2$, Equation 2.3 resumes to $\pi r_{c_i}^2$, which is the hypervolume of an hypersphere in a bidimensional Euclidean space. The same reasoning applies to the other dimensions.

$$density_{c_i} = \frac{n_{c_i}}{hyperVolume_{c_i}} \tag{2.2}$$

$$hyperVolume_{c_i} = \alpha r_{c_i}^d \tag{2.3}$$

In order to make the minority class as dense as the majority class, we first need to find the density of the majority class. Using Equations 2.2, 2.3 and since the number of majority class examples as $1,000$, we get the result shown in Equation 2.4. After calculating the density of the majority class, we can isolate the radius of the minority class hyper sphere (r_{min}) by combining Equations 2.2 and 2.3 and replacing $density_{min}$ by $density_{maj}$. Equation 2.5 represents the radius found for the minority class in the case it is set as dense as the majority class. In this Equation, n_{min} is the number of examples in the minority class.

$$density_{maj} = \frac{1000}{\alpha 2^d} \tag{2.4}$$

$$r_{min} = 2\sqrt[d]{\frac{n_{min}}{1000}} \tag{2.5}$$

The radius of the minority class can be either two for a sparse dataset or given by Equation 2.5 for a dense distribution. Then, the factor k, which is the proportion used to adjust the position of every example, can be calculated. k is defined by Equation 2.6, in which r_{new} is the new desired radius and r is the original radius of the set of examples that should be adjusted. Equation 2.7 calculates the new value of the i-th attribute of an example $\mathbf{x}^{c_j}$ from class c_j, where $\mu_i^{c_j}$ is the mean value of the i-th attribute of all examples in class c_j. After applying Equation 2.7 on all examples from the class and all their attributes, the class becomes as dense as desired.

$$k = \frac{r_{new}}{r} \tag{2.6}$$

$$\{x_i^{c_j}\}_{new} = k(x_i^{c_j} - \mu_i^{c_j}) + \mu_i^{c_j} \tag{2.7}$$

For each scenario, 1,000 datasets are generated, giving a total of 180,000 different datasets. The size of the minority class was varied as 10, 50, 100, 500 and 1000 (from a high imbalance ratio to a balanced problem). To simulate class overlapping, the distances between the centroids of the classes distributions varied as 0 (completely overlapped), 0.5, 1, 1.5, 2 and 4 (completely separated). We varied the number of input attributes as 2, 3 and 5. Regarding density, we considered the scenario in which the minority and majority classes have the same densities and another where the minority class examples occupy the same hyper-volume as the examples from the majority class. The original data complexity measures and the adapted measures were applied to all datasets.

A test set for each training set was created using the same distribution as their original counterparts. For each training example, 500 test examples are created, so that the proportions of examples per class are kept the same as those of the training sets. Equation 2.7 was also applied to the test sets using the k values obtained from the training sets. SVMs classifiers with radial kernel are induced for all training datasets and tested on their corresponding test sets. The R package (MEYER *et al.*, 2017) was used in this induction, with default parameter values. The radial kernel was employed because the generated datasets come from normal distribution. The performance measures of total accuracy, accuracy per class and the geometric mean of the accuracies per class were calculated for the test sets. The performance metrics recorded are used as indicators of the difficulty of the classification problems. Therefore, the correlation between the performance of the classifiers and the complexity measures is used to evaluate their ability in estimating classification difficulty.

We used 100 times a 5 fold cross validation. We chose 5 folds because some of the datasets have a low number of minority examples and using 10 folds would result in test sets with 1 or 2 minority examples. For each training set, we calculated the complexity measures. We also computed the performance measures accuracy, positive accuracy, negative accuracy and gmean for the following classification algorithms: SVM (with C and gamma from 2^{-12} to 2^{12}); Random Forest; KNN (with k from 1 to 50); and Naive Bayes.

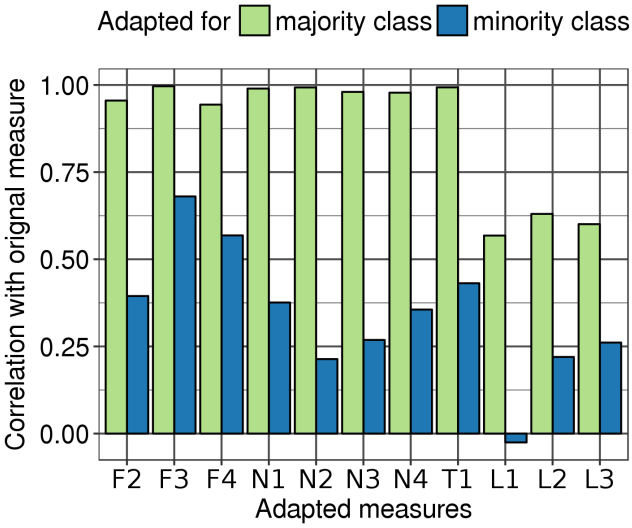
2.5 Results and Discussion

The main reason for our proposed adaptations of the data complexity measures is that their original definition poses a bias towards the majority class. We will show that this bias affects their ability to properly assess the difficulty in imbalanced classification tasks, in which the minority class is usually the class of interest. First, we show that the original data complexity measures have a similar behavior as the measures capturing the difficulty of the majority class only. Next, we show that the adapted measures properly assess the difficulty of imbalanced tasks, while the original complexity measures do not. We also detail the behavior of some of the complexity measures (highlighted in the previous analysis) by the level of class overlapping. Finally, we summarize all the analysis by regarding on the behavior of N3, which is the adapted complexity measure that showed a highlighted correlation to the difficulty of the imbalanced classification problems generated.

2.5.1 Correlation between the original and the adapted complexity measures

Figure 5 shows the Pearson correlation between the values of the adapted and the original data complexity measures. This analysis is performed by class, that is, the green bars show the correlations obtained for the complexity recorded for the majority class only, whilst the complexity for the minority class is shown by the blue bars. The x-axis represents each of the complexity measures and the y-axis represents the correlation of the adapted complexity measure to the corresponding original complexity measure.

Figure 5 – Pearson correlation between the adapted complexity measures and the original data complexity measures



Source: Barella *et al.* (2018).

For all measures, it is possible to notice that the complexity of the majority class correlates more to the values of the original complexity measures than the minority class complexity. The results are particularly highlighted for the overlapping and neighborhood measures. This demonstrates that the original complexity measures are biased towards the majority class. The linearity measures for the majority class are less correlated to the original values than the others mainly because of the fact that most of the datasets are highly overlapped. In these datasets, the linear models cannot distinguish any minority class example and they are accurate for all examples in the majority class. Taking only the datasets for which the distance between the classes are equal or greater than one (with less overlapping), the correlation between the original linearity complexity measures and the adapted counterparts to assess the majority class complexity gets higher. They are 86%, 92% and 83% for L1, L2 and L3, respectively.

2.5.2 Correlation between complexity measures and performance measures

In Figure 6a, we calculated the Pearson correlation between the original complexity measures values and the performance measures of the SVM classifiers in order to assess the potential of the complexity measures to estimate the classification difficulty posed by the synthetic classification datasets generated. The values of each complexity measure were correlated with two performance measures: accuracy (green bars) and the gmean (blue bars), which is a performance measure more suited to evaluate imbalanced classification problems. The x-axis represents the complexity measures and the y-axis shows the correlations. Figures 6b and 6c show similar plots for the adapted complexity measures considering the minority class and the majority class, respectively.

It is important to highlight that in Figure 6 some correlations are positive and others are negative. This happens because while for some of the complexity measures higher values indicate a more complex classification problem, for others an opposite relationship is verified. But all measures are behaving as expected considering these aspects, so that for more complex problems a low accuracy is verified.

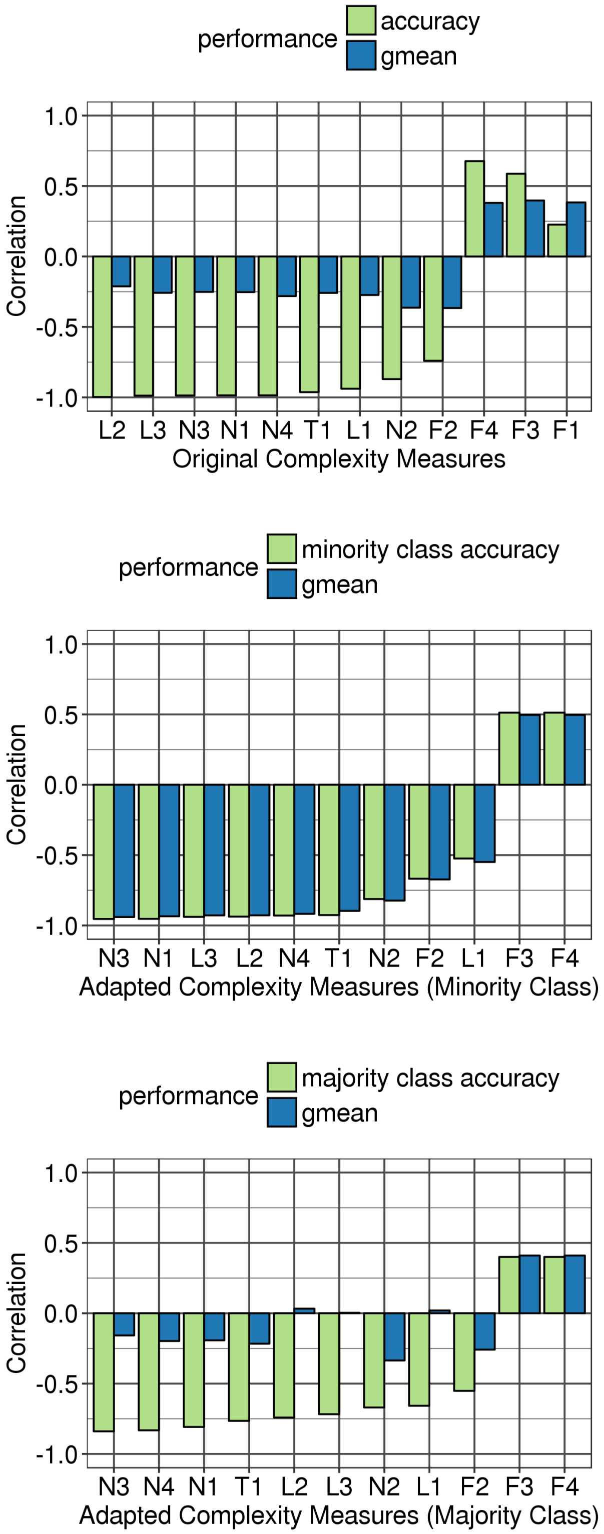
The original data complexity measures are extremely correlated to the accuracy performance metric. The problem is that accuracy is biased towards the majority class. For example, taking a test set with 90% of the examples in the majority class, a model that classifies all examples in this class will show an accuracy of 90%. Therefore, the standard accuracy measure is inadequate in such scenarios.

The correlation values between the original complexity measures and the gmean performance, on the other hand, decrease considerably. This indicates that the original data complexity measures do not point properly the difficulty of imbalanced classification problems. Not only the values are lower but the order of relevance of the complexity measures is inverted. That is, the complexity measures correlate the most with the accuracy are the least correlated with gmean and vice-versa. It is interesting to notice that the complexity measure F1 is the only one which is more correlated to the gmean than to the accuracy. Differently of the other complexity measures discussed, the mathematical definition of F1 cannot be directly associated with the accuracy bias, which can explain this result.

Although the original data complexity measures show a low correlation with gmean, our adaptation of the complexity measures for the minority class are highly correlated, as illustrated in Figure 6b. They are also highly correlated with the minority class accuracy. It is interesting to observe that the most relevant measures in Figure 6a are the same as those in Figure 6b, but with a slightly different order.

In Figure 6c, the adapted complexity measures values are highly correlated with the majority class accuracy, although the correlations are lower than those of the original complexity

Figure 6 – Graph bars showing the Pearson correlation between the data complexity measures and performance measures of the SVM classifiers. The measures are ordered by the correlation magnitude.



(a) Correlation between the original data complexity measures and performance measures accuracy and gmean

(b) Correlation between the adapted data complexity measures assessing the minority class and the performance measures minority class accuracy and gmean

(c) Correlation between the adapted data complexity measures assessing the majority class and the performance measures majority class accuracy and gmean

Source: Barella *et al.* (2018).

measures and of the adapted complexity measures for the minority class. We can also notice that the difficulty of the majority class is not highly correlated to the gmean performance. The correlations between the linear measures and the gmean are close to zero. This corroborates that the linear models usually are perfectly accurate on the majority class, but fail for the minority class examples.

Overall, the obtained results demonstrate that computing the complexity measures per class is more suitable in imbalanced classification datasets than using the original complexity measures.

2.5.3 Correlation between complexity measures and gmean by class overlapping

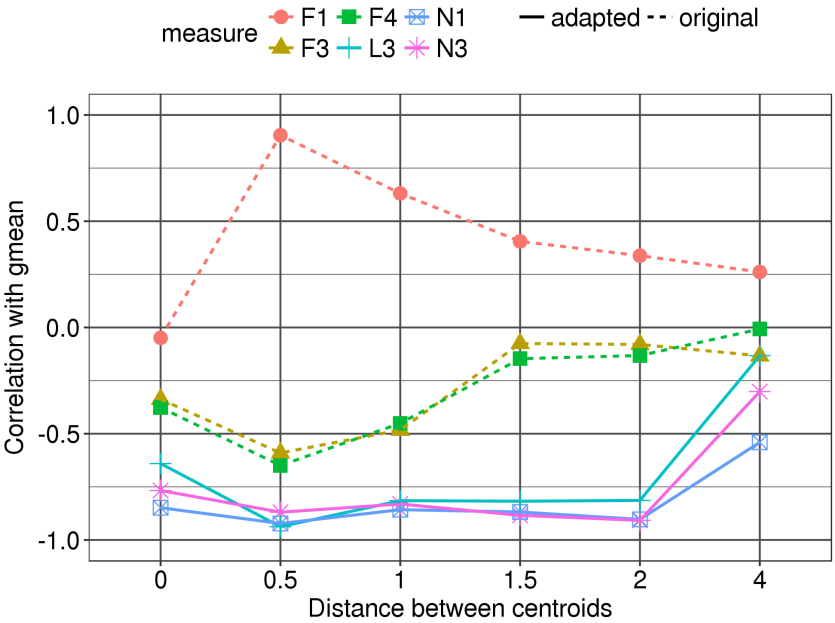
Overlapping, imbalance, density and dimensionality were varied in the simulated datasets generated. The correlations between the values of each of these characteristics and the gmean performance are higher for the overlapping factor - there is a 70% of Pearson correlation between the class overlapping (distances of the classes centroids) and gmean. The imbalance, density and dimensionality factors have, respectively, 33%, 7% and 0% of Pearson correlation with gmean.

Focusing on the class overlapping aspect, Figure 7 shows the correlation between the top three complexity measures (original and adapted) and the SVM gmean, detailed by the distance between the centroids of the classes. The solid lines correspond to the top three adapted measures for the minority class, namely L3, N1 and N3. The dashed lines correspond to the top three original data complexity measures, which are F1, F3, and F4.

Figure 7 reveals that the adapted measures estimate classification difficulty properly for all levels of class overlapping, except the last (less overlapping or distance four). In that case, the classes are linearly separable and the class imbalance is not an issue, as previously stated. The original data complexity measures F3 and F4 show their highest absolute value of correlation when distances are between 0 and 1 (high overlapping). So when the datasets have a high overlapping (distances between 0 and 1) and they have also a high imbalance rate, the problem gets very hard and minority class gets ignored by the classifier. But the original complexity measures still estimate those problems as simple, since the majority class is easy to identify. When the datasets have a high overlapping and a low imbalance ratio (the classes are more equilibrated), the majority class gets more difficult to be identified and the gmean values tend to be higher too. In summary, when there is a high overlapping, the original complexity measures tend to estimate erroneously the actual difficulty of the imbalanced classification datasets. After distance one, the correlations of the complexity measures values with gmean approach zero, indicating that they have lost the ability to assess the difficulty of imbalanced tasks in those scenarios.

The measure F1 is an exception. It does not assess the difficulty of the majority class

Figure 7 – Pearson correlation between measures and gmean by distance between centroids.



Source: Barella *et al.* (2018).

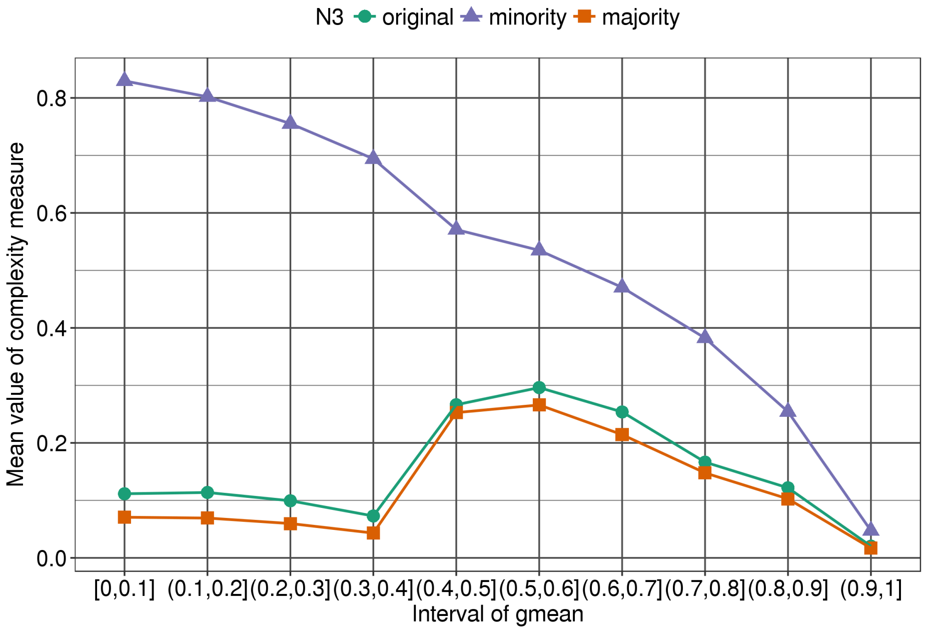
neither the minority class. It is a relation between the distance of the classes and their sparseness. When the classes are completely overlapped, F1 has a correlation close to zero with gmean. Although there is no correlation, it does not mean that F1 is not properly assessing the difficulty in those datasets. In those cases, F1 is mostly zero - showing that this measure describes those tasks as extremely difficult. When the distance between the classes is 0.5, F1 is highly correlated with gmean, showing that it is correctly evaluating the classification difficulty. After that distance, the correlation gets lower as the distances are increased. Despite the proper difficulty assessment ability F1 showed in most of the artificial datasets, its correlation with gmean is lower to that of the adapted complexity measures when the distances are higher than 0.5 (medium to low overlapping).

2.5.4 Behavior of N3

N3 was chosen as a representative to summarize the analysis performed so far. Figure 8 shows the behavior of this measure. The x-axis shows the gmean values, divided into intervals; and the y-axis shows the mean values of N3 for each gmean interval. The original N3 measure values are represented by green circles, the N3 values for the majority class are represented by the orange squares and the N3 values for the minority class are represented by the purple triangles. The original N3 measure and the N3 measure for the majority class behave similarly, as occurred in Figure 5. The N3 measure for the minority class is extremely correlated with gmean, while the original N3 complexity measure shows a low correlation, as in Figure 6.

It is interesting to detail the behavior of N3 by intervals of gmean. When the tasks are

Figure 8 – Behavior of measure N3 according to the variability of gmean.



Source: Barella *et al.* (2018).

very difficult (gmean values between 0 and 0.4), there is no balanced dataset and the imbalanced datasets have a high overlap - 95% of the datasets have distances between the centroids values in the interval $[0,1]$. In those cases, the majority class is not difficult to predict and the minority class is extremely difficult to predict, as the adapted complexity measures per class show.

In the interval of gmean between 0.4 and 0.6, 42% of the datasets are completely balanced and completely overlapped (the distance is 0 between the classes centroids). Also, 40% of the datasets are imbalanced, but with more than 10 examples in the minority class, and have distances values of 0.5 and 1. Those scenarios are the most difficult ones for the majority class. Our adaptation for the majority class is capturing this difficulty as shown by the rise of values in the opposite sense of the adapted measure for the minority class. This behavior can exemplify the inversed correlation for the original measures discussed in Figure 7. Also, the minority class is less difficult compared to what happens for the datasets from the gmean interval of 0 to 0.4, but they are still not simple either. Our adapted N3 assessing the minority class is estimating that appropriately.

In the interval of gmean between 0.6 to 1, the classes get more distant from each other until they are completely separated. The behavior of all N3 measures values properly shows that the tasks are getting easier for both minority and majority class.

It is important to mention that we do not suggest using the difficulty of the minority class to understand the whole dataset. Alternatively, it would be more interesting to assess the complexity of all classes separately to better understand the difficulty of the classification task as a whole.

2.6 Conclusion and Future Works

Imbalanced classification tasks are still an open problem in ML. For that sort of classification problems, the overlapping between the classes is one of the main issues, but calculating the overlapping level is not straightforward. One way to assess information from the training dataset is through the use of data complexity measures. We showed that the main data complexity measures from the literature do not properly assess the difficulty in the case of imbalanced classification problems. We adapted them to evaluate the classes separately, giving some focus to the minority class. Through an experiment with synthetic datasets, we showed that our adaptations estimate correctly the classification difficulty in imbalanced scenarios. We also showed that the original data complexity measures estimate mainly the difficulty of the majority class, which may be erroneous especially for imbalanced tasks with a high class overlapping. Although the difficulty of the minority class is usually the main challenge in imbalanced tasks, we still suggest assessing the difficulty of both classes to understand the whole classification problem.

We only analyzed the behavior of the measures through artificial datasets and we will investigate them in real datasets afterward. The datasets created are only binary classification problems and some measures cannot be directly used in multi-class classification problems - such as the linearity measures, which use linear SVM models. It is necessary to investigate approaches to assess the difficulty of the classes in imbalanced multi-class classification problems too.

The adapted measures can also be useful in meta-learning as meta-features and to guide the recommendation of pre-processing techniques, methodologies and classification algorithms. We expect that the presented tools can help data specialists to mitigate the problem of data imbalance in classification tasks.

Acknowledgments

The authors would like to thank FAPESP (processes 2015/01382-0, 2016/18615-0, 2013/07375-0 and 2012/22608-8), CAPES, CNPq and Intel for their financial support.

CHAPTER

3

ASSESSING THE DATA COMPLEXITY OF IMBALANCED DATASETS

Authors

Victor H. Barella *University of São Paulo, São Carlos, São Paulo, Brazil*

Luís P. F. Garcia *University of Brasília, Brasília, Distrito Federal, Brazil*

Marcilio P. de Souto *University of Orleans, Orleans, France*

Ana C. Lorena *Aeronautics Institute of Technology, Praça Marechal Eduardo Gomes, 50, São José dos Campos, São Paulo, Brazil*

André de Carvalho *University of São Paulo, São Carlos, São Paulo, Brazil*

Abstract

Imbalanced datasets are an important challenge in supervised Machine Learning (ML). According to the literature, class imbalance does not necessarily impose difficulties for ML algorithms. Difficulties mainly arise from other characteristics, such as overlapping between classes and complex decision boundaries. For binary classification tasks, calculating imbalance is straightforward, e.g., the ratio between class sizes. However, measuring more relevant characteristics, such as class overlapping, is not trivial. In the past years, complexity measures able to assess more relevant dataset characteristics have been proposed. In this paper, we investigate their effectiveness on real imbalanced datasets and how they are affected by applying different data imbalance treatments (DIT). For such, we perform two data-driven experiments: (1) We adapt the complexity measures to the context of imbalanced datasets. The experimental results show that our proposed measures assess the difficulty of imbalanced problems better than the original ones. We also compare the results with the state-of-art on data complexity measures

for imbalanced datasets. (2) We analyze the behavior of complexity measures before and after applying DITs. According to the results, the difference in data complexity, in general, correlates to the predictive performance improvement obtained by applying DITs to the original datasets.

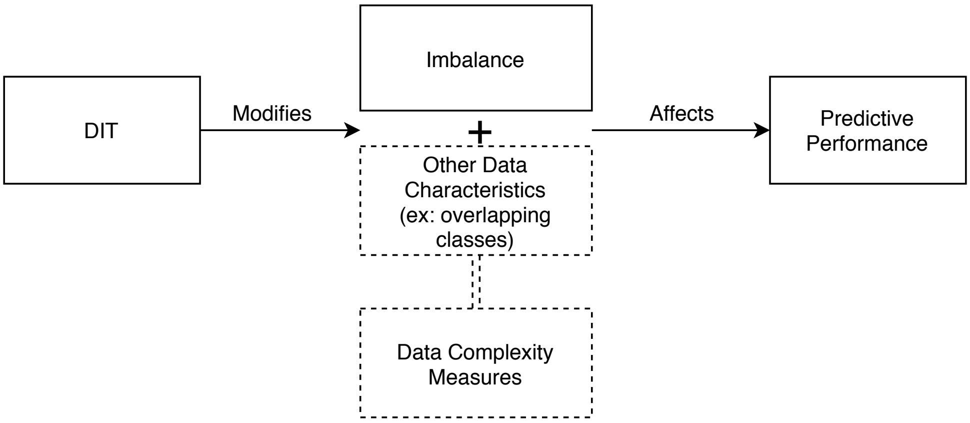
3.1 Introduction

In classification tasks, class imbalance is a disproportion of the number of instances from each class in the dataset. Although several articles report poor predictive performances of traditional Machine Learning (ML) algorithms when applied to these datasets (HE; GARCIA, 2008; FERNáNDEZ *et al.*, 2018; CHAWLA *et al.*, 2002; KUBAT; MATWIN *et al.*, 1997; BARUA *et al.*, 2014; ABDI; HASHEMI, 2016), Batista, Prati and Monard showed that imbalance is not a problem per se. In fact, it increases the adverse effect of other data intrinsic characteristics, such as class overlapping. Data topology characteristics, such as overlapping, linear separability, among others, are not easily measured. They have many more complex concepts and aggregate more information about the data than a simple class imbalance ratio.

Topological characteristics can be estimated by using data complexity measures, which were initially proposed by Ho and Basu and extended by many other authors (HO; BASU; LAW, 2006; ORRIOLS-PUIG; MACIá; HO, 2010; LORENA; de Souto, 2015; LORENA *et al.*, 2019). Several studies investigate the use of these measures in classification tasks (MACIÀ; BERNADÓ-MANSILLA, 2014; GARCIA; CARVALHO; LORENA, 2015; LUENGO; HERRERA, 2015; GARCIA *et al.*, 2018), some of them for imbalanced datasets (LUENGO *et al.*, 2011; DÍEZ-PASTOR *et al.*, 2015; FERNÁNDEZ; JESUS; HERRERA, 2015). Although their use in imbalanced classification tasks seems straightforward, Barella *et al.* showed that for artificial datasets, the complexity measures do not adequately represent the difficulties found in imbalanced datasets. To deal with this deficiency, the authors proposed modifications to these measures. The modifications consist of decomposing the data complexity for each class in the dataset. We investigate, using real imbalanced datasets, the effectiveness of those measures and the original ones. Additionally, we define them formally, make a package publicly available, and compare them with the state-of-the-art complexity measures for imbalanced datasets.

Data Imbalance Treatments (DITs) have been proposed to balance the number of instances between the dataset classes (FERNáNDEZ *et al.*, 2018; CHAWLA *et al.*, 2002; KUBAT; MATWIN *et al.*, 1997; BARUA *et al.*, 2014; ABDI; HASHEMI, 2016). Furthermore, they can modify other characteristics of the datasets, which can affect their predictive performance. In this paper, we also investigate the relation between data complexity measures and predictive performance before and after applying DITs. Figure 9 illustrates DITs modifying characteristics of a dataset and affecting the predictive performance. In this figure, the solid box represents what the literature usually discusses, which is the improvement of predictive performance based on balancing the classes. In this study, we investigate the improvement of predictive performance based on the decrease of data complexity, represented by the dashed box in the figure.

Figure 9 – Diagram of DIT techniques modifying data characteristics and affecting predictive performance



Source: Barella *et al.* (2020).

We believe this paper will help to understand the difficulty that an imbalanced dataset may pose to any classification algorithm and how DITs can deal with this problem. Thus, the main contributions of this paper are:

1. Formally define the adapted complexity measures to imbalanced domains;
2. Show that the adapted data complexity measures assess the difficulty on real imbalanced datasets;
3. Show that the adapted data complexity measures assess the difficulty of a dataset before and after applying DITs.

This paper is organized in five sections. Section 3.2 describes the original complexity measures, how they are adapted to estimate the difficulty of each individual class, and the state-of-art on complexity measures for imbalanced datasets. Moreover, it presents the main DITs considered in this paper, as well as related works. Next, Section 3.3 presents the experimental designs for this study. We show and discuss the experimental results in Section 3.4. In Section 3.5, we stress the main contributions and limitations, and indicate future work directions.

3.2 Background

In this section, we describe the main data complexity measures and our proposed adaptations of them for DIT. We also describe the main pre-processing techniques found in the literature for imbalanced classification.

3.2.1 Data Complexity Measures and Adaptations

The original data complexity measures were proposed by Ho and Basu and extended by many studies (HO; BASU; LAW, 2006; ORRIOLS-PUIG; MACIá; HO, 2010; LORENA; de Souto, 2015; LORENA *et al.*, 2019). Orriols-Puig, Maciá and Ho implemented a package called DCoL (Data Complexity Library) and proposed generalizations of complexity measures for multiclass problems. Moreover, limitations remained and some were solved later by Lorena *et al.*, who surveyed, standardized and implemented the cutting edge measures in a revised R package called ECoL (Extended Complexity Library) (GARCIA; LORENA, 2018). In order to adapt them for the imbalance problem, Barella *et al.* decomposed the measures to assess the complexity of each class separately. This subsection describes the original data complexity measures used in this paper, as defined by Lorena *et al.*, and their adaptations for estimating the difficulty of each class in an imbalanced dataset, proposed by Barella *et al.* and formalized here. The aim is to decompose the measures per class, enabling us to assess classification difficulties from the perspective of the minority class.

To describe the measures, we consider a training set T with n instances, in which each instance is a pair $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is a vector of characteristics (which we will call features) $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$, m is the number of features and $y_i \in \{0,1\}$. Consider also a function $c(T)$ whose output is the value of complexity measure c applied in dataset T, with $c(T) \in [0,1]$. According to Lorena *et al.*, the higher the $c(T)$ value, the more complex the dataset. The complexity measures are organized into three main groups: feature overlapping, neighborhood information, and linear separability.

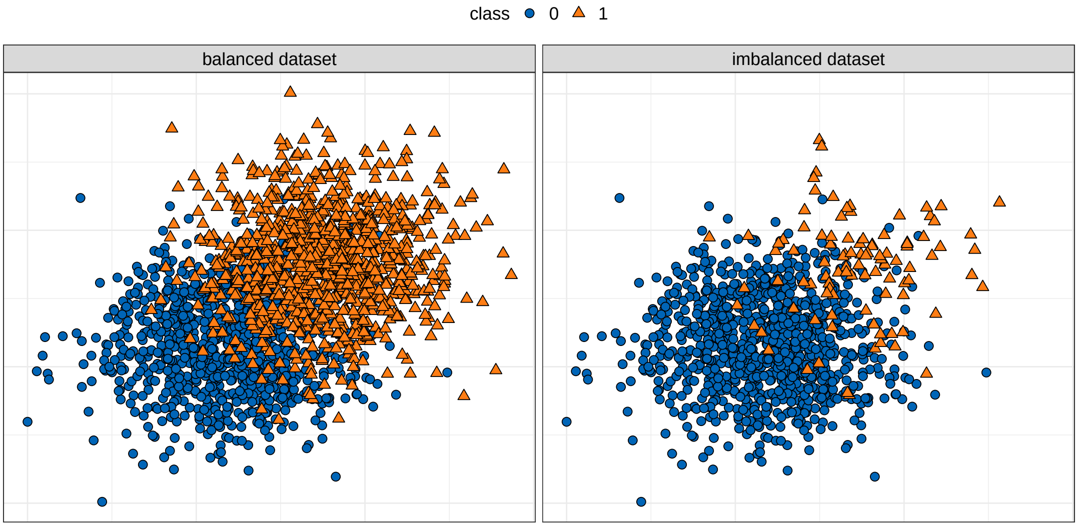
To illustrate the differences between the original and the adapted measures in balanced and imbalanced datasets, we use two artificial datasets. They are shown in Figure 10, where the classes were sampled from multivariate normal distributions. The class 0 represents the negative class, and the class 1 represents the positive class. Both classes have 1000 instances in the balanced datasets, and the class distribution in the imbalanced dataset is 1000 and 100 examples for class 0 and class 1, respectively. We show the values of the data complexity measures for the two datasets in addition to their detailed description.

3.2.1.1 Feature overlapping measures

The feature overlapping measures assess the discrimination power of the predictive attributes. Most of them evaluate the features individually and the most discriminate feature is selected, while others use a combination of the individual feature assessments. The feature overlapping measures considered in this article are F1, F2, F3, and F4. The feature overlapping complexity measures are detailed next, as well as a description of the proposed adaptation.

- **F1: Maximum Fisher's discriminant ratio.**

Figure 10 – Example datasets to illustrate the differences between the measures



Source: Barella *et al.* (2020).

F1 computes the Fisher's discriminant ratio for each attribute. The aim is to assess how close the classes, for each of the features in the feature space. To do this, the measure considers the mean and variance values of each feature in each class.

$$F1(T) = argmax_{j=1}^{m}(f_j) \tag{3.1}$$

$$f_j = \frac{(\mu_{fc_0} - \mu_{fc_1})^2}{\sigma_{fc_0}^2 + \sigma_{fc_1}^2} \tag{3.2}$$

F1 is defined by Equation 3.1 for a problem with two classes, where μ_{jc_y} and σ_{jc_y} are, respectively, the mean and the variance of the values of the feature j in the objects from class y. F1 outputs the maximum f among all features. This measure has an unbounded limit interval, since the values are in the interval $[0, \infty[$. For normalization matters, Lorena et al. (2019) (LORENA *et al.*, 2019) applied Equation 3.3, where M is the value of the measure. The implementation in the package ECoL (GARCIA; LORENA, 2018) also uses this value. Thus, we will also use it. This equation guarantees that the measured value is in the interval $]0, 1]$, whereby the more complex the dataset is, the higher the value.

$$M_{norm} = \frac{1}{M+1} \tag{3.3}$$

Since F1 relates two means and variances, it was not possible to adapt it and obtain similar information per class. Therefore, F1 is maintained in the experiments as initially proposed. We opted to include F1 in the analysis because its use is reported in previous papers dealing with imbalanced datasets, e.g., (LUENGO *et al.*, 2011; FERNÁNDEZ; JESUS; HERRERA, 2015; DÍEZ-PASTOR *et al.*, 2015).

The F1 values for the datasets in Figure 10 are shown in Table 1. F1 assessed that the imbalanced dataset is more difficult than the balanced one.

Table 1 – F1 values for the datasets in Figure 10

Dataset	F1
Balanced Dataset	0.58
Imbalanced Dataset	0.82

- **F2: Volume of overlap region**

F2 computes the volume of class overlapping regions, using the minimum and maximum values of each feature per class. It considers, for each feature, the range of possible values in which instances belonging to both classes can be found. It is calculated using Equation 3.4,

$$F2(T) = \prod_{i=1}^{l} \frac{max\{0, minmax(f_i) - maxmin(f_i)\}}{maxmax(f_i) - minmin(f_i)} \tag{3.4}$$

where:

$$minmax(f_i) = min(max(f_i^{c_0}), max(f_i^{c_1})) \tag{3.5}$$

$$maxmin(f_i) = max(min(f_i^{c_0}), min(f_i^{c_1})) \tag{3.6}$$

$$maxmax(f_i) = max(max(f_i^{c_0}), max(f_i^{c_1})) \tag{3.7}$$

$$minmin(f_i) = min(min(f_i^{c_0}), min(f_i^{c_1})) \tag{3.8}$$

The values $max(f_i^{c_j})$ and $min(f_i^{c_j})$ are the maximum and minimum values of feature f_i in a class c_j, respectively.

Thus, if the attribute ranges overlap in a region, this region is considered ambiguous regarding the attribute. Next, a product of the normalized size of the ambiguous regions for all attributes is output. As an example, suppose an attribute whose values for class 0 range between 0 and 1, and values for class 1 range between 0.75 and 1.25. The overlapping region for this attribute has size 0.25. Taking the full range of values for normalization, the final overlapping for this attribute is $\frac{0.25}{1.25} = 0.2$. F2 is zero if at least one of the attributes does not have any overlapping region and is equal to 1 when the classes are entirely overlapped for all attributes.

Classes may have different overlapping regions, and a single measure value may not represent the real complexity of the dataset, especially when the classes are imbalanced.

Considering the previous example, although half of the class 1 range is inside the ambiguous region, F2 evaluates that only 20% of the attribute's range represents the ambiguous region. F2 tends to underestimate the complexity of the dataset with the smallest range, which can undermine the proper assessment of the minority class complexity. The proposed adaptation considers the impact of the overlapping volume per class. The difference between the original F2 and the adaptation is the division of the size of the ambiguous region of each attribute by the range of value for the class of interest, instead of the range of all values of the attribute. This is illustrated by Equation 3.9 for class c_1. Considering the previous example, F2 for class 0 would be $\frac{0.25}{1} = 0.25$ and F2 for class 1 would be $\frac{0.25}{0.5} = 0.5$.

$$F2_{c_1}(T) = \prod_{i=1}^{l} \frac{max(0, minmax(f_i) - maxmin(f_i))}{max(f_i^{c_1}) - min(f_i^{c_1})} \tag{3.9}$$

The F2 values for the datasets in Figure 10 are shown in Table 2. The original F2 assessed that the balanced and the imbalanced dataset with similar complexity. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 2 – F2 values for the datasets in Figure 10

Dataset	**Original F2**	**Negative class F2**	**Positive class F2**
Balanced Dataset	0.33	0.52	0.56
Imbalanced Dataset	0.34	0.43	0.74

- **F3: Feature efficiency**

In F3, the number of instances inside the ambiguous region defines the inefficiency of a feature. The greater the amount of instances inside the ambiguous region, the more inefficient this feature is in separating the classes. Equation 3.10 illustrates how F3 is calculated.

$$F3(T) = \min_{i=1}^{m} \frac{n_o(f_i)}{n} \tag{3.10}$$

In this equation, $n_o(f_i)$ returns the number of instances in the overlapping region for f_i, whose value is defined by:

$$n_o(f_i) = \sum_{j=1}^{n} I(x_{ji} > maxmin(f_i) \wedge x_{ji} < minmax(f_i)) \tag{3.11}$$

where I is the indicator function that returns 1 if its argument is true and 0, otherwise.

Similar to F2, F3 has a bias towards the majority class, since the whole minority class can be inside the ambiguous region and F3 can still be close to 1. The adaptation of F3 considers one class at a time. The F3 per class divides the number of instances from that class of interest inside the ambiguous region by the number of instances from the class only. In our adaptation, Equations 3.10 and 3.11 are changed to Equations 3.12 and 3.13 respectively, where n_{c_1} is the number of instances from class c_1 and $x_{ji}^{c_1}$ is the value of the j-th attribute from the i-th instance of class c_1.

$$F3_{c_1}(T) = \min_{i=1}^{m} \frac{n_o^{c_1}(f_i)}{n_{c_1}} \tag{3.12}$$

$$n_o^{c_1}(f_i) = \sum_{j=1}^{n_{c_1}} I(x_{ji}^{c_1} > maxmin(f_i) \wedge x_{ji}^{c_1} < minmax(f_i)) \tag{3.13}$$

The F3 values for the datasets in Figure 10 are shown in Table 3. The original F3 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 3 – F3 values for the datasets in Figure 10

Dataset	Original F3	Negative class F3	Positive class F3
Balanced Dataset	0.92	0.89	0.90
Imbalanced Dataset	0.83	0.81	0.94

- **F4: Collective feature efficiency**

F4 is similar to F3, but instead of using the minimum value from all attributes, it combines their discrimination power. The proportion of instances remaining, after using all features to discriminate them, is the outcome of F4. For this purpose, first, it finds the most discriminative attribute according to $\underset{i=1}{\overset{m}{argmin}} \frac{n_o(f_i)}{n}$; next, it removes the instances correctly discriminated by this attribute. It repeats the previous steps until all instances are correctly discriminated or until all attributes are removed. F4 is equal to the proportion of instances not discriminated at the end of the process. Equation 3.14 illustrates how F4 is calculated, where T_l is the dataset of the l-th iteration (with l in interval $[1,m]$) and $n_o(f_{min}(T_l))$ measures the number of instances in the overlapping region of attribute f_{min} from dataset T_l.

$$F4(T) = \frac{n_o(f_{min}(T_l))}{n} \tag{3.14}$$

Considering any i-th iteration of F4, the most discriminative attribute (f_{max}) of dataset T_i can be found using Equation 3.15, where $n_o(f_j)$ is computed according to Equation 3.11.

The dataset of each iteration can be defined by Equations 3.16 and 3.17. Thus, the dataset at the i-th iteration is a subset of the previous dataset (T_{i-1}), considering only the instances inside the overlapping region of f_{min}.

$$f_{min}(T_i) = \{f_j | \min_{j=1}^{m}(n_o(f_j))\}_{T_i} \tag{3.15}$$

$$T_1 = T \tag{3.16}$$

$$T_i = T_{i-1} - \{\mathbf{x}_j | x_{ji} < maxmin(f_{min}(T_{i-1})) \vee x_{ji} > minmax(f_{min}(T_{i-1}))\} \tag{3.17}$$

Our adaptation of F4 computes the number of misclassified instances in each class divided by the number of instances from that class, i.e., we adapted F4 to calculate the complexity of a class c_1. For such, Equations 3.14, 3.15 and 3.17 must be substituted, respectively, by Equations 3.18, 3.19 and 3.20, where $n_o^{c_1}$ is defined in Equation 3.13.

$$F4_{c_1}(T) = \frac{n_o^{c_1}(f_{min}^{c_1}(T_l))}{n_{c_1}} \tag{3.18}$$

$$f_{min}^{c_1}(T_i) = \{f_j | \min_{j=1}^{m}(n_o^{c_1}(f_j))\}_{T_i} \tag{3.19}$$

$$T_i = T_{i-1} - \{\mathbf{x}_j | x_{ji} < maxmin(f_{min}^{c_1}(T_{i-1})) \vee x_{ji} > minmax(f_{min}^{c_1}(T_{i-1}))\} \tag{3.20}$$

The F4 values for the datasets in Figure 10 are shown in Table 4. The original F4 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 4 – F4 values for the datasets in Figure 10

Dataset	Original F4	Negative class F4	Positive class F4
Balanced Dataset	0.87	0.89	0.90
Imbalanced Dataset	0.71	0.81	0.94

3.2.1.2 Neighborhood measures

The neighborhood measures use the concept of Nearest Neighbor (NN) to assess classification difficulties. They use the distance between instances to assess, for example, the shape of decision boundaries and class distributions. In this paper, we considered the measures N1, N2, N3, N4, and T1. A description of the original data complexity measures and an explanation of our adaptations are presented next.

- **N1: Fraction of points on the class boundary**

  N1 builds a minimum spanning tree (MST) that connects all instances in a dataset based on their pairwise distances, despite their classes. Next, it counts the number of instances connected to at least one instance from another class. These instances are possibly borderline and the ratio between their number and the total number of instances is the final N1 measure. N1 is bounded between 0 and 1, the closer to 0, the lower the complexity. Equation 3.21 expresses N1, where $(\mathbf{x}_i, \mathbf{x}_j)$ represents a connection between instances $\mathbf{x}_i$ and $\mathbf{x}_j$ and MST represents the set of all connections in the tree.

$$N1(T) = \frac{1}{n}\sum_{i=1}^{n} I((\mathbf{x}_i, \mathbf{x}_j) \in MST \wedge y_i \neq y_j) \tag{3.21}$$

  N1 has a bias towards the majority class since the use of the normalization factor n leads to underestimation of the minority class complexity as the imbalance aggravates. Our adaptation considers each class separately. For such, considering one class at a time, we calculate the proportion of instances from that class that connects with an instance from a different class. With this adaptation, we can measure how complex a class is considering the concept of the N1. Equation 3.22 shows the adaptation, where $\mathbf{x}_i^{c1}$ denotes an example of class c_1.

$$N1_{c_1}(T) = \frac{1}{n_{c_1}}\sum_{i=1}^{n_{c_1}} I((\mathbf{x}_i^{c_1}, \mathbf{x}_j) \in MST \wedge y_j \neq c_1) \tag{3.22}$$

  The N1 values for the datasets in Figure 10 are shown in Table 5. The original N1 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 5 – N1 values for the datasets in Figure 10

Dataset	Original N1	Negative class N1	Positive class N1
Balanced Dataset	0.25	0.26	0.24
Imbalanced Dataset	0.13	0.08	0.64

- **N2: Ratio of average intra/inter class NN distance**

  N2 compares the intraclass and interclass dispersions of the classes. For each instance, its distance to the NN of the same class (intraclass) and its distance to the NN of a different class (interclass) are computed. N2 is the ratio of the intraclass distance average and the interclass distance average. Higher values represent problems of higher complexity. Equation 3.23 shows how N2 is calculated. In this equation, $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance function between $\mathbf{x}_i$ and $\mathbf{x}_j$, $NN(\mathbf{x}_i) \in \{T|y = y_i\}$ is the nearest neighbor of $\mathbf{x}_i$ from the same class and $NN(\mathbf{x}_i) \in \{T|y \neq y_i\}$ is the nearest neighbor of $\mathbf{x}_i$ from a different class.

$$N2(T) = \frac{\sum_{i=1}^{n} d(\mathbf{x}_i, NN(\mathbf{x}_i) \in \{T|y = y_i\})}{\sum_{i=1}^{n} d(\mathbf{x}_i, NN(\mathbf{x}_i) \in \{T|y \neq y_i\})} \tag{3.23}$$

By taking the averages of all instances, N2 values are biased towards the majority class. Our adaptation takes the averages for one class at a time. Therefore, the N2 value for a specific class, for example a class 1, will be the ratio of two averages: the average of intraclass distances for class 1 (i.e., the distance between each instance from class 1 and its NN from also class 1) and the average of the interclass distances for class 1 (i.e., the distance between each instance of class 1 with its NN from a different class). Equation 3.24 shows the modified N2, where $\mathbf{x}_i^{c1}$ denotes an example of class c_1.

$$N2_{c_1}(T) = \frac{\sum_{i=1}^{n_{c_1}} d(\mathbf{x}_i^{c1}, NN(\mathbf{x}_i^{c1}) \in \{T|y = c_1\})}{\sum_{i=1}^{n_{c_1}} d(\mathbf{x}_i^{c1}, NN(\mathbf{x}_i^{c1}) \in \{T|y \neq c_1\})} \tag{3.24}$$

The N2 values for the datasets in Figure 10 are shown in Table 6. The original N2 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 6 – N2 values for the datasets in Figure 10

Dataset	**Original N2**	**Negative class N2**	**Positive class N2**
Balanced Dataset	0.18	0.18	0.19
Imbalanced Dataset	0.13	0.11	0.44

- **N3: Leave-one-out error rate of the NN classifier**

N3 is the ratio between the number of examples whose NN are from a different class and the total number of examples from T. It is the same concept of the leave-one-out error of a NN classifier, which is easy to calculate and is a good indicator of the separability of classes. The following equation expresses how N3 is defined:

$$N3(T) = \frac{\sum_{i=1}^{n} I(NN(\mathbf{x}_i) \neq y_i)}{n} \tag{3.25}$$

When a dataset is highly imbalanced, N3 tends to be closer to the majority class error, which can be inadequate. To overcome this problem, we adapted N3 to take into account the error per class, i.e., the ratio between the number of examples from the class of interest whose NN are from a different class and the number of examples from that class. Equation 3.26 represents our adaptation in which c_1 represents the class of interest.

$$N3_{c_1}(T) = \frac{\sum_{i=1}^{n_{c_1}} I(NN(\mathbf{x}_i^{c_1}) \neq c_1)}{n_{c_1}} \tag{3.26}$$

The N3 values for the datasets in Figure 10 are shown in Table 7. The original N3 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 7 – N3 values for the datasets in Figure 10

Dataset	**Original N3**	**Negative class N3**	**Positive class N3**
Balanced Dataset	0.17	0.18	0.17
Imbalanced Dataset	0.10	0.05	0.52

- **N4: Nonlinearity of a 1-NN classifier**

N4 uses a method that creates a new test set by interpolating randomly selected instances from the same class. Next, an NN classifier uses training set T to predict the labels of the instances in the interpolated test set. N4 returns the error rate obtained. A value closer to 1 may indicate either overlapped classes or that the classes do not form convex sets. Equation 3.27 shows how to calculate F4, where l is the number of interpolated instances, $\mathbf{x}_i'$ is an interpolated instance, $NN_T(\mathbf{x}_i')$ is the NN from T to $\mathbf{x}_i'$ and y_i' is the class of $\mathbf{x}_i'$.

$$N4(T) = \frac{1}{l}\sum_{i=1}^{l} I(NN_T(\mathbf{x}_i') \neq y_i') \tag{3.27}$$

Using the same criterion as in N3, N4 was adapted to return the error rate per class. Thus, an NN classifier using the dataset T labels each interpolated instance $\mathbf{x}_i'$ from the class of interest c_1. The error rate is used as a measure. Considering a c_1 as the class of interest, Equation 3.28 represents our adaptation for F4. In this adaptation, l_{c_1} is the number of interpolated instances from class c_1 and $\mathbf{x}_i^{c_1'}$ is an interpolated example from class c_1.

$$N4_{c_1}(T) = \frac{1}{l_{c_1}}\sum_{i=1}^{l_{c_1}} I(NN_T(\mathbf{x}_i^{c_1'}) \neq c_1) \tag{3.28}$$

The N4 values for the datasets in Figure 10 are shown in Table 8. The original N4 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 8 – N4 values for the datasets in Figure 10

Dataset	Original N4	Negative class N4	Positive class N4
Balanced Dataset	0.13	0.13	0.13
Imbalanced Dataset	0.06	0.03	0.42

- **T1: Fraction of maximum covering spheres**

T1 looks for an interpretation of a training set using hyper-spheres. To explain how it works, suppose that every instance in the training set has a hypersphere with radius zero. If we gradually increase the radius of all hyperspheres, some of them will touch a hypersphere from a different class. When this occurs, both hyperspheres stop expanding. The method finishes when there is no more expanding hypersphere, discarding the hyperspheres contained in another hypersphere. T1 is the ratio between the number of remaining hyperspheres and the number of instances in the dataset. A number closer to 0 indicates that there is no need for many hyperspheres to describe the training set. A number closer to 1 indicates a higher complexity and that as many hyperspheres as number of instances are needed to describe the training set. Equation 3.29 represents T1, where $Hyperspheres(T)$ calculates the number of hyperspheres needed to cover the dataset.

$$T1(T) = \frac{Hyperspheres(T)}{n} \tag{3.29}$$

Consider a binary training set entirely overlapped and highly imbalanced, T1 may be low for this training set, since a small number of hyperspheres is needed to describe the data compared to the number of instances. However, to describe the minority class we need almost the same number of minority class instances as hyperspheres. Therefore, our adaptation of T1 takes the ratio between the hyperspheres necessary to describe each class and the number of instances in the class. Equation 3.30 substitutes Equation 3.29 in our definition, when $Hyperspheres(T, c_1)$ calculates the number of hyperspheres needed to cover the examples of class c_1.

$$T1_{c_1}(T) = \frac{Hyperspheres(T, c_1)}{n_{c_1}} \tag{3.30}$$

The T1 values for the datasets in Figure 10 are shown in Table 9. The original T1 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 9 – T1 values for the datasets in Figure 10

Dataset	**Original T1**	**Negative class T1**	**Positive class T1**
Balanced Dataset	0.26	0.27	0.26
Imbalanced Dataset	0.13	0.08	0.64

3.2.1.3 Linear Separability Measures

These measures assess whether the classes can be linearly separable in the attribute space. They assume that a classification problem solved with a hyperplane is simpler than another with a non-linear boundary. The measures from this category considered in this article are L1, L2, and L3.

To build the linear classifier for the complexity measures, Ho and Basu (2002) (HO; BASU, 2002) suggest solving the optimization problem proposed by Smith (1968) (SMITH, 1968). Recent studies propose the using a Support Vector Machine (SVM) with a linear kernel (ORRIOLS-PUIG; MACIá; HO, 2010; LORENA *et al.*, 2019). SVM obtains the hyperplane by solving the following optimization problem:

$$\underset{w,b,\varepsilon}{Minimize} \quad \frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_{i=1}^{n} \varepsilon_i\right) \tag{3.31}$$

$$Subject\,to: \begin{cases} y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \varepsilon_i, \\ \quad \varepsilon_i \geq 0, i = 1,\dots,n \end{cases} \tag{3.32}$$

where C is the trade-off between the margin maximization, achieved by minimizing the norm of $\mathbf{w}$, and the minimization of the training errors, modeled by ε. The hyperplane is given by $\mathbf{w}\cdot\mathbf{x}+b=0$, where $\mathbf{w}$ is a weight vector and b is an offset value. All the linearity measures described in this article will adopt this notation. Next, we describe the measures investigated in this study.

- **L1: Minimized sum of error distance of a linear classifier**

  L1 uses a linear model (e.g., a linear SVM) induced by a training set and the distances between misclassified instances and a hyperplane representing the model. L1 returns the average of these distances, which is equal to 0 for linearly separable problems.

  Considering the SVM hyperplane, L1 can be calculated using all ε_i, as shown in Equation 3.33. We normalize L1 to the interval $[0,1]$, whereby the larger the value, the more complex the dataset, using $1-\frac{1}{L1+1}$.

$$L1(T) = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i | h(\mathbf{x}_i) \neq y_i \tag{3.33}$$

where $h(\mathbf{x}_i)$ represents the SVM prediction for the i-th training example.

As L1 has a bias towards the majority class, we adapt it so that only the distances of misclassified instances from each specific class are summed up, as shown in Equation 3.34:

$$L1_{c_1}(T) = \frac{1}{n_{c_1}} \sum_{i=1}^{n_{c_1}} \varepsilon_i^{c_1} | \mathbf{h}(\mathbf{x}_i) \neq c_1 \tag{3.34}$$

The L1 values for the datasets in Figure 10 are shown in Table 10. The original L1 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 10 – L1 values for the datasets in Figure 10

Dataset	**Original L1**	**Negative class L1**	**Positive class L1**
Balanced Dataset	0.08	0.08	0.09
Imbalanced Dataset	0.05	0.00	0.33

- **L2: Training error of a linear classifier**

  L2 is the training error of a linear classifier. For its calculation, we induce a linear classifier from the training set and use its classification error rate. The higher the values the less linear is the classification boundary. Equation 3.35 shows how L2 is calculated. In this equation, $h(\mathbf{x}_i)$ is the predicted class for the instance $\mathbf{x}_i$.

$$L2(T) = \frac{\sum_{i=1}^{n} I(h(\mathbf{x}_i) \neq y_i)}{n} \tag{3.35}$$

  Our adaptation returns the error rate per class, using Equation 3.36:

$$L2_{c_1}(T) = \frac{\sum_{i=1}^{n_{c_1}} I(h(\mathbf{x}_i^{c_1}) \neq c_1)}{n_{c_1}} \tag{3.36}$$

  The L2 values for the datasets in Figure 10 are shown in Table 11. The original L2 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

- **L3: Nonlinearity of the linear classifier**

  Similar to N4, L3 interpolates a test set and, instead of a KNN classifier, uses a linear classifier to classify instances from the test set. Equation 3.37 shows how L3 is calculated.

Table 11 – L2 values for the datasets in Figure 10

Dataset	**Original L2**	**Negative class L2**	**Positive class L2**
Balanced Dataset	0.11	0.11	0.12
Imbalanced Dataset	0.05	0.01	0.44

In this equation, $h_T(\mathbf{x}_i')$ is the prediction of the linear model induced using training set T for the interpolated instance $\mathbf{x}_i'$.

$$L3(T) = \frac{1}{l}\sum_{i=1}^{l} I(h_T(\mathbf{x}_i') \neq y_i') \tag{3.37}$$

Our adaptation returns the error rate per class, using Equation 3.38:

$$L3_{c_1}(T) = \frac{1}{l_{c_1}}\sum_{i=1}^{l_{c_1}} I(h_T(\mathbf{x}_i^{c_1\prime}) \neq c_1) \tag{3.38}$$

The L3 values for the datasets in Figure 10 are shown in Table 12. The original L3 assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 12 – L3 values for the datasets in Figure 10

Dataset	**Original L3**	**Negative class L3**	**Positive class L3**
Balanced Dataset	0.08	0.06	0.07
Imbalanced Dataset	0.04	0.00	0.46

3.2.1.4 Other Complexity Measures for Imbalanced Datasets

Recently, four other data complexity measures were proposed specifically for imbalanced datasets (ANWAR; JONES; GANESH, 2014; SINGH; GOSAIN; SAHA, 2020; LU; CHEUNG; TANG, 2019). They are CM, wCM, dwCM, and BI^3. All of them use a kNN classifier in their calculation. In the experimental analysis, we compare them with our adaptations on the original data complexity measures. Next, we describe these four measures.

- **CM: Complexity measure for imbalanced datasets**

  CM considers the k nearest neighbors of each minority class instance (ANWAR; JONES; GANESH, 2014). If the majority of the k nearest neighbors does not belong to the minority class, this instance is considered difficult. CM is the percentage of difficult minority class instances. Equation 3.39 shows how CM is calculated, considering c_1 as the minority class,

k as a parameter defined by the user, and $NN_j(\mathbf{x}_i^{c_1})$ as the j-th nearest neighbor of instance $\mathbf{x}_i^{c_1}$.

$$CM(T,k) = \frac{1}{n_{c_1}} \sum_{i=1}^{n_{c_i}} I\left(\frac{\sum_{j=1}^{k} I(NN_j(\mathbf{x}_i^{c_1}) \neq c_1)}{k} > 0.5\right) \tag{3.39}$$

The CM values for the datasets in Figure 10 are shown in Table 13. We used the CM for the whole dataset and a k optimization defined by Anwar, Jones and Ganesh. The CM for the whole dataset assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 13 – CM values for the datasets in Figure 10

Dataset	**Dataset CM**	**Negative class CM**	**Positive class CM**
Balanced Dataset	0.14	0.13	0.15
Imbalanced Dataset	0.07	0.03	0.49

- **wCM: Weighted complexity metric**

  wCM extends CM using a distance weighted kNN classifier instead of a kNN (SINGH; GOSAIN; SAHA, 2020). On this measure, each neighbor j of each instance i from the minority class has a weight defined by their distance. The weights are normalized using the distances of the closest neighbor and the farthest neighbor. The calculation of W_{ij}, which is the weight of the j-th neighbor of the i-th minority instance is defined by Equation 3.40. wCM then uses the weights on its calculation, as defined by Equation 3.41.

$$W_{ij} = \begin{cases} \dfrac{d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) - d(\mathbf{x}_i, NN_j(\mathbf{x}_i))}{d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) - d(\mathbf{x}_i, NN_1(\mathbf{x}_i))}, \text{if } d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) \neq d(\mathbf{x}_i, NN_1(\mathbf{x}_i)) \\ 1, \text{if } d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) = d(\mathbf{x}_i, NN_1(\mathbf{x}_i)) \end{cases} \tag{3.40}$$

$$wCM(T,k) = \frac{1}{n_{c_1}} \sum_{i=1}^{n_{c_i}} I\left(\frac{\sum_{j=1}^{k} W_{ij} I(NN_j(\mathbf{x}_i^{c_1}) \neq c_1)}{\sum_{j=1}^{k} W_{ij}} > 0.5\right) \tag{3.41}$$

  The wCM values for the datasets in Figure 10 are shown in Table 14. We used the wCM for the whole dataset and $k = 11$ as suggested in Singh, Gosain and Saha. The CM for the whole dataset assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

- **dwCM: Dual weighted complexity metric**

  According to the authors, wCM may not be robust enough depending on the value of k, and therefore they also propose a dual weighted complexity metric, the dwCM (SINGH;

Table 14 – wCM values for the datasets in Figure 10

Dataset	Dataset wCM	Negative class wCM	Positive class wCM
Balanced Dataset	0.13	0.12	0.14
Imbalanced Dataset	0.06	0.02	0.5

GOSAIN; SAHA, 2020). The difference between wCM and dwCM are the weights. In dwCM, the weights are calculated according to the Equation 3.42.

$$W_{ij} = \begin{cases} \dfrac{d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) - d(\mathbf{x}_i, NN_j(\mathbf{x}_i))}{d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) - d(\mathbf{x}_i, NN_1(\mathbf{x}_i))} \times \dfrac{d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) + d(\mathbf{x}_i, NN_1(\mathbf{x}_i))}{d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) + d(\mathbf{x}_i, NN_j(\mathbf{x}_i))}, \\ \qquad \text{if } d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) \neq d(\mathbf{x}_i, NN_1(\mathbf{x}_i)) \\ 1, \text{if } d(\mathbf{x}_i, NN_k(\mathbf{x}_i)) = d(\mathbf{x}_i, NN_1(\mathbf{x}_i)) \end{cases} \tag{3.42}$$

The dwCM values for the datasets in Figure 10 are shown in Table 15. We used the dwCM for the whole dataset and $k = 11$ as suggested in Singh, Gosain and Saha. The CM for the whole dataset assessed that the imbalanced dataset is easier than the balanced one. The decomposed measures assessed that the imbalanced dataset is easier for the majority class and more difficult for the minority class.

Table 15 – wCM values for the datasets in Figure 10

Dataset	Dataset dwCM	Negative class dwCM	Positive class dwCM
Balanced Dataset	0.14	0.12	0.15
Imbalanced Dataset	0.06	0.02	0.48

- **BI^3: Bayes imbalance impact index**

  Inspired by the Bayes optimal classifier, Lu, Cheung and Tang proposes a measure called Bayes Imbalance Impact Index (BI^3). It is calculated according to Equation 3.43, where $f_n(\mathbf{x}_i, k) = \frac{\sum_{j=1}^{k} I(NN_j(\mathbf{x}_i) \neq c_1)}{k}$, $f_p(\mathbf{x}_i, k) = \frac{\sum_{j=1}^{k} I(NN_j(\mathbf{x}_i) = c_1)}{k}$, and $f'_p(\mathbf{x}_i, k) = \frac{n_{c_0}}{n_{c_1}} \times f_p(\mathbf{x}_i, k)$.

$$BI^3(T,k) = \frac{1}{n_{c_1}} \sum_{i=1}^{n_{c_i}} \frac{f'_p(\mathbf{x}_i, k)}{f_n(\mathbf{x}_i, k) + f'_p(\mathbf{x}_i, k)} - \frac{f_p(\mathbf{x}_i, k)}{f_n(\mathbf{x}_i, k) + f_p(\mathbf{x}_i, k)} \tag{3.43}$$

  The BI^3 values for the datasets in Figure 10 are shown in Table 16. We used $k = 5$ as suggested by Lu, Cheung and Tang. BI^3 assessed that the imbalanced dataset is more difficult than the balanced one.

All four measures described above are parameter dependent. The user must set the parameter k, and its choice may change the outcome of the measure. For example, Singh, Gosain

Table 16 – BI^3 values for the datasets in Figure 10

Dataset	BI^3
Balanced Dataset	0.00
Imbalanced Dataset	0.29

and Saha reported that CM and wCM may be sensitive to the parameter choice. CM proposes a strategy to choose k. The other three fix a value for the parameter. BI^3 presents a strategy of flexible k to avoid 0 values on its calculation. They also do not compare their results with N3 - which has a similar concept in terms of assessing the data complexity. In this paper, we evaluate the related work aforementioned not only with N3, but all our proposed adaptations on the data complexity measures.

In this work, we consider only these data complexity measures because they are the most used, studied and have different biases. We also consider measures proposed specifically for imbalanced datasets. Nevertheless, there are other complexity measures that were not described (KOLACZYK, 2009; SMITH; MARTINEZ; GIRAUD-CARRIER, 2014). For example, measures extracted from a structural representation of the dataset using graphs, which take into account the relationship between instances (KOLACZYK, 2009). In Smith, Martinez and Giraud-Carrier, a subset of measures that extract instance hardness is proposed, i.e. considering an instance as hard if it is misclassified by a diverse set of simple classification algorithms.

3.2.2 Pre-processing techniques for imbalanced classification tasks

There are two main approaches to deal with imbalanced data classification tasks: (1) pre-processing the data to make it more balanced (CHAWLA *et al.*, 2002; HAN; WANG; MAO, 2005; HE *et al.*, 2008; JO; JAPKOWICZ, 2004; KUBAT; MATWIN *et al.*, 1997); (2) developing classification algorithms which are more robust to imbalanced data (GONZALEZ-ABRIL *et al.*, 2014; CIESLAK *et al.*, 2012; CANO; ZAFRA; VENTURA, 2013; DIAMANTINI; POTENA, 2009). Pre-processing techniques are usually independent from classification algorithms. However, they may modify the original data distribution, removing important instances or adding noise (HE; GARCIA, 2008). Adapted classification algorithms reduce Data Mining pipelines, but do not improve data quality. In this paper, we focus on the former, which is more often adopted.

Pre-processing techniques are used based on data undersampling and/or oversampling (FERNáNDEZ *et al.*, 2018). To balance the data, undersampling techniques remove instances from the majority class and oversampling techniques insert instances in the minority class (FERNáNDEZ *et al.*, 2018). Both undersampling and oversampling can occur randomly or based on some criteria. Next, we discuss some of the main pre-processing strategies for balancing datasets in ML.

3.2.2.1 Random sampling

Random undersampling (RU) removes instances from the majority class at random until obtaining a data distribution considered balanced (HE; GARCIA, 2008). Random oversampling (RO) replicates instances from the minority class at random until obtaining a data distribution considered balanced (HE; GARCIA, 2008). In the literature, they are usually used until classes are equally represented in the number of instances.

3.2.2.2 Synthetic minority oversampling techniques

SMOTE (*Synthetic Minority Oversampling Technique*) (CHAWLA *et al.*, 2002) is an oversampling technique that creates artificial data by interpolation, as follows. At each iteration, SMOTE selects at random an instance **x** from the minority class. Next, it uses KNN to find the k closest instances to **x** in the minority class. It selects one of the neighbors **z** at random and creates a new instance that is a combination of **x** and **z**. The combination is an interpolation that randomly creates any possible point between **x** and **z**. This step is repeated until a distribution of instances considered balanced is obtained.

BorderlineSMOTE is a version of SMOTE that searches for minority class instances close to decision boundaries to interpolate (HAN; WANG; MAO, 2005). Instead of selecting minority class instances from all training sets, it selects minority class instances close to the decision boundary. The procedure that BorderlineSMOTE uses to select them is: (1) find the k NN for a minority class instance **x**; (2) count the number N_{maj} of neighbors that belongs to the majority class; (3) if $\frac{k}{2} \leq N_{maj} < k$ then **x** is put in a set called DANGER; (4) repeat the steps for all minority class instances. Afterwards SMOTE is run to balance the dataset but it selects only instances from the DANGER subset.

ADASYN, also based on SMOTE, addresses the number of instances to be interpolated by each minority class instance (HE *et al.*, 2008). For such, it follows three steps: (1) it defines G, which indicates how many instances should be interpolated for the entire minority class; (2) for each instance in the minority class, it calculates the percentage of majority class instances in the k nearest neighbors; (3) it normalizes the set of all percentages (Γ_i, where i is the minority class instance), so that $\sum \Gamma_i = 1$; finally, $\Gamma_i \times G$ gives the number of instances to be interpolated using SMOTE for each minority class instance i.

3.2.2.3 Cluster based oversampling

CBO (*Cluster-Based Oversampling*) (JO; JAPKOWICZ, 2004) is an oversampling technique that takes into account both inter and intraclass imbalance. Differently from the inter and intraclass distance defined in Section 3.2.1, inter and intraclass imbalance considers the disproportion between classes and inside a class, respectively. Interclass imbalance is the concept commonly used to describe a disproportion between classes in number of instances. The intraclass imbalance describes the disproportion inside a class, i.e when the subconcepts of the same class have a disproportion between them.

For such, CBO first applies, separately, a clustering algorithm to the instances from the majority class and to the instances from the minority class, generating two sets of clusters - one for each class. Next, *CBO* oversamples all clusters belonging to the majority class, except the largest cluster. In the end, each cluster of the majority class should have the same number of instances as the largest cluster. Finally, oversampling is applied to all clusters belonging to the minority class, making (1) the number of instances in the minority class equal to the number of instances in the majority class after oversampling, and (2) each cluster in the minority class equally balanced.

3.2.2.4 One-sided selection

OSS (*One-sided Selection*) (KUBAT; MATWIN *et al.*, 1997) is an undersampling technique that keeps only the most representative instances of the majority class. For such, *OSS* initially chooses one instance **x** of the majority class at random. Next, using the instances of the minority class and **x** as training data, *OSS* applies the *k*-Nearest Neighbors (*KNN*) algorithm with $k = 1$ to classify the remaining instances of the majority class. The correctly classified instances are excluded from the majority class, as they are considered redundant. Thus, after the undersampling, the majority class will have only the instances that were incorrectly classified by $k-$NN and **x**. Finally, *OSS* uses a data cleaning technique to remove borderline and noisy instances, originally, *Tomek Links* (TOMEK, 1976).

All techniques modify the values of the data complexity measures described in Section 3.2.1. For example, SMOTE modifies the neighborhood measures by generating new instances near existing ones. More specifically, the N3 measure may be reduced by generating samples near overlapping decision borders; and OSS may modify overlapping measures, such as F2, when it reduces the range of the values considered by the measures.

In the same way that, according to their bias, pre-processing techniques modify the complexity measures, these techniques can artificially modify the complexity measure values. For example, since N3 is based on NN, the duplication of instances in the training set by RO decreases the N3 value. In an extreme case, when all minority class instances are duplicated, the N3 value for this class would become 0, but the predictive performance of a classifier using the new dataset would not improve.

New pre-processing techniques for imbalanced classification have been recently proposed, including other SMOTE adaptations (BARUA *et al.*, 2014; ABDI; HASHEMI, 2016), undersampling based on clustering (NG *et al.*, 2015), and sampling based on evolutionary algorithms (YU; NI; ZHAO, 2013). According to Barua *et al.*, SMOTE adaptations favored noisy instances. To overcome this problem, the authors proposed a new approach to select minority class instances that discard those with no minority neighbor. This SMOTE adaptation, called MWMOTE (Majority Weighted Minority Oversampling Technique), weights the minority class instances and generates new instances within a minority cluster. MDO (Mahalanobis Distance-based Over-sampling technique) (ABDI; HASHEMI, 2016), another SMOTE adaptation, generates synthetic minority class instances that have the same Mahalanobis distance to the class mean as other existing minority class instances. In this article, we use the standard pre-processing techniques described in Section 3.2.2 because they are the most used and studied.

Data complexity measures have also been used to tackle the imbalance problem. Luengo *et al.* used complexity measures to predict whether a DIT technique would be useful. They found intervals of values for some complexity measures in which the techniques were useful. Complexity measures have also been used to analyze the suitability of using a specific DIT technique. Díez-Pastor *et al.* used complexity measures to predict data complexity intervals in which some diversity-enhancing techniques may improve the results of an ensemble of classifiers. Fernández, Jesus and Herrera used one complexity measure combined with other characteristics (such as imbalance) in a multi-objective approach to select attributes and instances from an imbalanced dataset. Fernandes and Carvalho adapted the N1 measure to the context of imbalanced multi-class classification and used it in a multi-objective approach as undersampling.

To the best of our knowledge, no work in the literature has analyzed the data complexity measures regarding the imbalance problem on real datasets by decomposing the original measures per class. All the works previously mentioned use the original data complexity measures. Next, we show experimentally that the traditional complexity measures do not capture complexity in imbalanced datasets properly. Therefore, the contributions of the aforementioned studies can be improved by using our adaptations.

3.3 Experimental Settings

The contributions of this study are guided by the following research questions: Are the original data complexity measures suitable for imbalanced datasets? Does a decomposition by class improve their performance on imbalanced datasets? Is there a correlation between the difference in data complexity and the difference in predictive performance after applying DITs?

To answer these questions, we performed an extensive empirical analysis, using 203 datasets, which were randomly divided into two groups. We use the first group of datasets to evaluate the performance of the data complexity measures on assessing imbalanced datasets. From these results, we select the most relevant complexity measures for the studied cases. Next, we use the selected measures to analyze the complexity after applying DITs. We collected the datasets from OpenML (VANSCHOREN *et al.*, 2013) and made them available, together with the experiment results[1]. We also implemented a package for the adapted data complexity measures,

[1] <https://github.com/victorhb/IS2020_results>

called ImbCoL [2].

3.3.1 Data Complexity Measures Experiments

In these experiments, we used a group of 102 datasets to investigate whether the original data complexity measures can assess how difficult an imbalanced classification dataset is. The predictive performance of a classification model was used to estimate the difficulty of a dataset using a grid search approach. Table 17 shows a summary of the 102 datasets used in this experiment. Minimum, maximum and mean values for the number of instances, number of features and percentage of the minority class are shown. For more details, please see Table 24 in the Appendices or the GitHub link[3]. 33 out of the 102 datasets have less than 25% of minority class instances. We call them the high imbalanced datasets. The remaining 69 ones are called the low imbalanced datasets.

Table 17 – Summary of the 102 datasets used on the experiment to evaluate the data complexity measures.

Dataset Characteristic	**Min Value**	**Max Value**	**Mean Value**
Number of Instances	36	2,534	486
Number of Features	3	95	16
% Minority Class	2.15	49.70	32.27

To reduce the influence of the bias of the ML algorithm, we used a pool of six algorithms, which were tuned using grid search. The hyperparameters and their possible values are listed in Table 18, in which m is the number of attributes and $a = \frac{(m+2)}{2}$. We considered all data complexity measures described in Section 3.2.1.

Table 18 – Classification algorithms used and their possible hyperparameter values

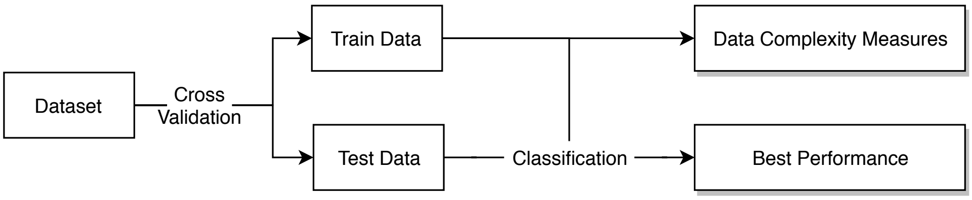
Classification Algorithms	**Hyperparameters**	**Values**
Support Vector Machines (SVM)	kernel	linear, radial, polynomial, sigmoidal
	cost	$2^{-10}, 2^{-9}, ..., 2^{10}$
	gamma	$2^{-10}, 2^{-9}, ..., 2^{10}$
	degree	2, 3, 4, 5
Random Forest (RF)	number of trees	100, 200, ..., 1000
	number of variables	$\frac{\sqrt{m}}{2}, \sqrt{m}, \sqrt{m} \times 2$
K-Nearest Neighbours (KNN)	k	1, 3, 5, ..., 31
Naive Bayes (NB)	None	None
C4.5	threshold for pruning	0.1, 0.2, ..., 0.5
	min instances per leaf	2, 3, ..., 10
Multi-Layer Perceptron Neural Networks (MLP)	learning rate	0.1, 0.2, ..., 1
	number of neurons in hidden layer	$a-3, a-2, ..., a+3$

[2] <https://github.com/victorhb/ImbCoL>

[3] <https://github.com/victorhb/IS2020_results>

Using 30 repetitions of stratified 5-fold cross-validation, we extracted, for each dataset, 150 sets of data complexity measures from the training subsets and 150 sets of predictive performances from the validation subsets. Next, we assigned, to each dataset, the mean of the 150 values of each complexity measure and the predictive performance. The best model was chosen according to the highest predictive performance on average for each dataset. Figure 11 illustrates the steps followed in these experiments. Afterwards, we correlated, for each dataset, the mean data complexity measures with the mean predictive performance.

Figure 11 – Diagram illustrating the steps of the first experiment



Source: Barella *et al.* (2020).

We measured the predictive performance using gmean, which is widely used in the imbalanced data literature. Gmean is the geometric mean between the true positive rate (TPR) and the true negative rate (TNR), defined by $gmean = \sqrt{\mathrm{TPR} \times \mathrm{TNR}}$. We used the Pearson correlation to correlate the data complexity measures and the Gmean. For the complexity measures dependent of k, we set the parameter according to their original publications: for CM, we estimated a different k for each dataset, for wCM and dwCM we set $k = 9, 11$, and for BI^3, we set $k = 5$.

3.3.2 Experiments with Data Imbalance Treatment Techniques

The second group of experiments, with 101 datasets, is carried out to assess the effectiveness of complexity measures when DIT techniques are applied to the training dataset. Table 19 shows a summary of the 102 datasets used in this experiment. Minimum, maximum and mean values for the number of instances, number of features and percentage of the minority class are shown. For more details, please see Table 25 in the Appendices or the GitHub link[4]. 29 out of the 101 datasets have less than 25% of minority class instances. We call them the high imbalanced datasets. The remaining 72 ones are called the low imbalanced datasets. The two sets of datasets used in both experiments, the one described in this section and the one described on Section 3.3.2, share similar characteristics regarding the number of instances, number of features and imbalance. For further details about the similarities between the two sets, please see Figure 34 in the Appendices.

Previous studies have shown that the application of DIT techniques to imbalanced datasets can improve the predictive performance obtained by ML algorithms (CHAWLA *et al.*,

[4] <https://github.com/victorhb/IS2020_results>

Table 19 – Summary of the 101 datasets used in the experiment using data imbalance treatment techniques.

Dataset Characteristic	Min Value	Max Value	Mean Value
Number of Instances	34	2,372	342
Number of Features	3	71	17
% Minority Class	2.33	49.80	32.57

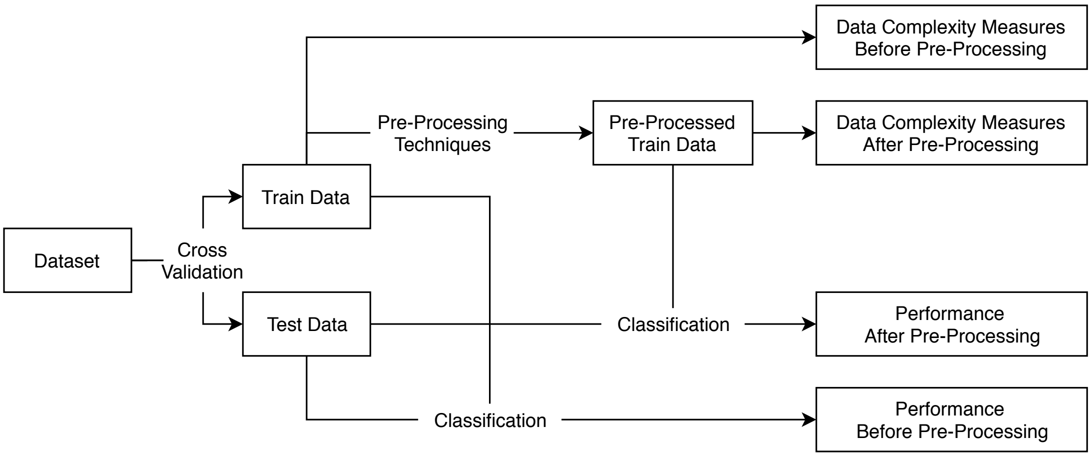
2002; HAN; WANG; MAO, 2005; HE *et al.*, 2008; JO; JAPKOWICZ, 2004; KUBAT; MATWIN *et al.*, 1997). However, there are situations in which their use either reduces or does not affect the predictive performance, and increases the overall computational cost. Additionally, as when using ML algorithms, each DIT technique has a bias, thus some techniques are better than others for particular data conformations (CHAWLA *et al.*, 2002; HAN; WANG; MAO, 2005; HE *et al.*, 2008; JO; JAPKOWICZ, 2004; KUBAT; MATWIN *et al.*, 1997). In these experiments, we investigate how the DIT techniques change the data complexity and whether the changes correlate with the predictive performance of ML algorithms.

Thus, for each dataset, we extracted the data complexity measures and predictive performance before and after applying the DIT techniques. In these experiments, we used the same ML algorithms previously mentioned, with default hyperparameter values. We used a different experimental design from the previous experiment because, in the second experiment, we apply DITs to the datasets. In the literature, when DITs are applied, no hyperparameter tuning is performed in the classification algorithms (SÁEZ *et al.*, 2015; ABDI; HASHEMI, 2016; CHAWLA *et al.*, 2002; HAN; WANG; MAO, 2005; JO; JAPKOWICZ, 2004). This decision is motivated by the fact that tuning would interfere with the DIT analysis, once it would not be possible to track if the observed behavior is due to tuning or the DIT application. The classification algorithms used, and their default hyperparameter values were: SVM with radial kernel, $cost = 1$, and $gamma = \frac{1}{m}$; Random Forest with 500 trees and $\sqrt{m}$ variables; k-NN with $k = 3$; Naive Bayes; C4.5 with 0.2 of threshold for pruning and 2 instances per leaf at minimum; MLP with 0.3 of learning rate and a neurons in the hidden layer; in which m is the number of attributes and $a = \frac{(m+2)}{2}$.

To assess the effect of applying DIT techniques, we measured the gmean of ML algorithms before and after the application. For each dataset, we used 5-fold cross-validation 30 times to compute the mean values for the data complexity measures and the predictive performance. Figure 12 illustrates the followed steps.

The final ratio hyperparameter of the DIT techniques was set to make the two classes completely balanced, except for OSS, which does not have this hyperparameter. For SMOTE, BorderlineSMOTE and ADASYN, the interpolation used the 3 nearest neighbors of each instance. For CBO, we used the k-means clustering algorithm with 4 groups and 100 iterations. We combined two oversampling strategies with CBO: random oversampling (CBO+RO) and SMOTE (CBO+SMOTE).

Figure 12 – Diagram illustrating the steps of the second experiment


Data Complexity Measures Before Pre-Processing
Pre-Processing Techniques
Pre-Processed Train Data
Data Complexity Measures After Pre-Processing
Train Data
Dataset
Cross Validation
Test Data
Classification
Performance After Pre-Processing
Classification
Performance Before Pre-Processing

Source: Barella *et al.* (2020).

3.4 Experimental Results and Discussion

Next, we present and discuss the main results obtained in the evaluation of the original and modified complexity measures for the artificial and real datasets and the effect of DIT techniques on these measures.

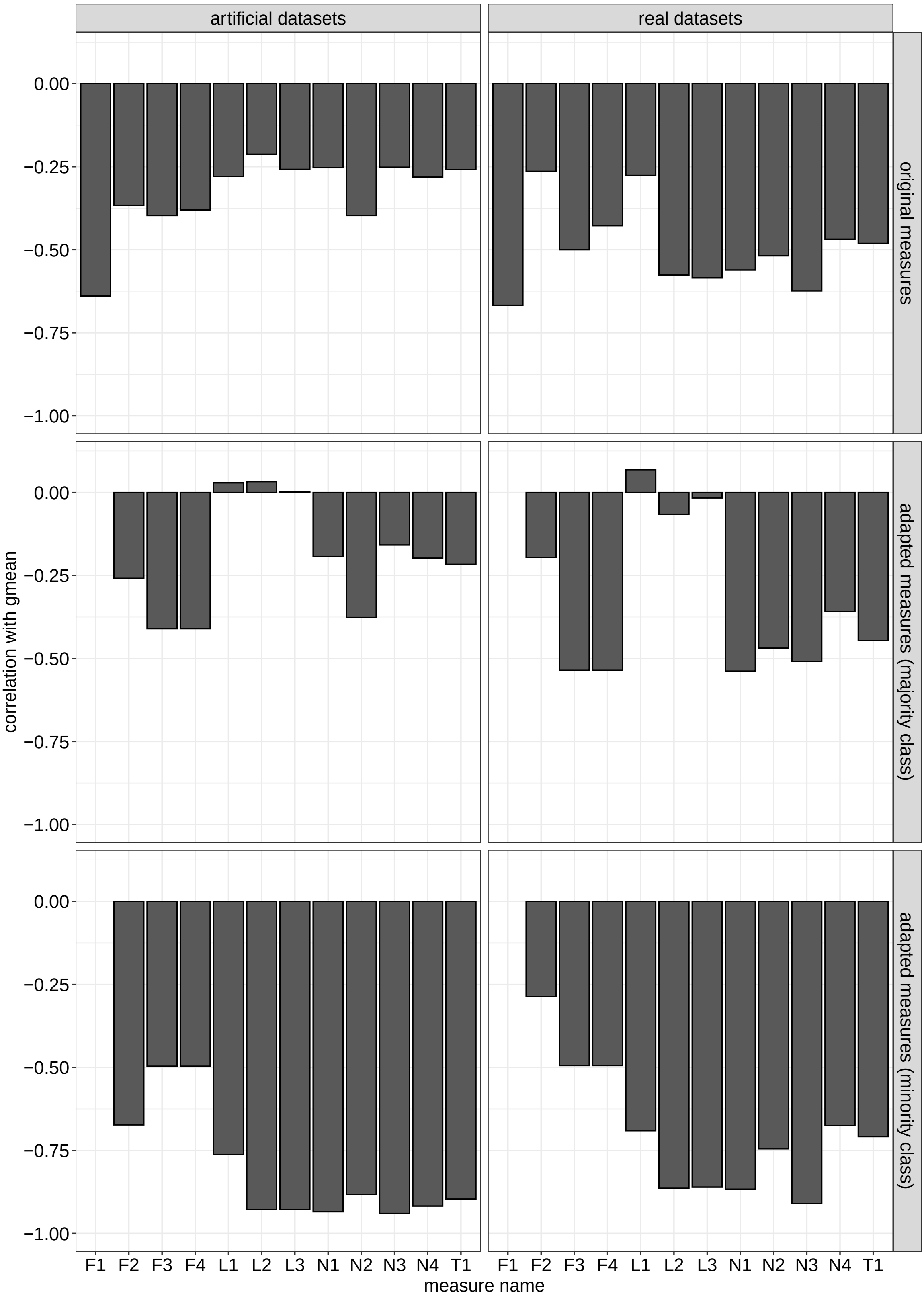
3.4.1 Data Complexity Measures and Real Datasets

In Barella *et al.*, the authors evaluated how data complexity measures performed on artificial imbalanced datasets. There, the authors generated artificial datasets in which the instances were sampled from multivariated normal distributions, and they varied the number of features, class density and imbalance ratio of the datasets. Their experimental results showed that, for the datasets used, they were not suitable for imbalanced data. In the same work, the authors proposed adaptations to these measures, which improved their adequacy to assess the difficulty of the artificial imbalanced datasets considered. In this paper, we expand this analysis by deepening the previous analysis, but this time on real datasets and performing additional evaluations.

Figure 13 compares the Pearson correlations of the gmean performance with the data complexity measures, both the original measures and their adaptations assessing the majority and the minority class. The figure shows the results for the artificial datasets described in Barella *et al.* and the real datasets described in Section 3.3.

Regarding the complexity measures, a value close to 1 should be read as describing a very difficult dataset, while, regarding the gmean performance, a value close to 1 means that a classifier achieved the perfect performance. Difficult datasets are expected to have high values of complexity measures and low values in gmean performance. Thus, in order for the complexity measures to adequately assess the difficulty of a dataset, negative correlations between the

Figure 13 – Correlation between the data complexity (original and adapted) and gmean measures, for the artificial and real datasets.



Source: Barella *et al.* (2020).

measures and the gmean performance are expected. Indeed, all observed correlations were negative, with the exception of the linearity measures for the adapted ones assessing the majority class.

The results show a low correlation between the original data complexity measures and gmean in both artificial and real datasets. The average of the absolute values of the correlations for these measures are 0.42 and 0.49 for artificial and real datasets, respectively. The F1 measure has the strongest correlation among all original complexity measures. However, it has a low value, smaller than 0.75 in the absolute value. It is important to point out that in previous work (BARELLA *et al.*, 2018) F1 had an even lower correlation with the gmean, probably because it is not standardized as described in Section 3.2.1.

Regarding the experimental results using data complexity measures adapted for the minority class, the correlations are improved. On average, the absolute correlation values are increased to 0.4 for the synthetic datasets and to 0.2 for the real datasets. The main difference when compared with the results with the original complexity measures is that the correlation is now higher. As an example, for the real datasets, the correlation of the original N3 is -0.63 and the correlation of N3 adapted for the minority class is -0.91.

Overall, the results for the artificial datasets are similar to those obtained using real datasets. Therefore, the benefits of the data complexity measures also apply to the real datasets.

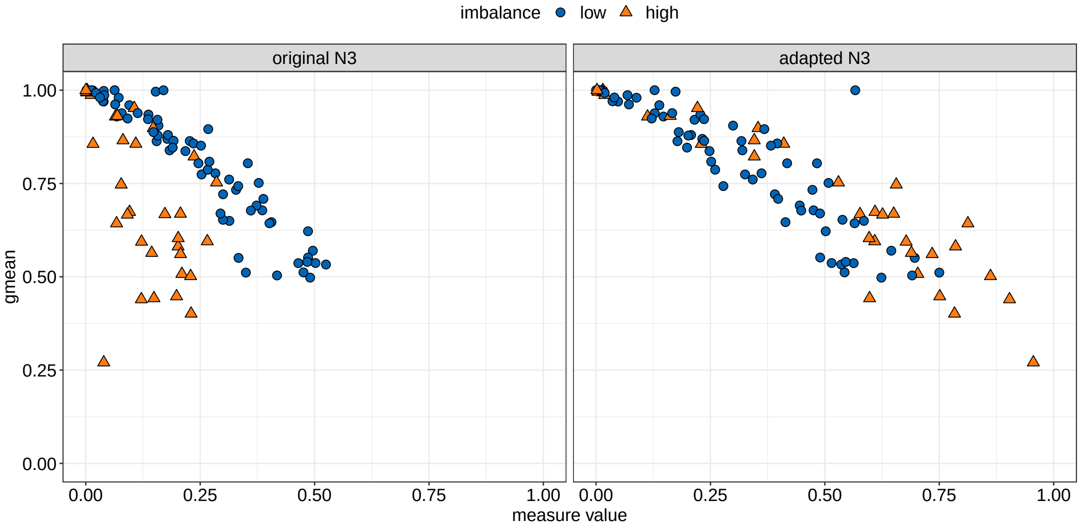
Regarding the results for the adapted measures in the majority class, they showed lower correlations, since the gmean performance is more affected by the performance in the minority class. The correlation between gmean and TPR was 0.94 and between gmean and TNR was 0.62. We expected gmean to be more correlated with TPR than TNR because the minority class is usually more difficult to learn. For this reason, from now on, we only consider the complexity of the minority class for the adapted measures in this paper. Gmean will continue to be calculated the same way, considering both classes.

In the imbalanced data literature, the imbalance ratio is mainly used to show the difficulty a dataset may impose on a classification task. In our experiment, the correlation between the predictive performance and the imbalance ratio was just 0.26 while the correlation between the N3 for the minority class and the predictive performance is 0.91, both in absolute values. These results show that our adaptations can provide relevant information for future studies in imbalanced data classification.

Next, we detail our analysis of the behavior of the most correlated measure, N3. Figure 14 illustrates the behavior of N3. In this figure, each triangle/circle is one dataset, with the shape and color representing different imbalance levels, high and low. The x-axis is the value of N3 measure and the y-axis is the gmean performance. We discretized imbalance into two categories: low imbalance (more than 25% of examples from the minority class in the dataset) and high imbalance (less or equal to 25% of examples from the minority class in the dataset).

It can be observed in Figure 14 that there is a high correlation between the original N3 measure and gmean when the datasets have low imbalance levels. When the datasets have a high imbalance level, the original N3 loses its ability to correlate with gmean, corroborating the fact that they do not correctly capture the difficulty in imbalanced scenarios. The Pearson correlations for the slightly imbalanced and the highly imbalanced scenarios are -0.92 and -0.41, respectively.

Figure 14 – Relation between N3 and gmean.



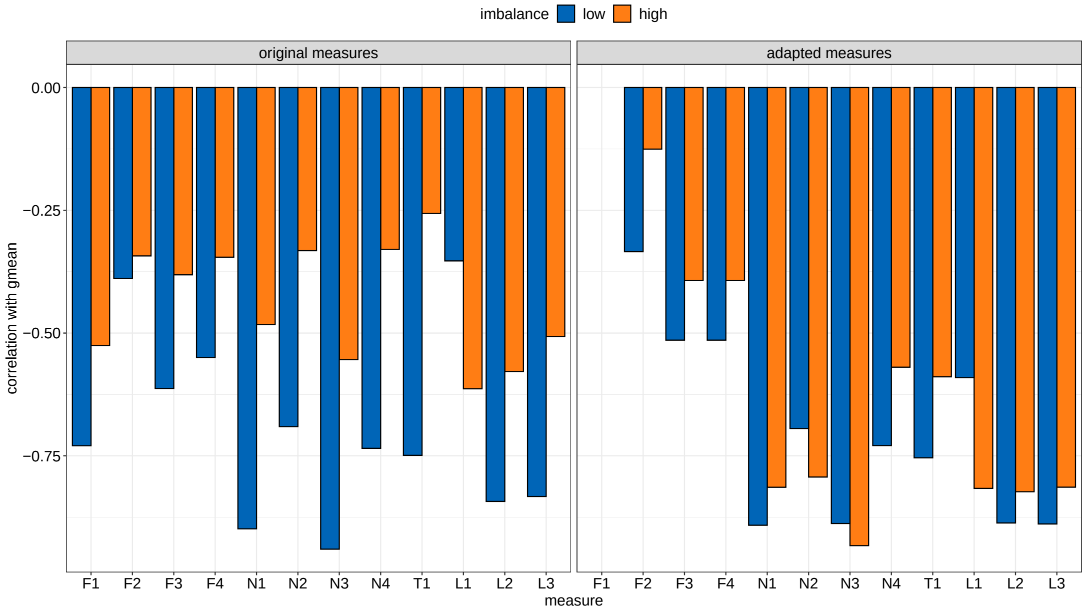
Source: Barella *et al.* (2020).

Regarding the adapted N3 for the minority class, we can see that the correlations are strong for both highly imbalanced and slightly imbalanced datasets. There is only one dataset in the figure with a high divergence between the difficulty assessed by N3 for the minority class and the predictive performance obtained. Apart from this dataset, all others compose a strong linear correlation. The Pearson correlations for the low imbalanced and the high imbalanced scenarios are -0.89 and -0.93, respectively. These results also show that the adapted N3 leads to a smaller difference between these two correlations than the original N3.

To check if the relation found in N3 can be generalized to the other complex measures, we plot their values in Figure 15. In this figure, we also separate the correlations into low and high imbalance. It can be seen that what was seen for N3 also holds for the other neighborhood and the linearity measures, N1, N2, N4, T1, L1, L2, and L3. Thus, again, while for the original measures there is a strong correlation for low imbalance and a weak correlation for high imbalance, for the adapted complexity measures, the correlations are strong for both slightly and highly imbalanced datasets.

These results indicate that the original data complexity measures work correctly only for datasets with low imbalance levels. When the datasets have high imbalance levels, they lose their ability to assess their difficulty. They also show that most of our adapted complexity measures

Figure 15 – Correlation between the data complexity measures and gmean, separated by imbalance degree.



Source: Barella *et al.* (2020).

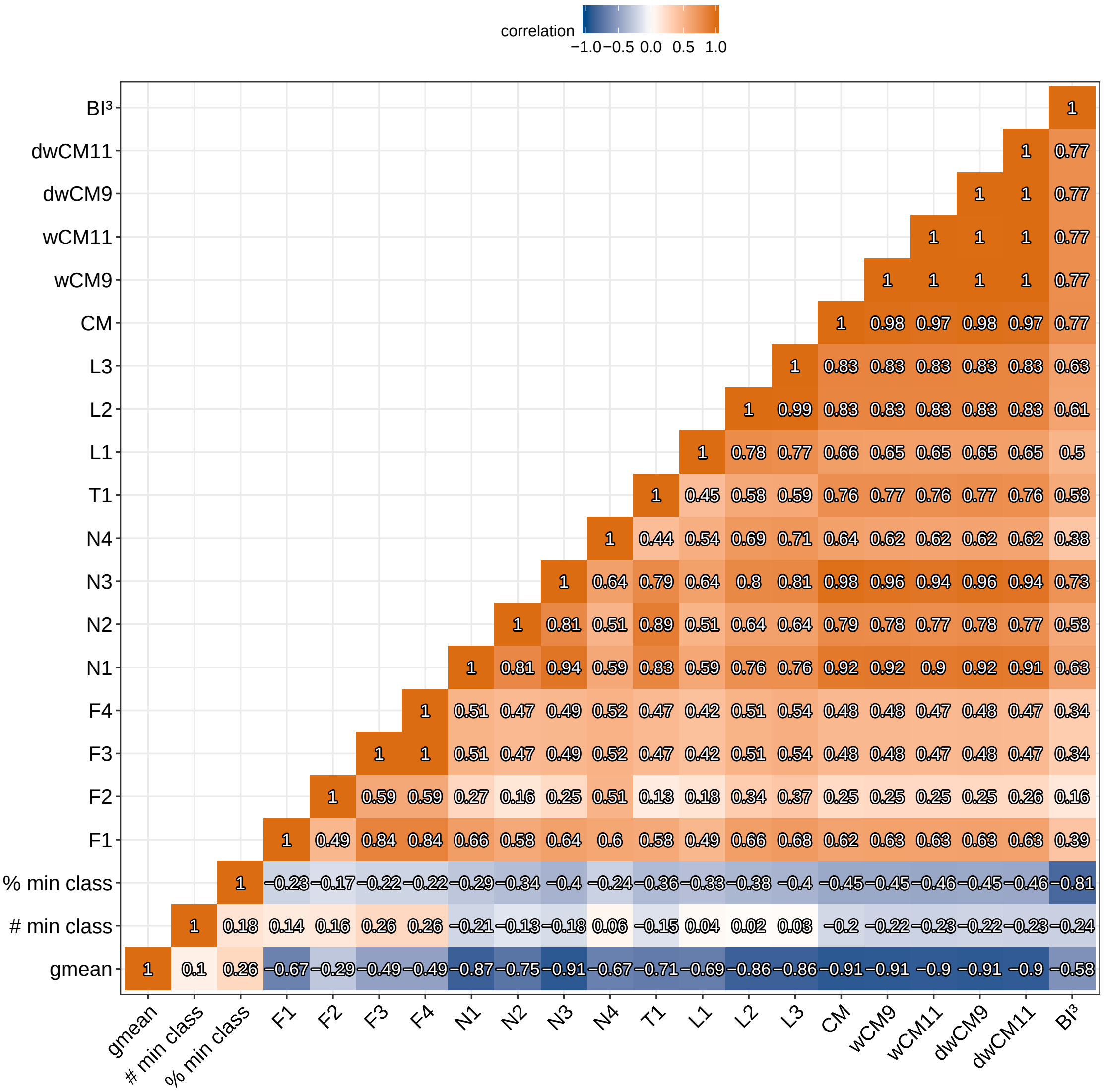
work well, regardless of the dataset imbalance level.

3.4.1.1 Comparison with related work

Figure 16 shows the correlation between the data complexity measures in the minority class and F1 over the 102 datasets. We also included the measures described in the related work considering only the minority class. The most correlated measures with gmean are N1, N3, L2, L3, CM, wCM9, wCM11, dwCM9, dwCM11. The correlation between the percentage of instances belonging to the minority class (% min class) and the number of instances in the minority class (# min class) are the least correlated with the gmean performance. BI^3 is highly correlated with % min class, but presented a low correlation with the gmean. L2 and L3 measures are highly correlated with each other, with a correlation of 0.99, indicating they are capturing similar characteristics from the datasets. Both measures assess the linear separability of the class, but L3 also considers the convexity of the class border. Moreover, N1, N3, CM, wCM9, wCM11, dwCM9, dwCM11 are correlated with each other with correlations above 0.9. All of these measures use the concept of nearest neighbors to be calculated. These correlations are stronger between CM, wCM9, wCM11, dwCM9, dwCM11, varying from 0.97 to 1, indicating they are capturing very similar characteristics. All CMs measures use a kNN classifier to calculate the data complexity.

We selected the measures with a correlation above 0.8 with gmean to understand how the main DIT techniques modify the data complexity. They are N1, N3, L2, L3, CM, wCM9, wCM11, dwCM9, dwCM11. In the next section, we show the results only for N1, N3, L2, and wCM11 to avoid redundancy. The complete table of results for the next section, with all 9

Figure 16 – Correlations between gmean performance, data characteristics, original F1, and complexity measures for the minority class.



Source: Barella *et al.* (2020).

measures, can be accessed from the GitHub link[5].

3.4.2 DIT Techniques Discussion

In this section, we show that the DIT techniques modify the data complexity of the datasets and that there is a correlation between the difference in data complexity and the improvement of predictive performance for most DIT techniques.

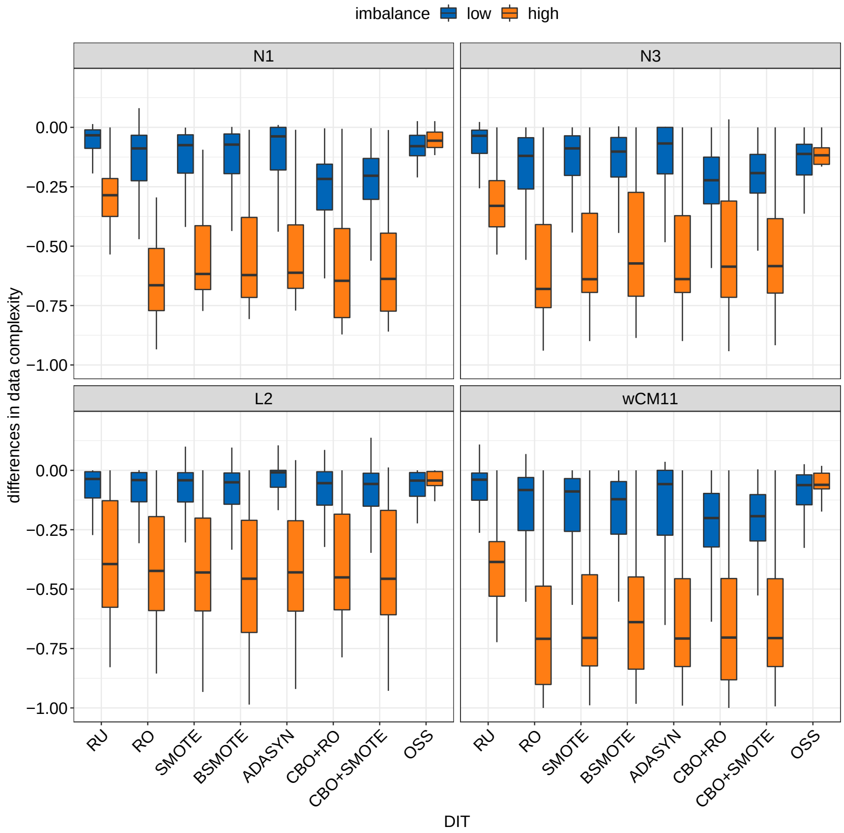
[5] <https://github.com/victorhb/IS2020_results>

3.4.2.1 The reduction of data complexity caused by DIT techniques

To see how DIT techniques affect the complexity of imbalanced datasets, we investigated their effect on the 4 previously selected data complexity measures before and after their application to 101 datasets not used in the experiments reported in the previous section.

Figure 17 shows four boxes, one for each data complexity measure considered. The boxplots represent the difference in data complexity between after and before the application of each DIT technique, represented on the x-axis. The orange boxplots consider the datasets whose minority class have less than 25% of representation (high imbalance). The blue boxplots show these results for the datasets with more than 25% of minority class representation (low imbalance).

Figure 17 – Data complexity measures differences before and after using the DIT techniques.



Source: Barella *et al.* (2020).

On the low imbalanced datasets, the application of DIT techniques had, in general, a small impact on their complexity, for the four data complexity measures. For N1 and N3 measures, the CBO-based techniques presented the highest reduction, with a statistical difference between them and the other techniques, but no statistical difference between the two CBO techniques considering a Friedman-Nemenyi test with a confidence level of 95%. For the N1 measure, also the CBO-based techniques showed the largest decrease, which is statistically different from all other techniques, besides BSMOTE. For the L2 measure, most of the DIT techniques were similar in median, with the exception of ADASYN, that was statistically different compared to CBO+RO, CBO+SMOTE, SMOTE, and BSMOTE.

Overall, the DIT techniques did not obtain a large complexity reduction when applied to the low imbalance datasets. One reason may be that the techniques completely balance the training set and, because those datasets have a low imbalance ratio, they modified them modestly.

The highly imbalanced datasets, on the other hand, were strongly modified by most of the DIT techniques, for the four data complexity measures. RO, e.g., obtained an average difference of 0.45 for N3 and it was statistically different to RU, SMOTE, BSMOTE, ADASYN, and OSS according to a Friedman-Nemenyi test with a confidence level of 95%. Considering N1, RO also obtained the highest difference in median, and was statistically different from all other techniques, except for BSMOTE and the CBO-based techniques. For wCM11, RO was statistically different from RU, OSS, SMOTE, and BSMOTE. Considering L2, BSMOTE was the best in median, but statistically different only to RU, and OSS.

In general, N1, N3 and wCM11 behaved similarly among all the DIT techniques. They had lower differences for RU and OSS, in both high and low imbalance; CBO-based techniques had a larger difference in median for the low imbalanced datasets; and RO had a slightly larger difference for the high imbalanced datasets. All three measures use the concept of NN in their calculation and were highly correlated to each other in the previous experiment, as shown in Figure 16.

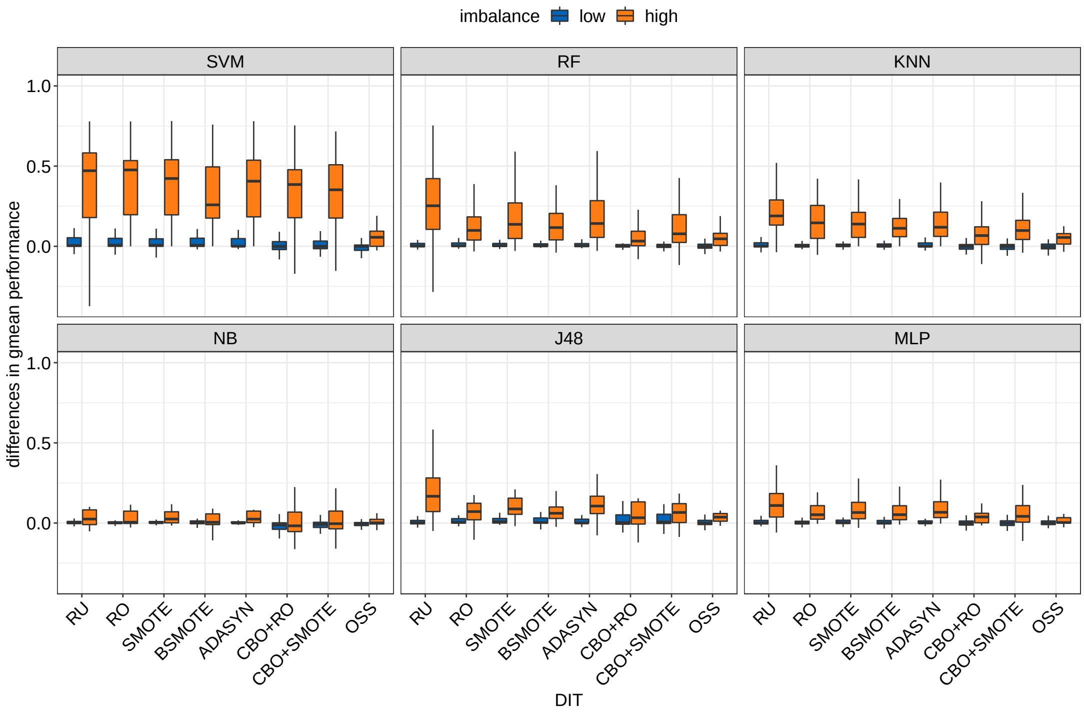
3.4.2.2 The performance gain caused by the DIT techniques

To investigate whether there is a relation between the values returned by the complexity measures and the gains obtained by the DIT techniques, we first investigated the effect of the DIT techniques on the predictive performance of ML algorithms for the 101 datasets used. For such, we assessed the gmean performance of six classification algorithms, using their default hyperparameter values. Figure 18 shows six boxes, one for each ML algorithm, with the differences in the predictive performance of each classifier before and after applying each DIT technique. The blue boxplots show the results for the datasets whose minority class has more than 25% of representation (low imbalance) and the orange boxplots show the results for the datasets with less than 25% of minority class representation (high imbalance).

Figure 18 shows an improvement for all DIT techniques for most classification algorithms, with the exception of OSS. The improvements are larger for the highly imbalanced datasets. In general, ML algorithms have more difficulty in learning good models from these datasets.

The SVM classifier was the algorithm with the largest improvement in the predictive performance after applying the DIT techniques. However, it was the algorithm with the lowest predictive performance on average before applying the DIT techniques. Besides, SVM predictive performance is highly affected by the hyperparameter values (MANTOVANI *et al.*, 2015), which

Figure 18 – Differences in gmean performance before and after applying DIT



Source: Barella *et al.* (2020).

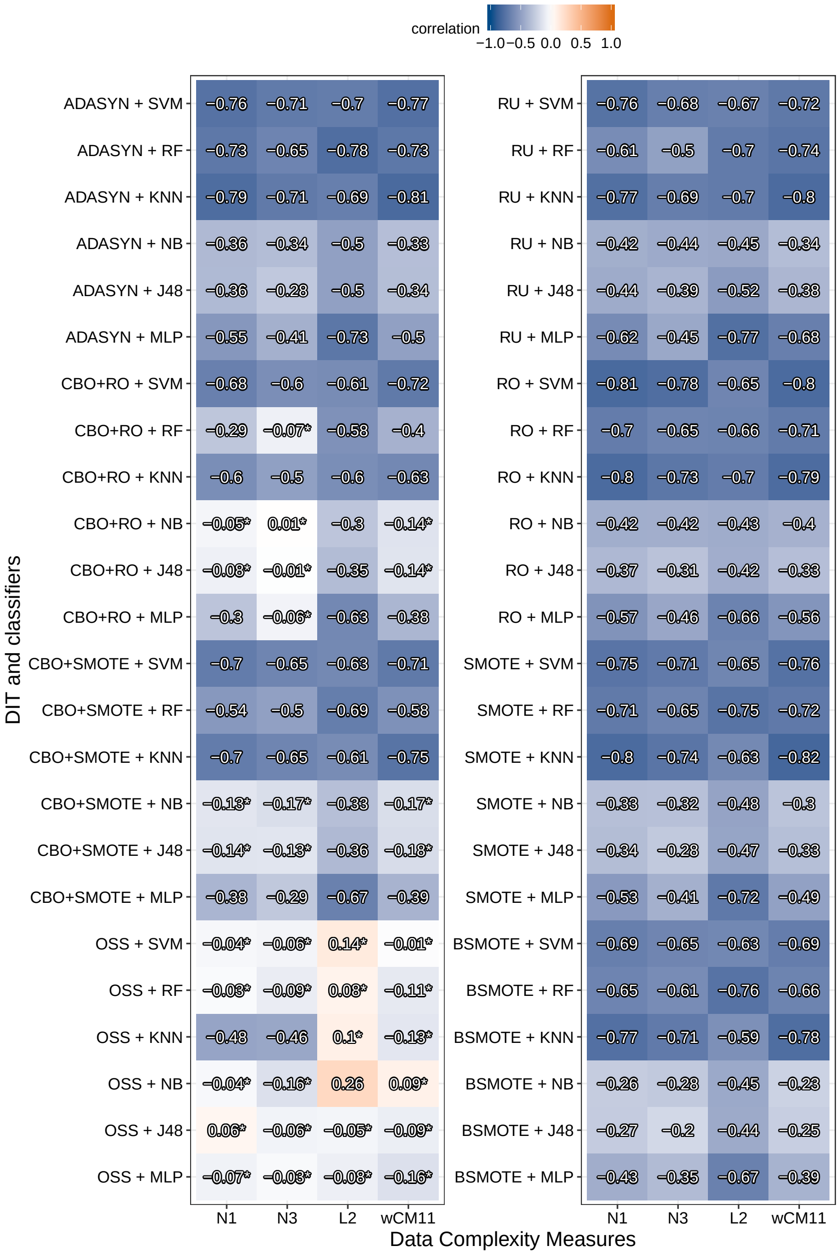
were not tuned in these experiments.

As discussed previously, RO was the DIT technique that reduced the data complexity the most regarding the N3 measure. However, RO did not outperform the other DIT techniques in predictive performance. Actually, in some cases, it performed worse than other techniques, for example RU and SMOTE, when inducing RF and J48. To balance the training set, RO duplicates the number of instances in the minority class. Since it was applied to highly imbalanced datasets, probably RO duplicated most, if not all, minority class examples from the datasets. As N1, N3, and wCM11 are based on the nearest neighbors, the duplication of the instances in the minority class affects their values. However, the improvement on the predictive performance was not better than the other techniques. We believe that RO artificially over reduces the values of those measures causing an underestimation of the data complexity after its application.

3.4.2.3 *The relation between data complexity modification and performance gain*

In order to verify if the reduction in data complexity and the improvement in predictive performance are correlated, we used the Pearson correlation. The results are shown in Figure 19, where the x-axis represents the data complexity measures considered, the y-axis represents the method used (combination of DIT technique and classification algorithm), and each cell of the heatmap is the value of the Pearson correlation between the differences in data complexity and differences in predictive performance when a method is applied to the 101 datasets considered.

Figure 19 – Correlation between the difference in data complexity and difference in predictive performance for all DIT techniques and classification algorithms considered



DIT and classifiers	N1	N3	L2	wCM11
ADASYN + SVM	−0.76	−0.71	−0.7	−0.77
ADASYN + RF	−0.73	−0.65	−0.78	−0.73
ADASYN + KNN	−0.79	−0.71	−0.69	−0.81
ADASYN + NB	−0.36	−0.34	−0.5	−0.33
ADASYN + J48	−0.36	−0.28	−0.5	−0.34
ADASYN + MLP	−0.55	−0.41	−0.73	−0.5
CBO+RO + SVM	−0.68	−0.6	−0.61	−0.72
CBO+RO + RF	−0.29	−0.07*	−0.58	−0.4
CBO+RO + KNN	−0.6	−0.5	−0.6	−0.63
CBO+RO + NB	−0.05*	0.01*	−0.3	−0.14*
CBO+RO + J48	−0.08*	−0.01*	−0.35	−0.14*
CBO+RO + MLP	−0.3	−0.06*	−0.63	−0.38
CBO+SMOTE + SVM	−0.7	−0.65	−0.63	−0.71
CBO+SMOTE + RF	−0.54	−0.5	−0.69	−0.58
CBO+SMOTE + KNN	−0.7	−0.65	−0.61	−0.75
CBO+SMOTE + NB	−0.13*	−0.17*	−0.33	−0.17*
CBO+SMOTE + J48	−0.14*	−0.13*	−0.36	−0.18*
CBO+SMOTE + MLP	−0.38	−0.29	−0.67	−0.39
OSS + SVM	−0.04*	−0.06*	0.14*	−0.01*
OSS + RF	−0.03*	−0.09*	0.08*	−0.11*
OSS + KNN	−0.48	−0.46	0.1*	−0.13*
OSS + NB	−0.04*	−0.16*	0.26	0.09*
OSS + J48	0.06*	−0.06*	−0.05*	−0.09*
OSS + MLP	−0.07*	−0.03*	−0.08*	−0.16*

DIT and classifiers	N1	N3	L2	wCM11
RU + SVM	−0.76	−0.68	−0.67	−0.72
RU + RF	−0.61	−0.5	−0.7	−0.74
RU + KNN	−0.77	−0.69	−0.7	−0.8
RU + NB	−0.42	−0.44	−0.45	−0.34
RU + J48	−0.44	−0.39	−0.52	−0.38
RU + MLP	−0.62	−0.45	−0.77	−0.68
RO + SVM	−0.81	−0.78	−0.65	−0.8
RO + RF	−0.7	−0.65	−0.66	−0.71
RO + KNN	−0.8	−0.73	−0.7	−0.79
RO + NB	−0.42	−0.42	−0.43	−0.4
RO + J48	−0.37	−0.31	−0.42	−0.33
RO + MLP	−0.57	−0.46	−0.66	−0.56
SMOTE + SVM	−0.75	−0.71	−0.65	−0.76
SMOTE + RF	−0.71	−0.65	−0.75	−0.72
SMOTE + KNN	−0.8	−0.74	−0.63	−0.82
SMOTE + NB	−0.33	−0.32	−0.48	−0.3
SMOTE + J48	−0.34	−0.28	−0.47	−0.33
SMOTE + MLP	−0.53	−0.41	−0.72	−0.49
BSMOTE + SVM	−0.69	−0.65	−0.63	−0.69
BSMOTE + RF	−0.65	−0.61	−0.76	−0.66
BSMOTE + KNN	−0.77	−0.71	−0.59	−0.78
BSMOTE + NB	−0.26	−0.28	−0.45	−0.23
BSMOTE + J48	−0.27	−0.2	−0.44	−0.25
BSMOTE + MLP	−0.43	−0.35	−0.67	−0.39

Source: Barella *et al.* (2020).

Values followed by "*" mean that the p-value for that correlation was above 0.05.

The DIT technique that, in general, presented the lowest correlations in magnitude was OSS. Moreover, NB and J48 performances are usually not well correlated with the reduction in data complexity of any of the measures considered compared to the other classification algorithms when the same DIT technique is applied. N1, N3 and wCM11 do not correlate well when CBO-based techniques were applied.

Despite some low correlations in magnitude pointed out previously, most of the differences in data complexity are highly correlated with the predictive performance of the methods considered. They corroborate with the evidence discussed in Section 3.4.1 that the adapted data complexity measures considered are suitable tools to assess the data complexity of imbalanced datasets, now considering when DIT techniques are applied.

3.5 Final Remarks

As confirmed in this paper, dataset imbalance is not a problem in itself. But it increases the chances of the adverse effects of other characteristics, such as overlapping and difficult border decisions. To investigate these effects, we used data complexity measures. The original data complexity measures have been used in the literature to assess these characteristics, including their occurrence in imbalanced tasks. We show in this paper that the original data complexity measures do not work well with imbalance in real datasets. Therefore we strongly discourage their use in these scenarios. However, we also show that simple adaptations of these measures can make them useful to assess the difficulty of ML classification algorithms to deal with imbalanced datasets.

According to our experimental results, most of the adapted data complexity measures correlated better with the difficulty in imbalanced tasks than the imbalance ratio itself. Thus, the adapted measures can assess the difficulty of inducing a good model from a dataset better than the imbalance ratios used in the literature. Thus, the adapted measures can provide meaningful insights for data science researchers and practitioners. They can improve the understanding of the difficulty of the datasets used and guide the application of ML algorithms to these datasets. Another contribution from this study is to show the importance of selecting DIT techniques that can effectively reduce the data complexity, instead of only balancing the training set.

The experimental results show that the reduction of data complexity obtained by using DIT techniques occur mainly for highly imbalanced datasets. They also show that, for most of the DIT investigated, there is a correlation between the reduction in data complexity and gain in predictive performance.

Our adaptations of complexity measures were designed and tested only on binary datasets. For use in multiclass datasets, some of the data complexity measures may be dependent on the class decomposition strategy used (one versus all; one versus one). We believe that this is a good direction for future work. Moreover, we considered only the gmean performance for the experiments, because it is the most popular in recent works. Although it is a widely used performance measure for imbalanced data classification tasks, it would be interesting to study the behavior of the data complexity measures with other metrics used in imbalanced dataset classification tasks, such as AUC, F-measure, and kappa.

Future work shall consider the use of the adapted complexity measures in the proposal of meta-learning systems for the recommendation of suitable DIT techniques for a new dataset. The measures values can also be explored in the proposal of new data balancing strategies. For instance, one may guide the generation of new instances in the minority class in order to optimize a given measure value.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES). The authors would like to thank CNPq, FAPESP (grants 2012/22608-8 and 2015/01382-0), CAPES (grant 88887.162551/2018-00), and CAPES/COFE-CUB (grant 99999.000555/2016-09) for their financial support. The authors would like to thank CeMEAI-FAPESP (grant 2013/07375-0) for the computational resources and Intel for the hardware and software server used in part of the experiments.

CHAPTER

4

THE INFLUENCE OF SAMPLING ON IMBALANCED DATA CLASSIFICATION

Authors

Victor H. Barella *University of São Paulo, São Carlos, São Paulo, Brazil*

Luís P. F. Garcia *University of Brasília, Brasília, Distrito Federal, Brazil*

André de Carvalho *University of São Paulo, São Carlos, São Paulo, Brazil*

Abstract

Classification tasks using imbalanced data are not challenging on their own. When the classes are linearly separable, a regular classification algorithm usually induces predictive models able to distinguish the classes properly. Imbalanced data poses difficulty for the minority class when the training sets have classes overlapping or a complex border decision. Assessing these characteristics is fundamental to understand the classification task difficulty and to choose adequate pre-processing techniques for imbalanced data. Measures able to identify the complexity of a classification task for a given dataset have been proposed. These measures use different criteria to identify how difficult it is to induce a classifier from a dataset. In this paper, we investigate the use of data complexity measures to estimate the best sample size for data imbalance pre-processing techniques. For such, this paper assesses the predictive performance and the data complexity of real datasets after applying pre-processing techniques using different sample sizes. According to experiments, the data complexity measures are a tool to help in choosing a proper sample size to improve the predictive performance of the classifiers. We also observe that only the difficulty of predicting the minority class is not enough when dealing with sampling. As an alternative to deal with this deficiency, we suggest a combination of the data complexity of both classes.

4.1 Introduction

Real world labeled dataset is often imbalanced. A dataset is imbalanced when there is a disproportion of the number of examples among the classes. Data imbalance can occur in binary and multiclass datasets. Although in this paper, we investigate imbalanced binary data, the same study can be easily adapted for multiclass imbalanced data. Several authors have found that imbalance usually relates to low performance on the minority classes, i.e., the classes less represented (KUBAT; MATWIN *et al.*, 1997; FERNáNDEZ *et al.*, 2018). Even so, imbalance does not impose a problem when the classes are well separated in the attributes space. The problem arises when characteristics such as overlapping and difficult border decisions combine with imbalance - in those cases, the majority classes tend to predominate over the minority ones on classification models (BATISTA; PRATI; MONARD, 2004).

The main pre-processing techniques for imbalanced tasks aim at balancing the number of examples from the classes in order to decrease the imbalance, the overlapping, and the difficult border decisions (HE; GARCIA, 2008). For example, some of them interpolate new instances while others remove majority class instances (CHAWLA *et al.*, 2002; KUBAT; MATWIN *et al.*, 1997). They claim that those techniques improve the performance obtained in several datasets, as shown in the literature (FERNáNDEZ *et al.*, 2018). However, there is a lack of understanding of what those techniques do to the datasets, i.e., how the interpolation or the RU and RO modify their characteristics.

Data complexity measures (CMs) allow one to estimate the expected difficulty of a classification task by extracting descriptions of the overlap between classes imposed by feature values, the separability, distribution of the data points, and certain structural characteristics of the task based only on the training set available for learning (HO; BASU, 2002). Barella *et al.* suggested adaptations in the CMs should be made to extract proper information of the datasets when they are imbalanced. Such adaptations consist of assessing the difficulty of each class separately, and they empirically demonstrate that the difficulty of the minority class is a good estimator of the difficulty of an imbalanced classification task. We show in this paper that a combination of the difficulty of all classes is preferred when using balancing techniques.

In this paper, we explore the main pre-processing techniques for imbalanced datasets regarding performance and CM. We vary the sample size generated by the techniques and observe the modification on the CMs and the performance considering several classification algorithms to identify the combinations that can generate better models. The obtained results indicate that the CMs are useful to estimate the sample size parameter on pre-processing techniques. Besides that, the results reinforce the idea using an empirical analysis that imbalance itself is not a problem except when combined with overlapping and difficult border decisions.

This paper is separated into five sections. Section 4.2 describes the CMs used to estimate the difficulty of each class separately. Moreover, we present the main pre-processing techniques for imbalanced learning considered in this paper. Next, Section 4.3 presents the experimental setups designed in this work. The experimental results are shown and discussed in Section 4.4. Section 4.5 concludes this paper with contributions, limitations and future works.

4.2 Background

Imbalance ratio measures are usually used to describe the difficulty of imbalanced datasets. They capture information about the disproportion between the classes, but they lack information about overlapping and border decision. To that purpose, the CMs are commonly applied. They were gathered and proposed by Ho and Basu and since then they are used on several domains (LUENGO; HERRERA, 2015). After, Barella *et al.* showed that the original CMs do not work correctly when the datasets are imbalanced and suggested adaptations to overcome it. We considered those adaptations on this paper.

4.2.1 Data Complexity Measures for Imbalanced Classification Tasks

Due to lack of space, we considered only a subset of the CMs. We chose those which are more correlated with the imbalance problem as described in (BARELLA *et al.*, 2018): N3, N1, and L2. They are described below.

4.2.1.1 N3: Leave-one-out error rate of the 1NN classifier

N3 gives the leave-one-out error of a nearest-neighbor (NN) classifier, which is easy to be calculated and is a good indicator of the separability of the classes. We considered the N3 adaptation per class which takes the NN training error for each class. Equation 4.1 represents our adaptation considering a *class* 1, where $NN(x_i)$ is the nearest neighbor of x_i, y_i is the class of example x_i, I is the indicator function that returns 1 if its argument is true and 0 if it is false, and n_{c_1} is the number of examples from class 1.

$$N3_{c_1} = \frac{\sum_{i=1}^{n_{c_1}} I(NN(x_i) \neq y_i)}{n_{c_1}} \tag{4.1}$$

4.2.1.2 N1: The fraction of points on the class boundary

N1 builds a minimum spanning tree (MST) that connects all the examples from a dataset based on their distances, despite their classes. Next, it counts the number of examples connected to at least one example from another class. Those examples are possibly borderline and the fraction of their number over the total number of examples is the final N1 measure. N1 is bounded between 0 and 1, and values closer to 0 represent a lower complexity. For this paper, we considered the adaptation which consists in calculating the N1 for each class. Equation 4.2

represents N1 considering a *class* 1, where (x_i, x_j) represents a connection between examples x_i and x_j, and *MST* represents the set of all connections in the tree.

$$N1_{c_1} = \frac{1}{n_{c_1}} \sum_{i=1}^{n_{c_1}} I((x_i, x_j) \in MST \wedge c_1 \neq y_j) \tag{4.2}$$

4.2.1.3 L2: The training error of a linear classifier

L2 is the training error of a linear classifier. For that purpose, it builds a linear model and the error rate of the classifier is computed. Higher values represent non-linear problems. We considered the adaptation that takes the error rate per class, as Equation 4.3 shows, where $h(x_i)$ is the class prediction of the linear classifier for example x_i.

$$L2_{c_1} = \frac{\sum_{i=1}^{n_{c_1}} I(h(x_i) \neq y_i)}{n_{c_1}} \tag{4.3}$$

Some works used them to tackle the imbalance problem. Luengo *et al.* used the CMs to predict whether a pre-processing technique can be successful or not. They found intervals of values of some of the CMs in which the techniques showed performance improvement. Other authors also used the CMs to analyze the suitability of using a specific technique in imbalanced datasets. Díez-Pastor *et al.* used them to predict CMs intervals for which some diversity-enhancing techniques may improve the results of an ensemble method. Fernández, Jesus and Herrera used one CM combined with other characteristics (such as imbalance) in a multi-objective approach to select attributes and instances from a dataset.

4.2.2 Pre-Processing Techniques

The techniques for imbalanced learning are usually divided into two main general approaches: (1) pre-processing the data in order to make it more balanced (CHAWLA *et al.*, 2002; HAN; WANG; MAO, 2005; HE *et al.*, 2008; JO; JAPKOWICZ, 2004; KUBAT; MATWIN *et al.*, 1997); and (2) development of algorithms in the classification step that are more tolerant and robust to handle imbalanced data (GONZALEZ-ABRIL *et al.*, 2014; CIESLAK *et al.*, 2012; CANO; ZAFRA; VENTURA, 2013; DIAMANTINI; POTENA, 2009). While dealing with imbalanced data in the pre-processing techniques allows the selection of the most appropriate technique and can include the expert feedback, the same task in the classification step with robust classifiers can avoid one more bias in the ML pipeline. In this paper, we focus on the former.

Regarding the pre-processing techniques, the methods can be categorized into two groups: undersampling and oversampling methods (FERNáNDEZ *et al.*, 2018). Undersampling methods make the data more balanced by removing instances of the majority class while oversampling methods do that by inserting instances in the minority class (FERNáNDEZ *et al.*, 2018). Both undersampling and oversampling can be done randomly or according to an informed strategy. Next, we discuss the main pre-processing strategies.

4.2.2.1 Random sampling

In the random undersampling (RU), instances of the majority class are removed at random until a more balanced class distribution is reached (HE; GARCIA, 2008). In the random oversampling (RO), instances of the minority class are replicated at random until a more balanced class distribution is reached (HE; GARCIA, 2008).

4.2.2.2 Synthetic minority oversampling techniques

The *Synthetic Minority Oversampling Technique* (SMOTE) (CHAWLA *et al.*, 2002) is an oversampling technique that creates artificial data by interpolation, as follows. At each iteration, SMOTE selects an instance x at random in the minority class, and then it looks for the k NNs of x. SMOTE then selects one of the neighbors z at random and creates a new instance, which is a combination of x and z. The combination is an interpolation that randomly creates any possible point between x and z. This step is repeated until a more balanced distribution of instances is reached.

The *Borderline SMOTE* (BSMOTE) is based on SMOTE and it searches for minority examples from decision boundaries to interpolate (HAN; WANG; MAO, 2005). Instead of selecting minority examples from all training set, it selects minority examples from the decision boundary. The method that BSMOTE uses to select them is: (1) find the k NN for a minority example x; (2) count the number N_{maj} of neighbors that belongs to the majority class; (3) if $\frac{k}{2} \leq N_{maj} < k$ then x is put in a set called DANGER; (4) repeat the steps for all minority examples. After SMOTE is run to balance the dataset but it selects only examples from the DANGER subset.

The *Adaptive Synthetic Sampling Approach* (ADASYN) is also based on SMOTE and it addresses the number of examples to be interpolated by each minority example (HE *et al.*, 2008). ADASYN follows the next steps: (1) first, it defines G, that indicates how many examples should be interpolated for the entire minority class; (2) next, for each example in the minority class, it calculates the percentage of majority examples in the k nearest neighbors; (3) the set of all percentages (Γ_i, where i is the minority example) is normalized so that $\sum \Gamma_i = 1$; finally, $\Gamma_i \times G$ gives the number of examples to be interpolated using SMOTE for each minority example i.

More recent pre-processing techniques for imbalanced classification have been proposed, including other SMOTE adaptations (BARUA *et al.*, 2014; ABDI; HASHEMI, 2016), undersampling based on clustering (NG *et al.*, 2015), and sampling based on evolutionary algorithms (YU; NI; ZHAO, 2013). For example, Barua *et al.*, claimed that the SMOTE adaptations were favoring noisy examples e proposed a new way of selecting minority examples that discard the ones with no minority neighbor. Their proposed adaptation, which is called *Weighted Minority*

Oversampling Technique (MWMOTE), weights the minority examples and generates new examples within a minority cluster. The *Mahalanobis Distance-based Over-sampling technique* (MDO) (ABDI; HASHEMI, 2016) is another SMOTE adaptation. It generates synthetic minority examples that have the same Mahalanobis distance from the class mean as other existing minority examples. In this article, we are considering the standard pre-processing techniques because they are most used and studied and the adaptation of the CMs proposed by Barella *et al.*.

4.3 Experimental Setup

The empirical analysis aims at evaluating how the main classification algorithms and CMs behave when varying the percentage of sampling using pre-processing techniques. Therefore, the objective is to determine whether the pre-processing techniques and the sample size build accurate models with low class overlapping or simple border decisions for a diverse set of classification techniques and classification problems.

For the empirical analysis, we collected 43 binary datasets from the OpenML repository (VANSCHOREN *et al.*, 2013) with less than 25% of minority class representation. For each dataset, we compute the CMs values described in Section 4.2.1 plus the predictive performance achieved by the ML techniques when applied to five pre-processing techniques described in Section 4.2.2 for each sampling rate.

The pre-processing techniques used are RU, RO, SMOTE, BSMOTE, and ADASYN because they are the most standard techniques in the literature. The sample size is added according to rates that range from 10% to 200%, with intervals of 10%. Each rate value corresponds to a percentage of the examples needed to completely balance the training set, i.e., a rate of 100% means a training set with a proportion of $1:1$ between the classes. To illustrate, consider a training set with 110 examples on the majority class and 10 examples on the minority class. A percentage of 10% means $(110-10)\times 0.1 = 10$ examples to be added to the minority class (in the case of oversampling techniques) or removed from the majority class (in the case of undersampling technique).

The classifiers used are: the ANN based on backpropagation (also called Multilayer Perceptron - MLP) with one hidden layer, learning rate of 0.3 and momentum of 0.5 (HAYKIN, 1999); the SVM with linear and radial kernel (CRISTIANINI; SHAWE-TAYLOR, 2000); the DT induced by the C4.5 algorithm with pruning (QUINLAN, 1986); the ensemble called Random Forest (RF) with 500 DTs, the kNN, a lazy learning technique with $k = 3$ and Naive Bayes (NB) classifier (MITCHELL, 1997).

In order to decrease the randomness, 10 different executions were made for each sampling rate. The predictive performance of the classifiers and CMs were evaluated with gmean. Once the datasets are imbalanced, the gmean performance measure is used and the performance was evaluated using the 5-fold stratified cross-validation. The gmean is defined by $\sqrt{TP_R \times TN_R}$, where TP_R is the true positive rate and TN_R is the true negative rate.

4.4 Results and Discussion

This paper aims at showing that the right choice in the sample size of the pre-processing techniques can improve the performance of the classifiers and the CMs. Also, the CMs have similar behavior to the performance measure. In other words, the CMs are potentially good estimators for pre-processing techniques sample size.

First, we show a summary of how the techniques behave for each classification algorithm and each sample size considering the gmean performance. After, it is shown a general overview of the performance results obtained through a ranking regarding combinations of pre-processing techniques and classification algorithms. Finally, we show the behavior of three data CMs and how they relate to the performance observed.

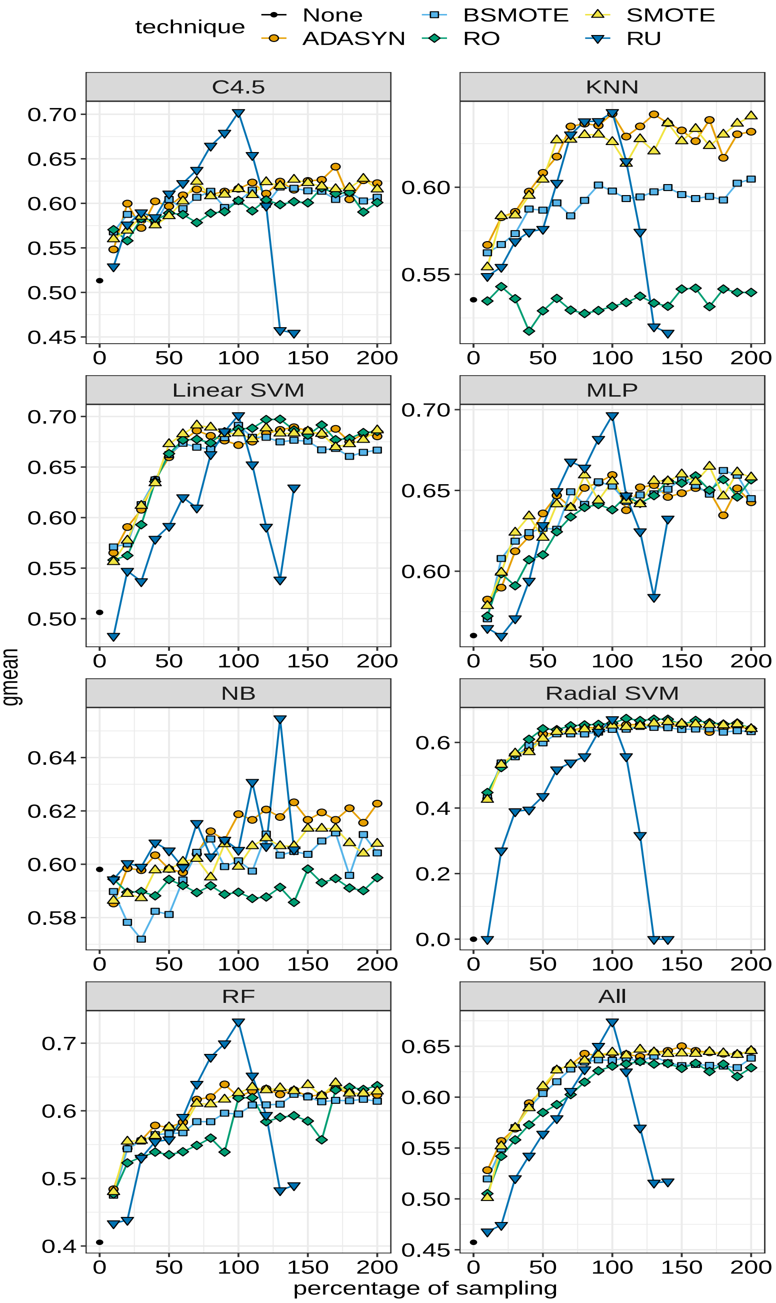
Figure 20 represents the median behavior of the pre-processing techniques when varying their sample size parameter regarding the performance of each classifier. The last plot summarizes the average gmean values for all of the classifiers. Each line represents a pre-processing technique where each point is the median value of the gmean performance considering the combination of that classification algorithm and that pre-processing technique with the sample size defined on axis x. The None performance is shown in black circles at 0% of sample size, the BSMOTE by blue squares, SMOTE by yellow triangles, ADASYN by yellow circles, RO by green lozenges and RU by dark blue upside down triangles.

For most of the classification algorithms, the oversampling techniques improved until it stabilizes and the RU improved until near 100% (completely balanced) and then it decreases. That happened for C4.5, Linear SVM, MLP, Radial SVM, RF, and the summary of all algorithms. The exception of the kNN is related to RO, which has not improved nor decreased the performance independently of sample size. The replication approach of RO was not effective on the observed results for kNN, possibly because it does not aggregate new information for the training set. Also, NB does not behave similarly, possibly due to its restricted assumptions.

In general, the pre-processing techniques behaved as following: for the undersampling technique, the performance improved accordingly to the sample size until it got completely balanced, after that point the performance decreased; for the overlapping techniques, the performance improved until a point where the curve became mostly flat. In that point, RU achieved the best performance around 100% of balance but the other techniques maintained a high performance under sampling sizes higher than 100%.

Table 20 shows the percentage that each technique ranked first place when comparing the gmean performance per classification algorithm. We included another row called "All" which represents the ranking for the pre-processing techniques independently of the classification

Figure 20 – Median behaviour of pre-processing techniques regarding gmean performance from several classification algorithms.



Source: Barella, Garcia and Carvalho (2019).

algorithms and another column called "None" which represents the ranking for the classifiers without any pre-processing technique.

RU ranked first place the most among the majority of the classification algorithms, except for linear and radial SVMs, for those algorithms the ADASYN and RO performed first place respectively in the ranking. Considering all classification algorithms, not applying any pre-processing technique performed better in 7% of the datasets, and although it is higher than the percentage of victories of SMOTE and BSMOTE (both had 2% of the victories) that does

Table 20 – Percentage of victories for each pre-processing technique per classification algorithm

	Techniques					
Class. Alg.	**None**	**ADASYN**	**BSMOTE**	**RO**	**SMOTE**	**RU**
C4.5	12%	9%	9%	19%	2%	48%
KNN	14%	12%	9%	0%	5%	59%
MLP	9%	7%	5%	9%	19%	50%
NB	14%	14%	19%	2%	21%	28%
RF	12%	2%	0%	24%	7%	55%
Linear SVM	12%	26%	12%	14%	19%	17%
Radial SVM	9%	9%	5%	40%	5%	31%
All	7%	19%	2%	14%	2%	55%

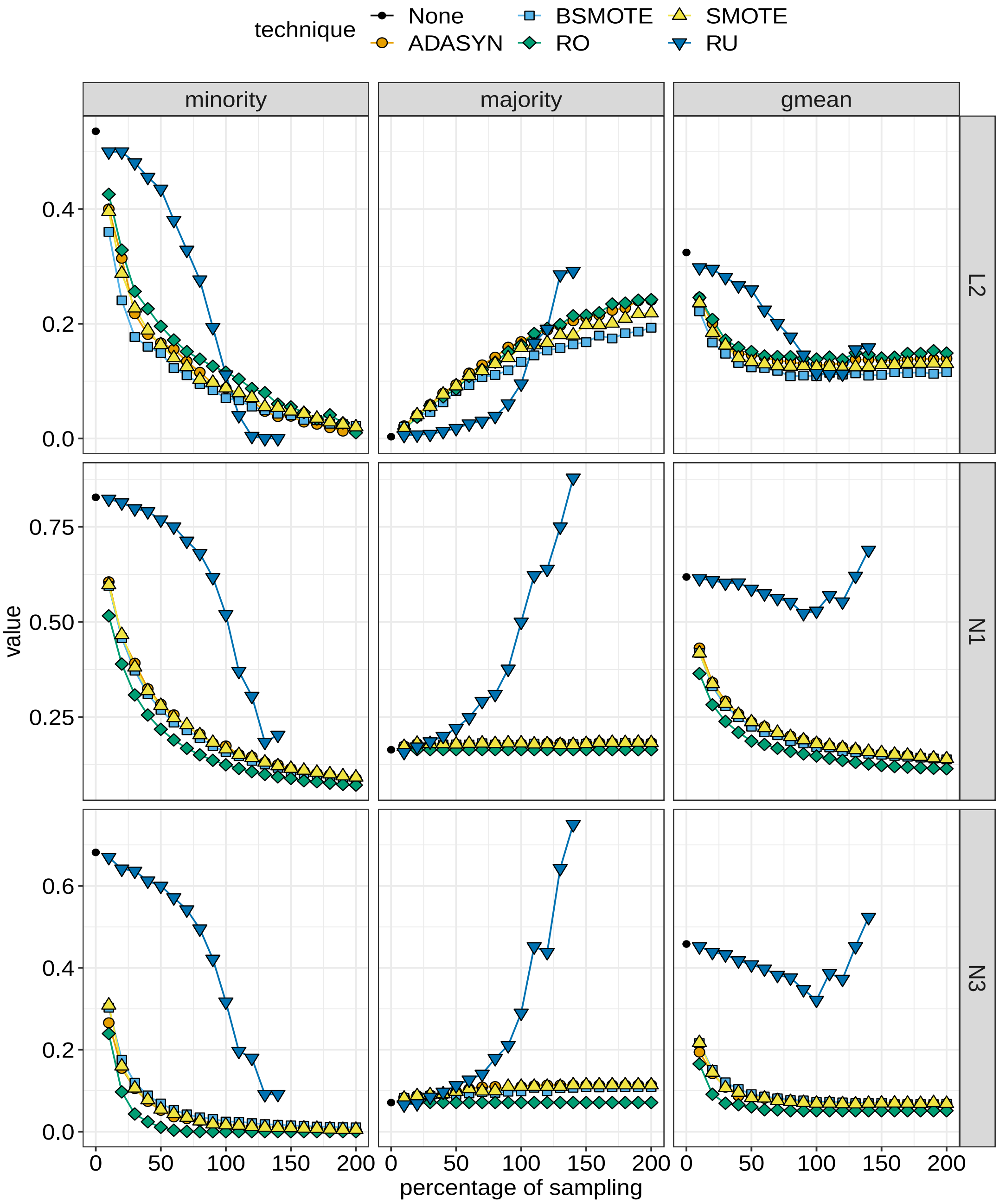
not mean that they performed worse than None. Since SMOTE, BSMOTE and ADASYN are all based on interpolation, they perform similarly.

Next, we observe the CMs to check if they behave similarly. Figure 21 shows the median behavior of the pre-processing techniques when varying their sample size parameter regarding the CMs. We first analyze the behavior of the CMs for the minority class since they are good estimators of the difficulty imposed by the data (BARELLA *et al.*, 2018). All measures for minority class, i.e., L2 minority, N1 minority, and N3 minority decrease the data difficulty along with the sample size - the higher the sample size, the lower the difficulty of the minority class. Different from Barella *et al.*, we observe that when a balancing technique is applied, the difficulty of the minority class no longer stands for the whole set anymore. A contrast is observed with the majority class: it gets more difficult with the sampling size increase.

The observed behavior shows that a combination of both minority and majority difficulties must be calculated in order to represent the difficulty of the whole training set when a sampling technique is applied. We opted for a geometric mean of both values since the gmean performance is used. The equation $1-\sqrt{(1-C_{maj})\times(1-C_{min})}$ is used to calculate the CMs geometric mean, where C_{maj} and C_{min} are the value of majority CM and minority CM respectively. The subtraction by one is performed in order to maintain the meaning of the CMs which 1 represents maximum difficulty, and 0 represents minimum difficulty.

The pattern noted on the performance analysis appeared when using the gmean of the CMs. In other words, the pre-processing techniques behaved on classification algorithms and CMs as follows. For the RU, the performance improved (and the difficulty decreased) accordingly to the sample size until it got completely balanced, after that point, the performance decreased, and the difficulty increased. For the overlapping techniques, the performance improved, and the CMs decreased until a point where the curve became mostly flat.

Figure 21 – Median behaviour of pre-processing techniques regarding data complexity measures.



Source: Barella, Garcia and Carvalho (2019).

4.5 Conclusion

Imbalanced datasets are challenging only when the minority class shows class overlapping or complex border decisions. Measuring such characteristics is fundamental to understand the classification task and to decide about adequate pre-processing techniques and their parameters to use. The CMs are a tool to describe datasets characteristics, and they can be used to understand and explain imbalanced domains. In this paper, we expand their use on analyzing balancing pre-processing techniques. We showed evidence that they are potentially useful for estimating sample size parameters of pre-processing techniques. As future work, we will investigate how to use their concepts on pre-processing techniques or adaptations of classification algorithms for imbalanced datasets.

Although the difficulty of the minority class is a good estimator of the whole imbalanced dataset, we show that only the minority class information is not enough when dealing with balancing techniques. We suggest a combination of both minority and majority by geometric mean when dealing with them. An analysis of the influence of sample parameter on the performance of several classification algorithms is also presented. We noted that usually a completely balanced dataset is better for RU and less is enough for oversampling techniques for most classification algorithms.

Due to computational cost, our analyses considered only 43 datasets, which have a maximum of around $1,000$ examples. The relatively small datasets may not represent big data tasks. Also, the pre-processing techniques are the most standard in the literature. In the next experiments, we shall evaluate the same analysis on the cutting edge methods. Lastly, we only considered binary classification tasks. Further investigation must be done for bigger and multi-class datasets.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors would like to thank CNPq, FAPESP (grant 2015/01382-0) and CAPES (grants 88887.162551/2018-00 and 99999.000555/2016-09 99999.000555/2016-09) for their financial support. The authors would like to thank CeMEAI-FAPESP (grant 2013/07375-0) for the computational resources and Intel for the hardware and software server used in part of the experiments.

CHAPTER

5

RECOMMENDING TECHNIQUES FOR IMBALANCED DATASETS USING META-LEARNING AND DATA COMPLEXITY MEASURES

Authors

Victor H. Barella *University of São Paulo, São Carlos, São Paulo, Brazil*

Nathalie Japkowicz *American University, Washington D.C., United States*

Luís P. F. Garcia *University of Brasília, Brasília, Distrito Federal, Brazil*

André de Carvalho *University of São Paulo, São Carlos, São Paulo, Brazil*

Abstract

Datasets with class imbalance are likely to impose difficulties in inducing classification models. Several techniques have been proposed to deal with this problem, such as data preprocessing, algorithm adaptation, ensemble methods, and one-class classification. Each technique has a different bias and performs better in some datasets and not in others. Since no technique works best for all datasets, selecting a good technique is fundamental. One approach to select techniques is through a recommendation system, whose main objective is to predict the user's score to an item. In this research, we implemented a recommendation system based on meta-learning, a subfield of machine learning that facilitates automatic learning based on metadata. Our system used a diverse set of meta-features, including the decomposed data complexity measures proposed specifically for imbalanced datasets, which had never been used as meta-features before. The system outputs a ranking of the methods according to the performances predicted by the

meta-regressors. We show that the decomposed data complexity measures are the most relevant meta-features in the system. The methods suggested by the system increased the predictive performance of the induced models compared to those suggested by a baseline indicating that taking the system into account on building machine learning pipelines has a high impact on their performance.

5.1 Introduction

A common issue in Machine Learning (ML) is having to learn from imbalanced datasets. A class labeled dataset is imbalanced when its classes have disproportionate number of examples. Although models induced from those datasets tend to perform poorly with respect to the minority classes, the problem is not caused by the imbalance alone. The problem arises when the imbalance is combined with other data characteristics, such as class overlap, difficult decision boundaries, small disjuncts, and noise (BATISTA; PRATI; MONARD, 2004; JO; JAPKOWICZ, 2004; LÓPEZ *et al.*, 2013; FRENAY; VERLEYSEN, 2014). The approaches to tackle the problem can be either by means of pre-processing (CHAWLA *et al.*, 2002; HE *et al.*, 2008) or adaptation on the learning steps of the classification algorithms (VEROPOULOS *et al.*, 1999; WANG; YAO, 2009). Due to its diverse nature, several techniques were proposed to mitigate its effect, but no technique works best for every dataset (FERNÁNDEZ *et al.*, 2018).

In order to find an adequate learning algorithm for a ML task, a greedy approach searches for a suitable combination of methods based on trial-and-error. Although it finds adequate ML methods, it has a high computational cost. Other approaches such as genetic programming and Bayesian optimization can help on reducing the search cost, but they still may be not fast enough for some applications.

One way to overcome the algorithm selection problem is through a recommendation system using meta-learning (MtL), which aims at predicting the best technique for a certain dataset based on previous experiences (RICE, 1976; SMITH-MILES, 2008; MUÑOZ *et al.*, 2018). To recommend a technique, usually a model is induced from a meta-dataset containing the datasets characteristics as predictive features and the technique that performed the best as target feature. For example, Morais et. al (MORAIS; MIRANDA; SILVA, 2016) implemented a system to recommend undersampling techniques. Furthermore, Smolyakov et al. (SMOLYAKOV *et al.*, 2019) implemented a recommendation system for sampling sizes for pre-processing strategies. More recently, Costa et al. (COSTA *et al.*, 2020) created a system able to recommend a variety of techniques, both for oversampling and undersampling. Nevertheless, little attention has been paid to a recommendation system able to recommend a broader range of techniques, including not only pre-processing techniques but algorithm-level, ensemble, and one-class classification techniques.

The selection of a proper set of meta-features is key to the success of a MtL system.

Statistical and landmarking meta-features (RIVOLLI *et al.*, 2018), as well as data complexity measures (DCMs) (HO; BASU, 2002; LORENA *et al.*, 2019; GARCIA *et al.*, 2020; ALCOBAçA *et al.*, 2020) are examples of commonly used meta-features in the literature. Although the DCMs have been successfully used as meta-features, it has been shown they do not properly measure imbalanced datasets (BARELLA *et al.*, 2020; BARELLA *et al.*, 2018). They were decomposed by class, which improved their ability to measure imbalanced datasets (BARELLA *et al.*, 2020). However, no research has approached the use of the decomposed DCMs in a MtL system.

In the present study, we evaluated a MtL system to recommend a wide range of techniques for imbalanced datasets. To implement it, we considered not only standard meta-features and the original DCMs, but also the decomposed DCMs. We showed that the system outperforms the baseline as the number of recommended techniques increases. Furthermore, through a meta-feature importance analysis, we showed that the decomposed DCMs are the most relevant meta-features for the system.

This paper is organized into five sections. Section 5.2 describes the concepts used in this work regarding MtL, meta-features, techniques for imbalanced datasets, and the related work. Next, Section 5.3 explains how the recommendation system was implemented, and the evaluation methods applied. In Section 5.4, we present and discuss the results obtained. Finally, Section 5.5 discusses the main contributions, limitations, and future work of this paper.

5.2 Background

In this section, we first introduce the MtL concepts. Secondly, we present the groups of traditional meta-features and DCMs considered in this work. Then, the pool of techniques for imbalanced datasets, which our MtL approach recommends, is described. Finally, we discuss the state-of-art of MtL for imbalanced datasets.

5.2.1 Meta-learning

Rice et al. (RICE, 1976) initially addressed the algorithm selection problem. In this study, the author proposed an abstract model to systematize the algorithm selection problem to predict the best algorithm when more than one algorithm is available. The main components in this model are the following: the problem instance space (P) composed of datasets; the instance feature space (F) based on the meta-features used to describe the datasets; the algorithm space (A) with the pool of ML algorithms that might be recommended; and the evaluation measure space (Y) responsible for assessing the performance of the ML algorithms in solving the problem instances contained in P. By using the previous sets, the MtL system can obtain an algorithm able to map a dataset x, described by the meta-features f, into one (or more) algorithm α able to solve the problem with an acceptable predictive performance according to Y, i.e., with maximum $y(\alpha(x))$

Smith-Miles et al. (SMITH-MILES, 2008) improved this abstract model by proposing generalizations that can also be applied to the algorithm design problem. This idea includes the following extra components: the set of MtL algorithms; the generation of empirical rules or algorithm rankings; and the examination of the empirical results, which may guide theoretical support to refine the algorithms.

More recently, Vanschoren et al. (VANSCHOREN, 2018) surveyed the MtL field based on three types of meta-data: (1) learning from model evaluations, (2) learning the relationships between data characteristics and predictive performance, and (3) transfer learning. Our work fits in the second category, and we describe the main components of this type of MtL approach on the following.

One crucial component of the previous models is the definition of the set of standard meta-features (F) used to describe the datasets' properties. These meta-features must be able to provide evidence about the algorithms' future performance in A (SOARES; PETRAK; BRAZDIL, 2001; REIF, 2012) and to discriminate, with a low computational cost, the performance of a group of algorithms. Rivolli et al. (RIVOLLI *et al.*, 2018) gathered the most used meta-features in the literature. We consider such meta-features in this paper. We also considered the DCMs (HO; BASU, 2002; LORENA *et al.*, 2019) and the decomposed DCMs (BARELLA *et al.*, 2020). In Section 5.2.2, we describe the meta-features used in this paper.

Defining the set of problem instances (P) is another concern. Ideally, a large number of diverse datasets should be used in order to induce a reliable meta-model. Unfortunately, it is not always possible due to the computational cost and/or availability of the datasets. Thus, in order to reduce the bias in this choice, datasets from different contexts should be retrieved from several data repositories, such as UCI[1] (DUA; GRAFF, 2017) and OpenML[2] (VANSCHOREN *et al.*, 2013). To the best of our knowledge, we considered the greatest amount of different datasets when compared to other works that tackled the recommendation problem of techniques for imbalanced datasets.

Muñoz et al. (MUÑOZ *et al.*, 2018) explored the problem of evaluating ML approaches on open repositories of datasets. According to them, most of those datasets usually do not pose a problem for some ML algorithms to induce a proper model. They visualized the datasets in a meta-feature space and implemented a method to fill the gaps in this space with new artificial datasets. Unfortunately, the method could not fill all the gaps in the meta-feature space, and it is necessary investigating further. In this paper, we used datasets from OpenML (VANSCHOREN *et al.*, 2013). Even though using datasets from open repositories has drawbacks, recommending ML methods for imbalanced datasets is still an open problem.

The algorithm space A represents a set of candidate algorithms recommended in the algorithm selection process. Ideally, these algorithms should also be sufficiently different and

[1] https://archive.ics.uci.edu/ml/index.php
[2] http://www.openml.org/

represent different regions in the algorithm space (MUÑOZ *et al.*, 2018). Different measures can evaluate the models induced by the algorithms. Although most of the studies in the MtL evaluate using accuracy, we considered gmean to take into account the imbalance of the datasets. We also considered several classification algorithms and techniques for imbalanced datasets with different biases. To the best of our knowledge, we are the only work to tackle both pre-processing and algorithmic level recommendation on imbalanced datasets. The performance of the techniques were calculated by means of gmean metric.

After extracting the meta-features and evaluating a set of algorithms' performance, the next step is labeling each meta-example in the meta-base. Brazdil et al. (BRAZDIL *et al.*, 2009) summarize the three main properties frequently used to label the meta-examples in MtL: (*i*) the algorithm that presented the best performance on the dataset (a classification task); (*ii*) the ranking of the algorithms according to their performance on the dataset (a ranking classification task), where the algorithm with the best performance is top-ranked; and (*iii*) the performance value obtained by each evaluated algorithm on the dataset (a regression task). To implement our MtL system, we labeled the meta-dataset considering the latter.

5.2.2 Meta-features

In our MtL system, we considered the traditional meta-features (RIVOLLI *et al.*, 2018), the DCMs (HO; BASU, 2002; LORENA *et al.*, 2019), and the decomposed DCMs (BARELLA *et al.*, 2020).

5.2.2.1 Traditional meta-features

The main standard meta-features used in the MtL literature can be divided 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 named *general* measures (CASTIELLO; CASTELLANO; FANELLI, 2005).

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

- **Information-theoretic:** meta-features based on information theory (CASTIELLO; CASTELLANO; FANELLI, 2005), usually entropy estimates (SEGRERA; PINHO; MORENO, 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 (DT) models (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000; PENG *et al.*, 2002), 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.

- **Others:** standalone, time-related, concept and case-based meta-features (VANSCHOREN *et al.*, 2012; MUÑOZ *et al.*, 2018), clustering and distance-based measures (VUKICEVIC *et al.*, 2016; PIMENTEL; CARVALHO, 2019), among others. These describe characteristics that do not fit into the other groups.

Although the traditional meta-features play an important role on general MtL systems, it is usually necessary to add domain specific meta-features in order to achieve a good MtL performance. Therefore, we also considered measures designed specifically for imbalanced datasets. Next, we discuss the DCMs and their adaptation for imbalanced datasets.

5.2.2.2 Data Complexity Measures

The DCMs were proposed to assess the difficulty in a training set (HO; BASU, 2002). They were extended by many studies (HO; BASU; LAW, 2006; ORRIOLS-PUIG; MACIá; HO, 2010; LORENA; de Souto, 2015; LORENA *et al.*, 2019). A package called DCoL (Data Complexity Library) popularized and implemented generalizations of DCMs for multiclass problems (ORRIOLS-PUIG; MACIá; HO, 2010). Some limitations of the package were solved (LORENA *et al.*, 2019), and they were standardized and implemented in a revised R package called ECoL (Extended Complexity Library) (GARCIA; LORENA, 2018). They were then adapted for the imbalance problem by decomposing the DCMs for each class separately (BARELLA *et al.*, 2020), which we call here the decomposed DCMs.

The measures can be classified into three different categories: overlapping, neighborhood, and linearity. These categories are described below. For more details of the measures mentioned below, check (LORENA *et al.*, 2019) for the DCMs and (BARELLA *et al.*, 2020; BARELLA *et al.*, 2018) for the decomposed DCMs. We also considered a geometric mean of the decomposed measures (BARELLA; GARCIA; CARVALHO, 2019).

- **Feature overlapping measures**

  The feature overlapping measures assess the discrimination power of the predictive attributes. Most of them evaluate the features individually and the feature that discriminate the most is selected, while others use a combination of the individual feature assessments. The overlapping DCMs considered in this article are F1, F2, F3, and F4. The decomposed DCMs are the decomposed versions of F2, F3, and F4.

- **Neighborhood measures**

  The neighborhood measures use the concept of Nearest Neighbor (NN) to assess classification difficulty. They use the distance between instances to assess, for example, the shape of decision boundaries and class distributions. This paper considered the measures N1, N2, N3, N4, T1, and their decomposed versions.

- **Linear Separability Measures**

  These measures assess whether the classes can be linearly separable in the attribute space. They assume that a classification problem solved with a hyperplane is simpler than another with a non-linear boundary. The measures from this category considered in this article are L1, L2, L3, and their decomposed versions.

In this paper, we used these DCMs as meta-features in an MtL system, and we performed a time analysis. We also performed a meta-feature importance analysis.

Although the DCMs showed to be useful for different applications, their computational cost might prevent them from being used in time-restricted applications. For such applications, estimations of the measures can be used (GARCIA *et al.*, 2020; BARELLA; GARCIA; CARVALHO, 2020). For this paper, we applied the precise calculation of the DCMs

5.2.3 Techniques for imbalanced datasets

The techniques for imbalanced learning are usually separated into two main general approaches: (1) pre-processing the data in order to make it more balanced (CHAWLA *et al.*, 2002; HAN; WANG; MAO, 2005; HE *et al.*, 2008; JO; JAPKOWICZ, 2004; KUBAT; MATWIN *et al.*, 1997); and (2) development of algorithms in the classification step that are more tolerant and robust to handle imbalanced data (WANG; YAO, 2009; VEROPOULOS *et al.*, 1999; LIU; TING; ZHOU, 2008). In this work, we built a MtL system able to recommend a pool of techniques, including pre-processing and algorithm-level techniques. Next, we describe the techniques considered here.

5.2.3.1 Pre-processing

Regarding the pre-processing techniques, the methods can be categorized into two groups: undersampling and oversampling methods (FERNÁNDEZ *et al.*, 2018). Undersampling methods make the data more balanced by removing instances of the majority class, while oversampling methods add instances to the minority class (FERNÁNDEZ *et al.*, 2018). Both undersampling and oversampling can be performed randomly or according to an informed strategy.

In the random undersampling (RU), some majority class instances are removed at random until a more balanced class distribution is reached (HE; GARCIA, 2008). In the random oversampling (RO), some minority class instances are replicated at random until a more balanced class distribution is reached (HE; GARCIA, 2008).

The *Synthetic Minority Oversampling Technique* (SMOTE) (CHAWLA *et al.*, 2002) generates artificial data by interpolation, as follows. At each iteration, SMOTE selects an instance x at random in the minority class, and then it looks for the k NNs of x. SMOTE then selects one of the neighbors z at random and creates a new instance, a combination of x and z. The combination is an interpolation that randomly creates any possible point between x and z. This step is repeated until a more balanced distribution of instances is reached.

The *Borderline SMOTE* (BSMOTE) is based on SMOTE, and it searches examples from the decision boundaries (HAN; WANG; MAO, 2005). Thus, instead of interpolating minority examples from the entire training set, it selects minority examples from the decision boundaries. The method that BSMOTE uses to select the examples from the boundaries is: (1) find the k NN for a minority example x; (2) count the number N_{maj} of neighbors that belongs to the majority class; (3) if $\frac{k}{2} \leq N_{maj} < k$ then x is put in a set called DANGER; (4) repeat the steps for all minority examples. Next, SMOTE is run to balance the dataset, selecting only examples from the DANGER subset.

The *Adaptive Synthetic Sampling Approach* (ADASYN) is also based on SMOTE, and it addresses the number of examples to be interpolated considering each minority example (HE *et al.*, 2008). ADASYN follows the next steps: (1) first, it defines G, that indicates how many examples should be interpolated for the entire minority class; (2) next, for each example in the minority class, it calculates the percentage of majority examples in the k nearest neighbors; (3) the set of all percentages (Γ_i, where i is the minority example) is normalized so that $\sum \Gamma_i = 1$; finally, $\Gamma_i \times G$ gives the number of examples to be interpolated using SMOTE for each minority example i.

5.2.3.2 Algorithm-level

While pre-processing techniques modify the data to improve the predictive performance on imbalanced datasets, algorithm-level techniques modify the learning step itself. Here, we considered three approaches: class weight, one-class learning, and ensemble.

Class weights are used during the models' induction, usually to give different penalties for the errors depending on the class. Each classification algorithm can be implemented in different ways to take into consideration the class weights. In this paper, we consider two implementations considering class weights. The first is the SVM considering one loss function for each class, attaching a different weight for each of them (VEROPOULOS *et al.*, 1999). The second is the Random Forest (RF) when the classes' priors are given (BREIMAN, 2001). We did not tune these hyper-parameters. Instead, we applied a rule of thumb where the weight for the majority class is 1 and the weight for the minority class is $\frac{N_{maj}}{N_{min}}$, where N_{maj} is the number of majority instances and N_{min} is the number of minority instances in the training set.

One-class classification, also known as learning in the absence of counterexamples, is an outlier identification approach used in imbalanced datasets (FERNÁNDEZ *et al.*, 2018). In this approach, a model can be induced to identify the majority class, which is the better-represented class and outputs everything else as outliers. This technique is beneficial when the minority class lacks structure, and noisy examples and small disjuncts compose it. In this paper, we considered the one-class classification approaches based on SVM and RF, respectively, One-Class SVM (SCHÖLKOPF *et al.*, 2001) and Isolation Forest (LIU; TING; ZHOU, 2008).

Adaboost is a well known adaptative ensemble method that at each iteration favors the instances misclassified by the previous classifiers (FREUND; SCHAPIRE, 1997). Adaboost can mitigate the effects of imbalanced datasets, but other ensemble techniques, designed specifically for imbalanced data, include a pre-processing step. In this paper, we considered the combination of RU and SMOTE with Bagging and Boosting, which are Under Bagging (BARANDELA; VALDOVINOS; SÁNCHEZ, 2003), SMOTE Bagging (WANG; YAO, 2009), Under Boosting (SEIFFERT *et al.*, 2009), and SMOTE Boosting (CHAWLA *et al.*, 2003).

5.2.4 Meta-learning to recommend techniques for imbalanced datasets

Several techniques to mitigate the effect of imbalanced classes on classification tasks were proposed. None of them works for all cases, which leads interest to a recommendation system capable of selecting a technique for a specific dataset. Next, we present the state-of-art of the MtL system to recommend techniques for imbalanced datasets. We detail the main differences between our work and the state-of-art, and also show a general comparison of them on Table 21.

To the best of our knowledge, (MORAIS; MIRANDA; SILVA, 2016) proposed the first MtL approach to recommend undersampling techniques. They investigated the recommendation of hyper-parameters for those techniques. They only considered SVM as the classification algorithm and undersampling techniques. The meta-learner was based on KNN and in only 29 datasets. In our work, we considered a larger number of datasets, a larger pool of classification algorithms, and techniques for imbalanced datasets, as well as more regression algorithms for the meta-models.

(SMOLYAKOV *et al.*, 2019) evaluated a MtL approach to suggest both over and undersampling on approximately 100 real datasets. Although their approach had a remarkable performance on artificial datasets, they did not achieve the same results in real datasets. They suggested that increasing the datasets' diversity and the quality of the meta-features might improve the performance of the MtL. In our work, we considered more datasets and more meta-features.

(BORSOS; LEMNARU; POTOLEA, 2018) investigated a new measure to assess overlapping in imbalanced datasets and applied it as a meta-feature on an MtL system to suggest classification algorithms. Although the results are compelling, they used datasets from the KEEL repository, a binary decomposition of multiclass datasets. In this repository, one multiclass dataset could be decomposed into several binary datasets. For example, the Glass dataset was decomposed into 13 datasets. In that case, during the evaluation of the MtL system using a random cross-validation approach, the training datasets must contain information about the testing datasets. To properly evaluate an MtL system, it is advisable not to include information about training datasets in testing datasets. To avoid impairing the proper evaluation of the MtL system, we manually removed datasets explicitly from the same context. As future work, the overlapping measure proposed by them should be considered as a meta-feature.

(COSTA *et al.*, 2020) analyzed the relation between meta-features and whether or not it is beneficial to apply a particular pre-processing technique. Although they found interesting relations, the analysis could have been improved by using the decomposed DCMs. They also focused on pre-processing techniques, not considering algorithm-level approaches.

Table 21 compares the general characteristics of this work and the works aforementioned. Each line represents a characteristic, including the reference, year of publication, number of real datasets used in the experiments, presence of pre-processing techniques for imbalanced datasets, presence of algorithmic-level techniques for imbalanced datasets, classification algorithms used, quantity of traditional meta-features considered, quantity of DCMs considered as meta-features, quantity of measures specifically designed for imbalanced datasets considered as meta-features. Each column represents a different paper, including this work. They show recent interest in the scientific community in applying MtL to recommend techniques for imbalanced datasets. Differently from what has been done, we recommended both pre-processing and algorithm-level approaches. We also considered a set of meta-features explicitly designed for the imbalance problem, and we show their relevance in the meta-models induced.

5.3 Materials and Methods

The purpose of this study is to answer the following questions:

1. Can meta-regressors predict the performance of pre-processing and algorithmic approaches for imbalanced datasets?
2. Does a meta-learning approach recommend both pre-processing and algorithm-level approaches for imbalanced datasets better than the baselines?
3. Were the decomposed DCMs important meta-features for the meta-models?

4. Is the meta-learning approach time efficient in comparison with performing a brute force approach?

Table 21 – Comparison of the state-of-art on MtL for imabalnced datasets and this work

Year	2016	2017	2018	2020	2021 (this paper)
Number of datasets	29	100	66	163	162
Pre-proc. techinques	Yes	Yes	Yes	Yes	Yes
Alg.-level techinques	No	No	No	No	Yes
Classification algorithms	SVM	Adaboost with DT	SVM, DT	SVM	SVM, RF, KNN, NB, DT, MLP, JRip, Adaboost
Traditional meta-features	Some	Some	Some	Several	Several
Traditional DCMs	None	None	One	None	Several
Meta-features for imbalanced datasets	None	None	One	None	Several

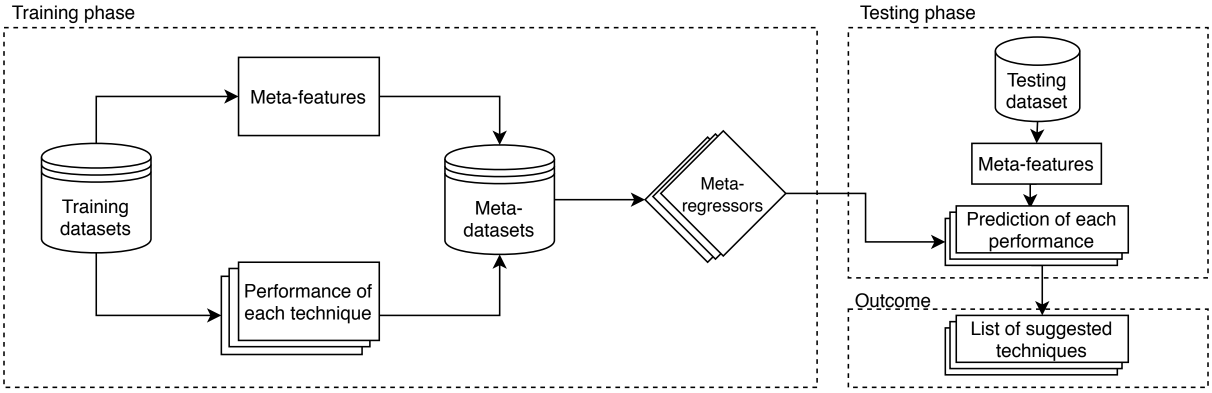
In order to answer them, a MtL system was implemented to predict a wide range of techniques for imbalanced datasets, including pre-processing and algorithm-level approaches. The system was compared to baselines on a meta-level, a base level, and time elapsed analyses. A meta-feature importance analysis was also performed.

First, for each training dataset, the meta-features were extracted, and the gmean performance of each technique was repeatedly calculated and averaged considering 10 times 5-fold cross-validation. The meta-datasets were formed by combining all training datasets as instances, the meta-features as predictive attributes, and the gmean performances as target values. Therefore, one meta-dataset was built for each technique. Then, for each meta-dataset, a meta-regressor was induced to predict the performance of that technique. The meta-features of a separated testing dataset were extracted and given to each meta-regressor as input. Finally, the predictions of the meta-regressors were ranked and a list of the best techniques was returned as output from the system. Figure 22 represents the steps described.

In order to evaluate the system, 162 datasets were collected from OpenML. Duplicated datasets and versions from the same datasets were manually identified and removed. A leave-one-out strategy was applied to define the training and testing datasets. The meta-features considered were all the ones described in section 5.2.2, available in the R pacakges mfe[3], ECoL[4]

[3] https://github.com/rivolli/mfe
[4] https://github.com/lpfgarcia/ECoL

Figure 22 – Representation of the implemented meta-learning system



Source: Elaborated by the author.

and `ImbCoL`[5]. The techniques considered were all the ones described in section 5.2.3, and summarized in Table 22. For each technique, a set of classification algorithms was applied, and all possible combinations summed 110 possible recommendations. Each combination's gmean performance was turned into a target value on a meta-dataset, a total of 110 meta-datasets. The meta-regressors were induced by DWNN, LASSO, RF, and SVR regression algorithms. As a baseline approach, the medians of the target values were predicted, ignoring the meta-features.

We opted to deal with the MtL task by combining a meta-regression approach with a ranking approach. Although a meta-classification task would be more straightforward, some tricky decisions would be necessary, such as choosing the best method for each dataset even when there is no clear winner. We remove this decision by considering a meta-regression approach, which facilitates the system to recommend a method with similar performance to the best performance achieved.

5.4 Results and Discussion

In this section, we evaluate the Mtl approach to recommending different techniques for imbalanced datasets. Generally speaking, we show that the meta-regressors' error is lower than the baseline in most cases in a meta-level analysis. Moreover, we show in a base-level analysis that the recommendations provided by the MtL system perform considerably better than the ones recommended by the baseline as the list of recommended techniques becomes larger. For two regression algorithms, we provide an analysis of the most important meta-features in which the decomposed DCMs are particularly positioned at the top of the meta-features ranking. Furthermore, we show that the system's application has a lower computational cost for the datasets with lower dimensionality.

[5] https://github.com/victorhb/ImbCoL

Table 22 – Description of the imbalance treatment techniques used

Name	Pre-processing	Algorithmic level	Classification Algorithms
Random Undersampling	✓		SVM, RF, KNN, NB, DT, MLP, JRip
Random Oversampling	✓		SVM, RF, KNN, NB, DT, MLP, JRip
SMOTE	✓		SVM, RF, KNN, NB, DT, MLP, JRip
Borderline SMOTE	✓		SVM, RF, KNN, NB, DT, MLP, JRip
ADASYN	✓		SVM, RF, KNN, NB, DT, MLP, JRip
Class weight		✓	SVM, RF
One class classification		✓	SVM, RF
AdaBoost		✓	SVM, DT
UnderBagging	✓	✓	SVM, DT
SMOTE Bagging	✓	✓	SVM, DT
UnderBoosting	✓	✓	SVM, DT
SMOTE Boosting	✓	✓	SVM, DT

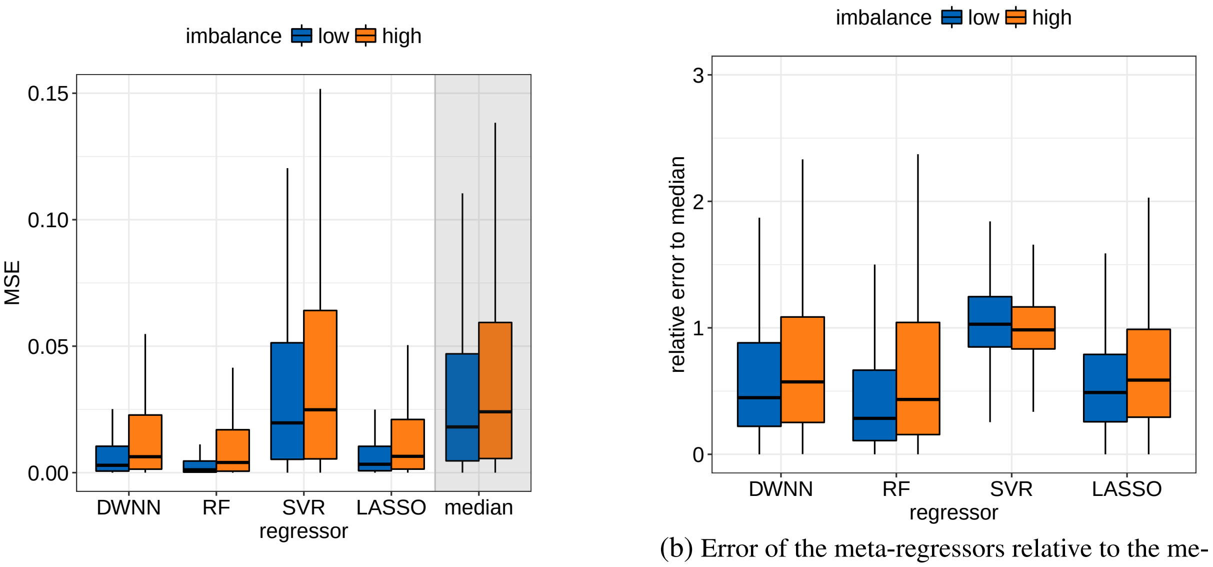
5.4.1 Meta-level analysis

We evaluated each individual meta-regressor according to two analyses, presented in Figure 23. The x-axis represents the regression algorithm used to induce each meta-regressor and the median as the baseline. Instead of using a regressor to predict a method's performance, the median approach takes the median value of performances in the training set as the prediction. In the meta-level analysis, the median approach is a baseline that represents the expected performance of each technique independently of the datasets' characteristics. That means that an informed system recommending based on datasets' characteristics must perform better than the median approach to justify its computational cost. The colors represent the datasets' imbalance level, where high imbalance corresponds to less than 25% of minority class instances in the dataset, and low imbalance corresponds to more than 25% instances of the minority class. Each boxplot represents the error of each meta-regressor on each dataset under a leave-one-out evaluation. In other words, each pair of boxplots shows *110 methods* × *162 datasets = 17,820 different values*.

Figure 23a shows the mean squared error (MSE) prediction on the y-axis. RF was the best regressor algorithm of all. The performance of SVR was similar to the baseline. DWNN, RF, and LASSO showed lower MSEs compared to the baseline.

The relative error of the meta-regressors compared to the median method is shown on the y-axis in Figure 23b. Thus, the relative error is the fraction of the meta-regressor error over the median approach error. Values greater than 1 mean the median approach performed better than the meta-regressor, and values lower than 1 mean the meta-regressor performed better. DWNN, RF, and LASSO performed substantially better than the median approach for most cases, while SVR performed similarly in the median.

Figure 23 – Meta-level analysis of the meta-regressors



(a) Mean squared error of the meta-regressors

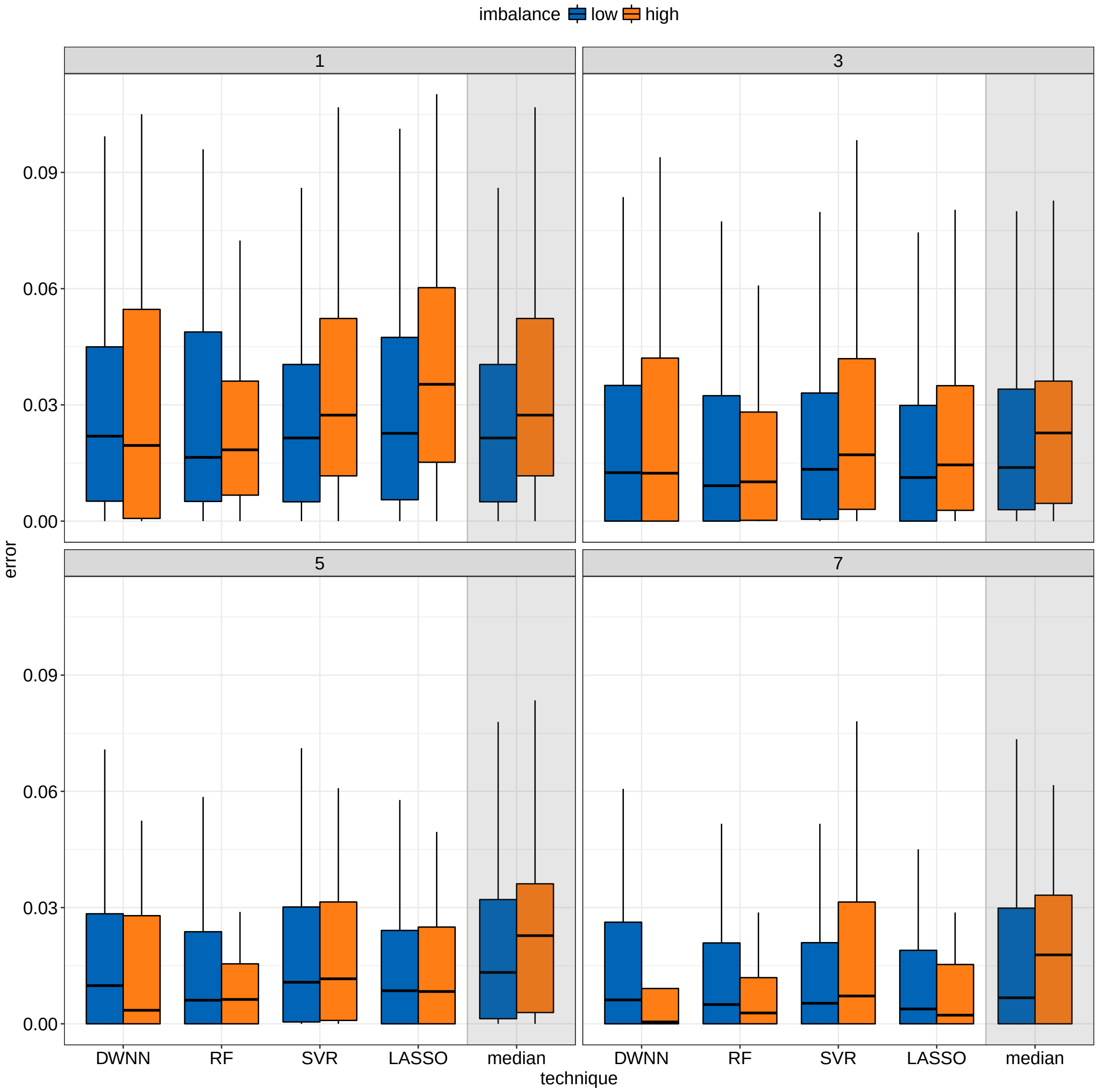
(b) Error of the meta-regressors relative to the median

Source: Elaborated by the author.

5.4.2 Base-level analysis

The base-level analysis considers the error of the techniques suggested by the MtL system. Figure 24 shows the results for the base-level analysis. The x-axis represents the recommendation approach, whether using meta-regressors or the median approach. On the y-axis, we show the difference between the upperbound baseline performance, which is the best performance achieved for each dataset on the experiments, and the recommended method's performance. Each panel represents a different number of recommended methods, where 1 means that the approaches recommended only the method with the highest predicted performance, 3 means the approaches recommended the top 3 methods with the highest predicted performance, etc. Independently of the number of recommended methods, only the best performance among the recommended is shown in the figure. As the number of recommended methods increased, the error of all approaches decreased. The MtL approaches performed better than the median approach, especially for the datasets with a high imbalance level, when recommending more methods. N.B., when recommending only one method, the meta-regressors' error is frequently similar to the error of the median. It seems that due to the large number of possible methods, it is advisable to allow the system to suggest more methods to find an adequate one.

Figure 24 – Base-level analysis



Source: Elaborated by the author.

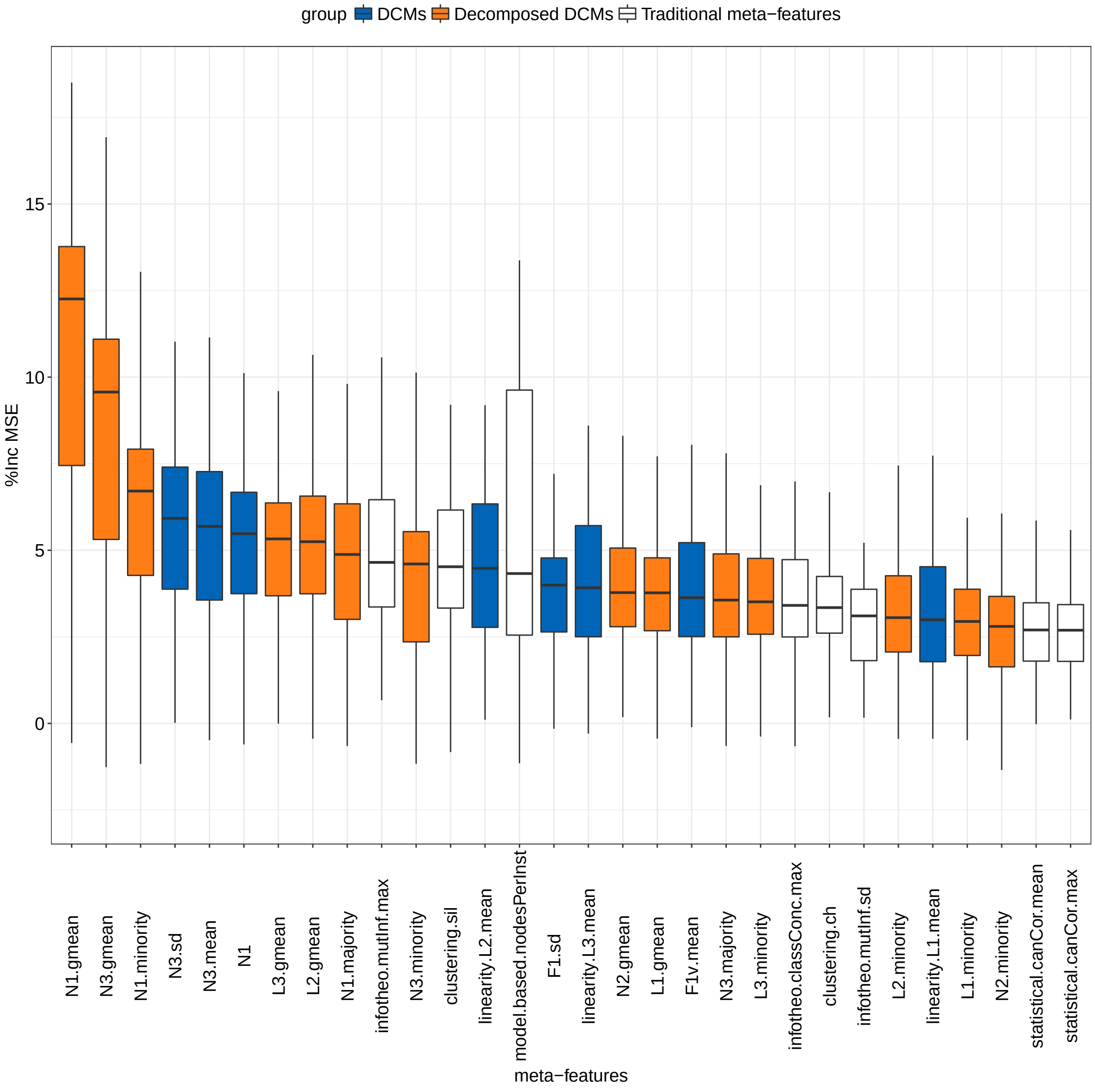
5.4.3 Meta-feature importance analysis

We also investigated what meta-features are relevant to the meta-regressors. RF and LASSO allow a better interpretation of the features used. We are interested in investigating the relevance of the decomposed DCMs, which we previously proposed and investigated.

Figure 25 shows the feature importance for RF. On the x-axis, the 30 meta-features with the highest values of importance are shown. Axis-y shows the percentage of increment of MSE when that feature is not used. The orange boxplots show the meta-feature previously proposed by us, named as decomposed DCMs, the blue boxplots show the original DCMs, and the white boxplots show the traditional meta-features. The three most relevant features are decomposed DCMs. Although the decomposed DCMs represented only 12% of all meta-features, they represented 47% of the top meta-features shown in the figure. The results are in accordance with previous publications, where we showed the same decomposed DCMs as the most correlated with the class imbalance problem (BARELLA *et al.*, 2018).

The most important meta-features are based on N1 and N3. The former builds a Minimum Spanning Tree of the dataset and measures the proportion of examples from different classes that are connected. The latter uses a nearest neighbour approach to measure how difficult it is to learn from this training set.

Figure 25 – Random Forest feature importance analysis

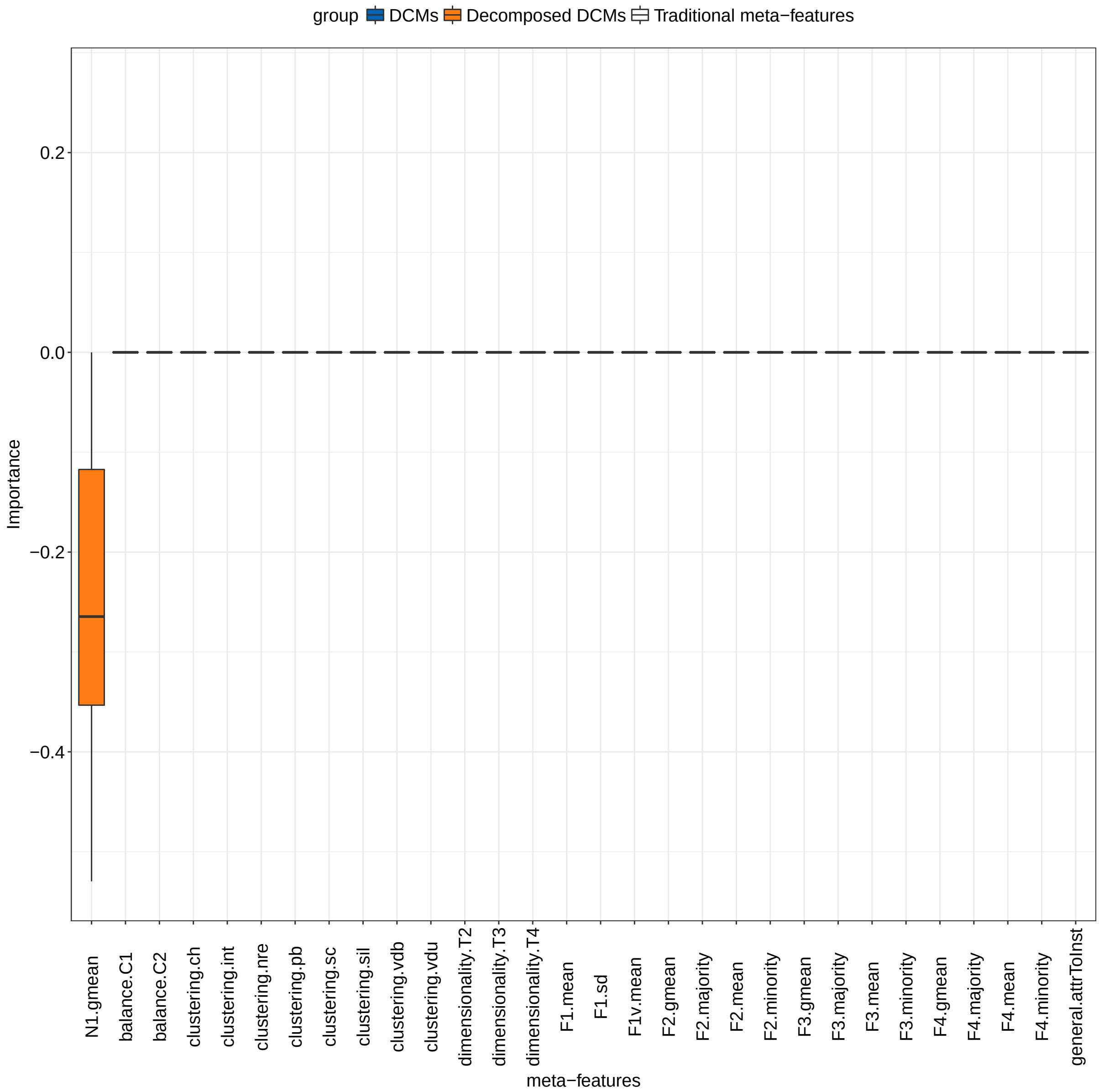


Source: Elaborated by the author.

Figure 26 shows the feature importance analysis for LASSO. On the x-axis, we show the meta-features, and on the y-axis, the feature importance for the LASSO algorithm. Since

LASSO computes a linear combination of the features, values greater and lower than 0 are evenly important, while values close to 0 represent negligible importance. For LASSO's models, one decomposed DCM, which is N1.gmean, was by far the most important meta-feature, while the others had importance virtually zero. The same one was also the most important to RF. It is important to stress that although LASSO's models seem less complex, they performed worse in comparison with RF.

Figure 26 – LASSO feature importance analysis

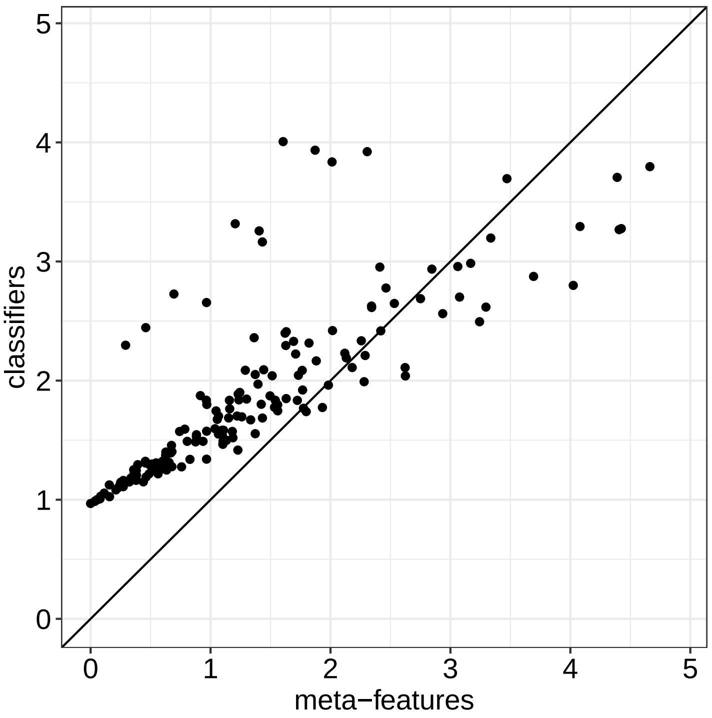


Source: Elaborated by the author.

5.4.4 Time analysis

Figure 27 shows the relation between calculating all meta-features and calculating all the performances of the classifiers using the methods considered. For most of the datasets, it is faster to calculate the meta-features than the classifiers' performances. For some more complex datasets, the meta-features are more time-consuming. This resulted from the computational cost for some of the DCMs and decomposed DCMs that are high depending on the dataset's size. Meta-features are planned to be fast and roughly capture characteristics of the datasets, while data complexity measures are planned to precisely assess a characteristic's complexity. Therefore, to use data complexity measures as meta-features may require some adaptations to reduce computational cost/time for large datasets. One way to overcome this is by estimating the value of the DCMs and the decomposed DCMs (BARELLA; GARCIA; CARVALHO, 2020).

Figure 27 – Time elapsed on calculating the meta-features vs time elapsed on inducing all classifiers.



Source: Elaborated by the author.

5.5 Conclusions

In this work, we investigated a system to recommend a large and diverse group of techniques for imbalanced datasets. The system had a low error rate in terms of suggesting the techniques as the number of recommendations increased. Our results have shown that the MtL approach can suggest techniques with different biases. In addition, decomposed DCMs play a significant role as meta-features as they are the most important ones in the system.

Although other works have addressed the problem of recommending techniques for imbalanced datasets using MtL, to the best of our knowledge, no work has approached both pre-processing and algorithm-level techniques simultaneously. The use of the decomposed DCMs as meta-feature is also a novelty of our work.

Our results are encouraging, and more and more complex datasets should be incorporated into the system. The computational cost of calculating the DCMs and the decomposed DCMs may become a problem in larger datasets. Future work should focus on reducing the computational cost of those measures maintaining their efficacy.

This approach has potential in areas such as recommendation systems, MtL, and end-to-end machine learning. It could also be used by any data scientist dealing with imbalanced datasets as a recommendation system for techniques.

CHAPTER

6

SIMULATING COMPLEXITY MEASURES ON IMBALANCED DATASETS

Authors

Victor H. Barella *University of São Paulo, São Carlos, São Paulo, Brazil*

Luís P. F. Garcia *University of Brasília, Brasília, Distrito Federal, Brazil*

André de Carvalho *University of São Paulo, São Carlos, São Paulo, Brazil*

Abstract

Classification tasks using imbalanced datasets are not challenging on their own. Classification models perform poorly on the minority class when the datasets present other difficulties, such as class overlap and complex decision border. Data complexity measures can identify such difficulties, better dealing with imbalanced datasets. They can capture information about data overlapping, neighborhood, and linearity. Even though they were recently decomposed by classes to deal with imbalanced datasets, their high computational cost prevents their use on applications with a time restriction, such as recommendation systems or high dimensional datasets. In this paper, we use a Meta-Learning approach to estimate the decomposed data complexity measures. We show that the simulated measures assess the difficulty of the dataset after applying preprocessing techniques to different sample sizes. We also show that this approach is significantly faster than computing the original measures, with a statistically similar estimation error for both classes.

6.1 Introduction

In Machine Learning (ML), standard classification algorithms tend to perform poorly on classes less represented on the training set. This problem is called the imbalanced data problem (FERNáNDEZ *et al.*, 2018). Several approaches have been proposed in the literature to mitigate the effects of such problem, some concerning preprocessing the training data to make it more balanced, others adapting standard classification algorithms to consider the imbalance on the learning or prediction steps, and others may combine both strategies (CHAWLA *et al.*, 2002; HE *et al.*, 2008; GONZALEZ-ABRIL *et al.*, 2014; CANO; ZAFRA; VENTURA, 2013). No technique performs well in all datasets, and their performance will depend on each dataset characteristics.

Data Complexity Measures (CMs) were proposed to assess dataset characteristics, such as data overlapping, neighborhood, linearity, and decision border complexity (HO; BASU, 2002; LORENA *et al.*, 2019). Their adaptations for imbalanced datasets are useful for understanding the imbalance problem and the techniques in the literature, as they correlate with the difficulty in imbalanced datasets and sampling sizes of preprocessing techniques (BARELLA *et al.*, 2018; BARELLA; GARCIA; CARVALHO, 2019). One disadvantage is that they have a high computational cost, making them unappropriated in approaches with time restrictions such as Meta-Learning (MtL), genetic algorithms, and iterative ones.

To overcome this challenge, we propose a MtL approach to estimate the data CM for imbalanced datasets. A MtL approach learns from previous experiences, considering, for example, previous applications of techniques on different datasets (SMITH-MILES, 2008; BRAZDIL *et al.*, 2009). A meta-dataset is usually created, in which each meta-instance represents a dataset, and each meta-feature represents a dataset characteristic. The approach recommends the target-feature, which can be algorithms, their performance, or a ranking of algorithms (MUÑOZ *et al.*, 2018). A MtL approach can induce a model to predict the performance of a technique on a dataset based on the dataset characteristics by using a meta-dataset.

In this work, we show that a MtL approach can estimate the CMs with a small predictive error for imbalanced datasets using regressor techniques and standard meta-features. This evaluation considers the CM for both classes, the positive (P) and negative (N). We also show that our approach has a low computational cost, which is faster than calculating the original CMs. We show that the simulated measures are as useful as the original ones on an analysis with real datasets and preprocessing techniques on different balance ratios. We also make available the models in an R package called `SImbCoL`[1].

This paper is separated into five sections. Section 6.2 describes the CMs used to estimate the difficulty of each class separately. Moreover, we present the main concepts about MtL and describe the standard meta-features used to predict CMs values. Next, Section 6.3 presents the

[1] https://github.com/victorhb/SImbCoL

experimental setups designed in this work. The experimental results are shown and discussed in Section 6.4. Section 6.5 concludes this paper with contributions, limitations and future works.

6.2 Background

This section presents the background information to describe the proposed approach: Section 6.2.1 describe the main concepts regarding data CMs and Section 6.2.2 introduces the MtL framework, including the process of building a meta-dataset and how to recommend algorithms.

6.2.1 Data Complexity Measures

The CMs were proposed to assess the difficulty in a training set (HO; BASU, 2002). They were extended by many studies (HO; BASU; LAW, 2006; ORRIOLS-PUIG; MACIá; HO, 2010; LORENA; de Souto, 2015; LORENA *et al.*, 2019). A package called DCoL (Data Complexity Library) popularized and proposed generalizations of CMs for multiclass problems (ORRIOLS-PUIG; MACIá; HO, 2010). Some limitations of the package were solved (LORENA *et al.*, 2019), and they were standardized and implemented in a revised R package called ECoL (Extended Complexity Library) (GARCIA; LORENA, 2018). They were adapted for the imbalance problem by a decomposition strategy measuring the CM for each class separately (BARELLA *et al.*, 2018).

The measures can be classified into three different categories: overlapping, neighborhood, and linearity. Such categories are described below.

6.2.1.1 Feature overlapping measures

The feature overlapping measures assess the discrimination power of the predictive attributes. Most of them evaluate the features individually and the most discriminate feature is selected, while others use a combination of the individual feature assessments. The overlapping measures considered in this article are F2, F3, and F4.

- **F2: Volume of overlap region.** F2 computes the volume of the classes' overlapping region using the minimum and maximum values of each input attribute per class. If the attribute ranges overlap in a certain region, this region is considered ambiguous for the attribute. Next, a product of the normalized size of the ambiguous regions for all attributes is output. For example, suppose an attribute with values for class 1 between 0 and 1, and the values for class 2 between 0.75 and 1.25. Taking the previous example, F2 for class 1 would be $\frac{0.25}{1} = 0.25$ and F2 for class 2 would be $\frac{0.25}{0.5} = 0.5$.

- **F3: Feature efficiency.** In F3, one feature is considered efficient, depending on how many examples are not in an ambiguous region. For each attribute, the number of examples from the class of interest out of the ambiguous region is divided by the total number of examples from the class of interest. Then, the maximum of such values among all the input attributes is calculated, which corresponds to the attribute that separates better. F3 is $1-$ the maximum value calculated.

- **F4: Collective feature efficiency.** F4 uses the main concept of F3, but instead of getting the maximum value from all attributes, it combines their discrimination power. First, the most discriminative attribute, according to F3, is found; next, the examples correctly separated by that attribute are removed. The previous steps are repeated until all examples are correctly discriminated or until all attributes are removed. F4 is the proportion of examples not discriminated at the end of the process.

6.2.1.2 Neighborhood measures

The neighborhood measures use the concept of Nearest Neighbor (NN) to assess classification difficulty. They use the distance between instances to assess, for example, the shape of decision boundaries and class distributions. In this paper, we considered the measures N1, N2, N3, N4, and T1.

- **N1: The fraction of points on the class boundary.** N1 builds a minimum spanning tree (MST) that connects all the examples from a dataset based on their distances, despite their classes. Next, it counts the number of examples connected to at least one example from another class. Those examples are considered borderline. The fraction of the number of borderline examples for each class over the size of each class is the final decomposed N1 measure.

- **N2: The ratio of average intra/inter class NN distance.** N2 compares the intraclass and interclass dispersions of the classes. For each example, its distance from the NN of the same class (intraclass) and its distance to the NN of a different class (interclass) are computed. Decomposed N2 is the ratio of the average of the intraclass distances for each class and the average of the interclass distances for each class.

- **N3: Leave-one-out error rate of the 1NN classifier.** N3 gives the leave-one-out training error of a nearest-neighbor classifier, which is easy to be calculated and is a good indicator of the separability of the classes. The decomposed N3 is the error rate per class.

- **N4: Nonlinearity of a 1-NN classifier.** N4 uses a method that creates a new test set by interpolating two randomly selected examples from the same class multiple times. Then an NN classifier using the training set is used to predict the labels of the examples in the interpolated test set. Decomposed N4 gives the error rate per class achieved in this procedure.

- **T1: Fraction of maximum covering spheres.** T1 tries to explain the training set with hyper-spheres. Suppose that every example in the training set has a hypersphere with radius zero. If we gradually increase the radius of all hyperspheres, some will touch a hypersphere from a different class. When that happens, both hyperspheres stop growing. The method stops when there is no more growing hypersphere. The hyperspheres that are contained in another hypersphere are discarded. Decomposed T1 is the ratio between the number of remaining hyperspheres for each class and the number of examples in each class.

6.2.1.3 Linear Separability Measures

These measures assess whether the classes can be linearly separable in the attribute space. They assume that a classification problem solved with a hyperplane is simpler than another with a non-linear boundary. The measures from this category considered in this article are L1, L2, and L3.

- **L1: The minimized sum of error distance of a linear classifier.** In L1, one linear model (e.g., a linear SVM) is built using the training dataset and calculating the distances of erroneous instances to the obtained hyperplane. Decomposed L1 is the sum of these distances per class. L1 is equal to 0 for linearly separable problems.

- **L2: The training error of a linear classifier.** Decomposed L2 is the training error of a linear classifier per class. Higher values are expected for non-linear separable classes.

- **L3: Nonlinearity of the linear classifier.** L3 is based on the same method of N4. A test set is interpolated, and instead of an NN classifier, N3 uses a linear classifier to predict the labels of the examples from the test set.

Although the data CMs showed to be useful for different applications, their computational cost may prevent them from being used on applications that have time restriction. To overcome this, we suggest in this paper to estimate them using a MtL approach.

6.2.2 Meta-learning

Rice, J. (1976) (RICE, 1976) initially addressed the algorithm selection problem. In this study, the author proposed an abstract model to systematize the algorithm selection problem to predict the best algorithm when more than one algorithm is available. The main components in this model are the problem instances space (P) composed by datasets, the instance features space (F) based on the meta-features used to describe the datasets, the algorithms space (A) with the pool of ML algorithms that might be recommended, and the evaluation measures space (Y) responsible for assessing the performance of the ML algorithms in solving the problem instances contained in P. By using the previous sets, the MtL system can obtain an algorithm able to map a dataset x, described by the meta-features f, into one (or more) algorithm α able to solve the problem with an acceptable predictive performance according to Y, i.e., with maximum $y(\alpha(x))$

Smith-Miles, K. (2008) (SMITH-MILES, 2008) improved this abstract model by proposing generalizations that can also be applied to the algorithm design problem. In this proposal, some components are added: the set of MtL algorithms; the generation of empirical rules or algorithm rankings; the examination of the empirical results, which may guide theoretical support to refine the algorithms.

One crucial component of the previous models is the definition of the set of standard meta-features (F) used to describe the general properties of datasets. These meta-features must be able to provide evidence about the future performance of the algorithms in A (SOARES; PETRAK; BRAZDIL, 2001; REIF, 2012) and to discriminate, with a low computational cost, the performance of a group of algorithms. (RIVOLLI *et al.*, 2018) gathered the most used meta-features in the literature. We consider such meta-features in this paper. Next, we describe the essential categories of meta-features. For further information, please check (RIVOLLI *et al.*, 2018).

The main standard meta-features used in the MtL literature can be divided into:

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

- **Statistical:** meta-features that capture statistical properties of the data (REIF *et al.*, 2014), mainly of localization and distribution, such as 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; PINHO; MORENO, 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 (DT) models (BENSUSAN; GIRAUD-CARRIER; KENNEDY, 2000; PENG *et al.*, 2002), 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.

- **Others:** standalone, time-related, concept and case-based meta-features (VANSCHOREN *et al.*, 2012; MUÑOZ *et al.*, 2018), clustering and distance-based measures (VUKICEVIC *et al.*, 2016; PIMENTEL; CARVALHO, 2019), among others. These describe characteristics that do not fit in the other groups.

The definition of the set of problem instances (P) is another concern, when the ideal would be to use a large number of diverse datasets, in order to induce a reliable meta-model. To reduce the bias in this choice, datasets from several data repositories, like UCI[2] (DUA; GRAFF, 2017) and OpenML[3] (VANSCHOREN *et al.*, 2013), can be used.

The algorithm space A represents a set of candidate algorithms to be recommended in the algorithm selection process. Ideally, these algorithms should also be sufficiently different from each other and represent all regions in the algorithm space (MUÑOZ *et al.*, 2018). Different measures can evaluate the models induced by the algorithms. For classification tasks, most of the studies in the MtL use accuracy. However, other indices, like F_β, AUC, and kappa coefficient, can also be used. For regression problems, Mean Squared Error (MSE) or Root MSE (RMSE) (or normalized versions of such measures) are usually employed.

After the extraction of the standard meta-features from the datasets and the evaluation of the performance of a set of algorithms for these datasets, the next step is labeling each meta-example in the meta-base. Brazdil et al. (BRAZDIL *et al.*, 2009) summarize the three main properties frequently used to label the meta-examples in MtL: (*i*) the algorithm that presented the best performance on the dataset (a classification task); (*ii*) the ranking of the algorithms according to their performance on the dataset (a ranking classification task), where the algorithm with

[2] https://archive.ics.uci.edu/ml/index.php

[3] http://www.openml.org/

the best performance is top-ranked; and (*iii*) the performance value obtained by each evaluated algorithm on the dataset (a regression task).

6.3 Methods

In this section, we describe the experimental setup performed in this paper. First, we describe how the meta-dataset was built, second, we describe how we evaluated the MtL that estimates the simulated CMs, and third, we explain the computational cost experiment to compare the runtime execution between the groups of measures. Finally, we analyzed the simulated CM on real datasets when preprocessing techniques are used to balance them.

6.3.1 The meta-dataset

We used 161 binary datasets, in which 41 datasets have less than 25% of minority class instances, while the remaining 120 ones have more than 25% of minority class instances. We call these two sets of datasets, respectively, the high imbalanced and the low imbalanced datasets. Table 23 shows the number of examples, features, and percentage of minority class of all 161 datasets considered.

Table 23 – Characteristics of the datasets used to build the meta-dataset

Characteristic	Min value	Max value	Mean value
Number of Instances	34	5,278	509
Number of Features	3	95	16
Percentage of minority class	4%	49%	33%

Both sets combined are used to build the meta-dataset. For each dataset, we extracted the standard meta-features and the decomposed data CMs. The standard meta-feature set corresponds to the meta-features of the meta-base, while the set of CMs corresponds to the target features.

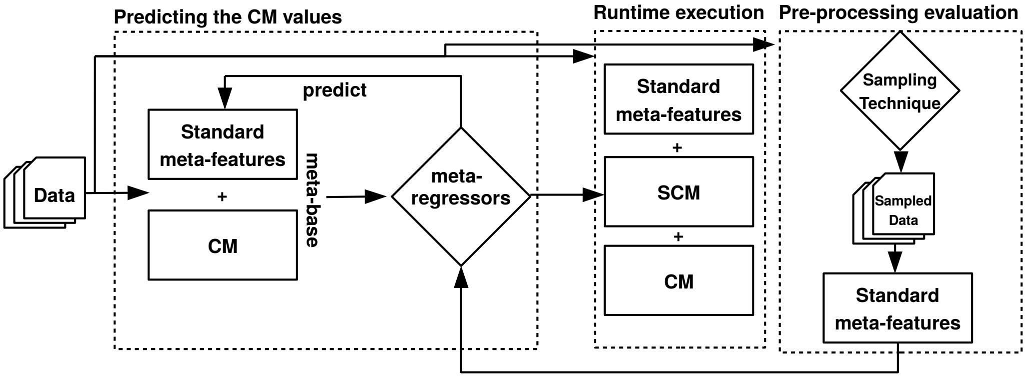
6.3.2 The meta-learning

We used regressor models to predict the value of each decomposed CM, induced by the Distance Weighted k-Nearest Neighbor (DWNN), Random Forest (RF) and Support Vector Regressor (SVR). As baselines, we used the Random (RD) and Mean (DF) approaches. The RD approach consists of selecting randomly one value for each CM using the training set. The DF approach consists of using the mean value of each CM on the training set. We performed a leave-one-out sampling to evaluate the strategies. We measured the error of the meta-regressor using Mean Squared Error (MSE). We also analyzed the trade-off between the computational cost of the standard meta-features, the original CMs, and the simulated CMs.

6.3.3 Preprocessing techniques analysis

In order to evaluate whether the simulated CMs can be helpful in practical analysis, we also performed an experiment using two traditional preprocessing techniques, Random Undersampling (RU) and Synthetic Minority Over-sampling Technique (SMOTE) (CHAWLA *et al.*, 2002). We randomly selected 19 datasets with less than 25% of the minority class. For each selected dataset, we applied the preprocessing techniques with different sample sizes, up to 100%, in which 0% represents that no instances were sampled and 100% represents a sampled dataset with a proportion of $1:1$ between the classes. Each selected dataset and its sampled datasets versions are not used in the training phase. For each sampled dataset, we extracted the standard meta-features, the CMs, and the simulated CMs, in which the latter has never seen this dataset nor its original one. In that way, we can track the evolution of both CMs, as the sample size increases. Figure 28 illustrates the experimental pipeline.

Figure 28 – Evaluation methodology used in the experiments.



Source: Barella, Garcia and Carvalho (2020).

The 161 datasets were selected from the OpenML repository (VANSCHOREN *et al.*, 2013). They represent diverse context datasets, with binary classes and no missing values. The standard meta-features were extracted using the `mfe`[4] package, whereas the CMs were extracted using the `ImbCoL`[5] package. The simulated CMs are available in a R package called `SImbCoL`[6].

6.4 Results and Discussion

In this paper, we show that a MtL approach is effective in simulating the CMs. For that, first, we evaluated a MtL approach to predict the CMs based on simple and fast meta-features. We show that our approach has a low error rate on estimating them, that it performs better than the

[4] https://github.com/rivolli/mfe
[5] https://github.com/victorhb/ImbCoL
[6] https://github.com/victorhb/SImbCoL

baselines. In order to prove the efficiency of that strategy, we also evaluate the time to simulate the CMs. The results indicate that they are faster than the original ones for all datasets. The last analysis shows that the simulated CMs are as helpful as the original ones when estimating the difficulty after applying preprocessing techniques.

Figure 29 shows the MSE for each regression approach for high and low imbalanced datasets. The x-axis shows the regressors, including the baselines in the shadowed area. The y-axis shows the MSE. The colors represent whether the simulation error is related to the positive (P) class, the minority class, or the negative (N), which is the majority class. On the right part of the figure, the name of the CMs in question are displayed.

The MSE analysis indicates that the meta-regressors outperformed the baselines with a better predictive performance for almost all cases. Even on F2 and T1, CMs that the MtL regressors had the highest MSEs, the regressors performed better than the baselines. Compared to the N class, the P class CMs tend to be more difficult to induce, especially on the high imbalanced datasets. Besides, the regressors showed lower MSE for the low imbalanced datasets, compared to those high imbalanced.

We performed a paired Friedman-Nemenyi statistical test with a confidence level of 95%. The test confirmed that both DWNN and RF regressors performed better than the baselines and SVR for almost all CMs. Also, the test showed that RF performed better than the DWNN on N2 and N3 CMs. For that reason, in the subsequent analysis, we only consider RF as meta-regressor.

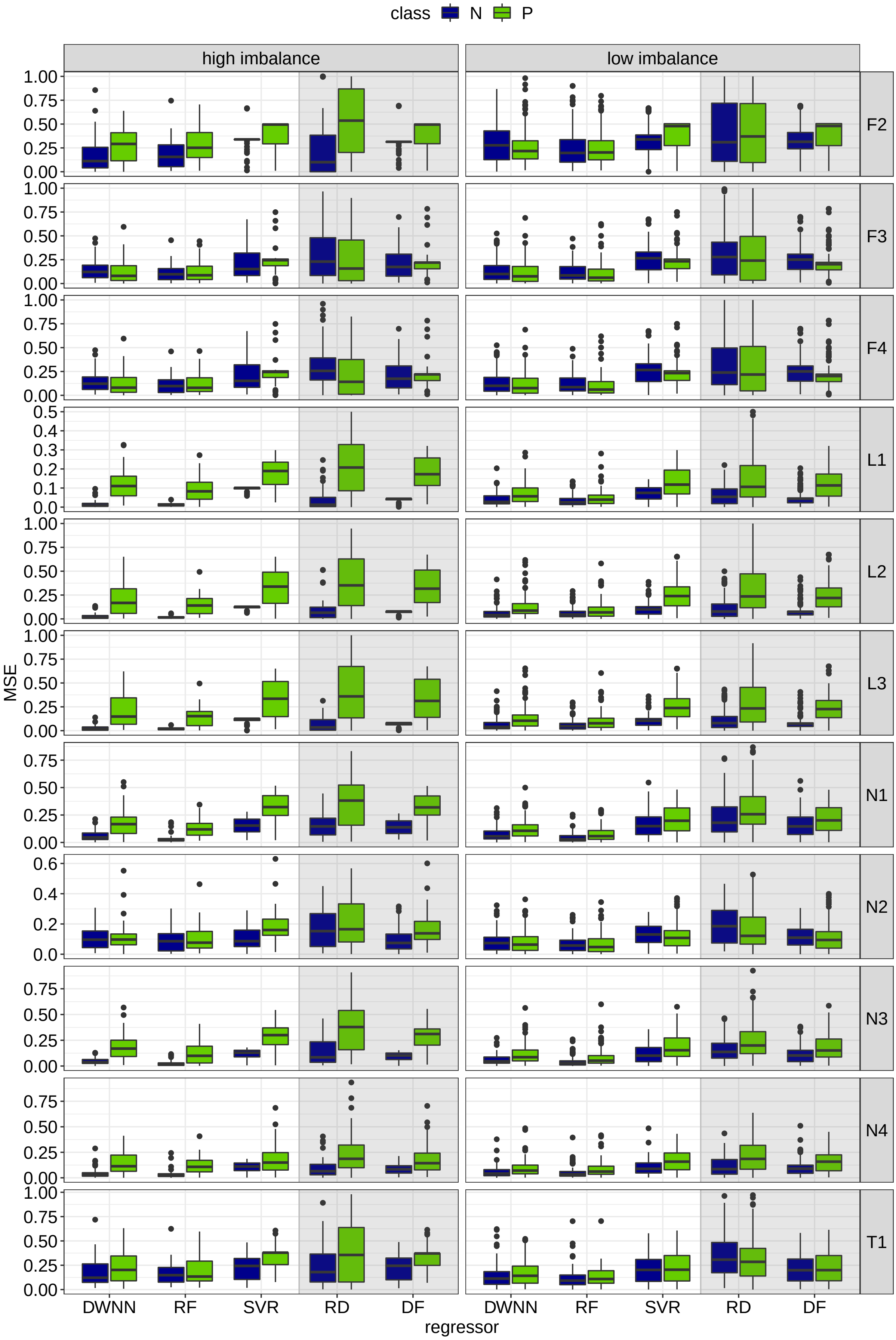
Figure 30 shows a heatmap of the Pearson's correlation between the original and simulated CMs using RF. Each column and row corresponds to the classes and the original CMs, respectively. Each box is colored according to the correlation, from white (lowest correlation) to gray (highest correlation). The correlation values are also shown inside the heatmap's cells.

Most correlations are higher than 70%, and all presented a p-value lower than 0.05. N1 is the CM with the highest correlation for both classes, corroborating with the results on MSE. Although the MSEs of the N class were lower than the P class, the mean values of correlations for the P class is 0.83, and 0.79 for the N class. The linearity measures are responsible for bringing down the mean correlations of the N class. Most of the original linearity CMs values for the N class is grouped close to zero, which made their MSE estimation small but affected negatively their correlation.

Figure 31 illustrates the feature importance of the RF meta-regressor through the increase of MSE considering the top 30 meta-features. The x-axis represents the meta-features sorted, and the y-axis shows the MSE generated by leaving out the meta-feature.

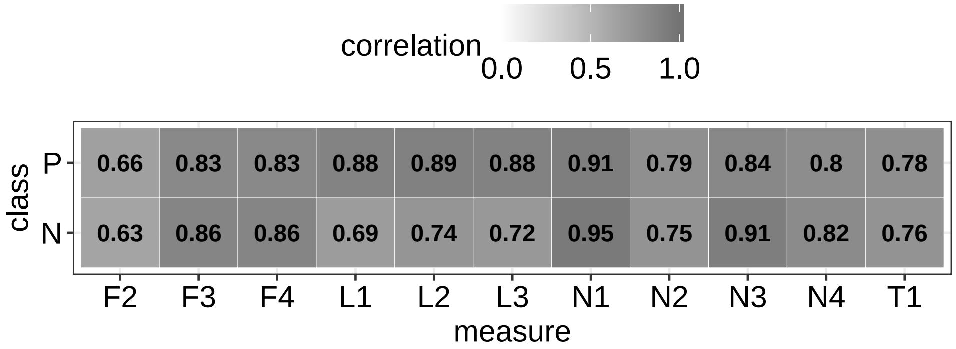
According to the results, the most important meta-features are based on statistical, landmarking, information-theoretic, and model-based. The statistical meta-features are the canonical correlation between the predictive attributes, and the class is present. From landmarking measures, they are related to the performance of simple meta-models induced by the k-NN, the

Figure 29 – MSE of the regressors, considering each CM for each class on different levels of imbalance.



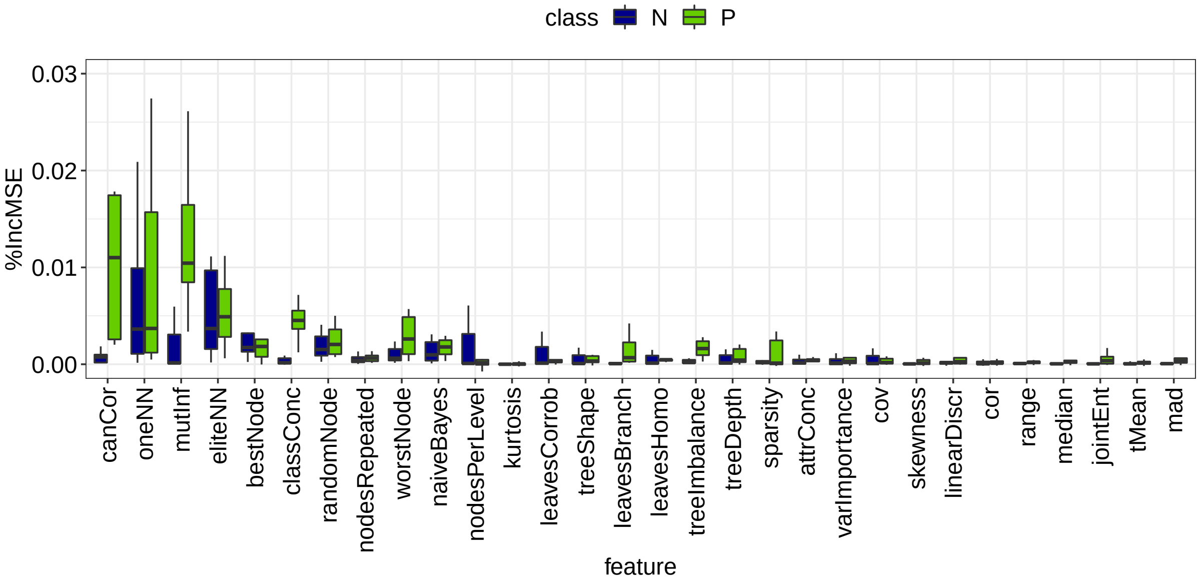
Source: Barella, Garcia and Carvalho (2020).

Figure 30 – The correlation between the original CM and the CM simulated by RF.



Source: Barella, Garcia and Carvalho (2020).

Figure 31 – The feature importance of the meta-dataset using RF.

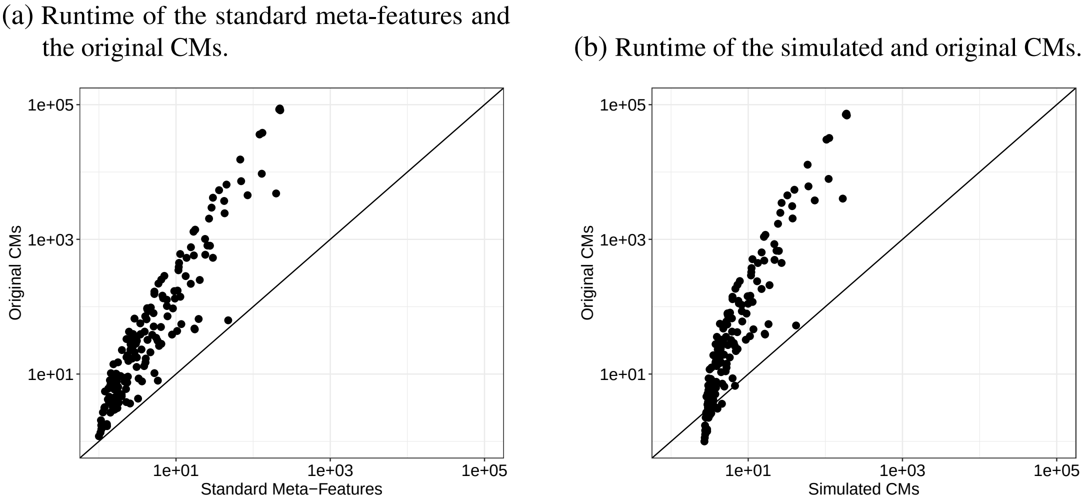


Source: Barella, Garcia and Carvalho (2020).

DT algorithm, and the Naive Bayes. The information-theoretic measures highlighted are the mutual information and the concentration coefficient for each pair of attributes. The model-based measures are related to the proportion of training instances to the DT model leaf, the number of nodes of the DT model per number of instances, and the number of nodes per attribute. We observe that there is a difference between the feature importance for the P class and the N class. The difficulty of the minority class is related to a group of meta-features that is, according to the results, less relevant to the majority class's difficulty.

Figure 32 compares the time to compute the standard meta-features, the original and simulated CMs. The time is presented on a log-scale to improve visualization. Each point represents a dataset, and those in the diagonal line indicate when the time is similar, the ones above the main diagonal means that y-axis spent more time to be computed than the strategy from the x-axis, while values below that line indicate the opposite.

Figure 32 – Time elapsed to extract the standard meta-features, original and simulated CMs for each dataset.



(a) Runtime of the standard meta-features and the original CMs.

(b) Runtime of the simulated and original CMs.

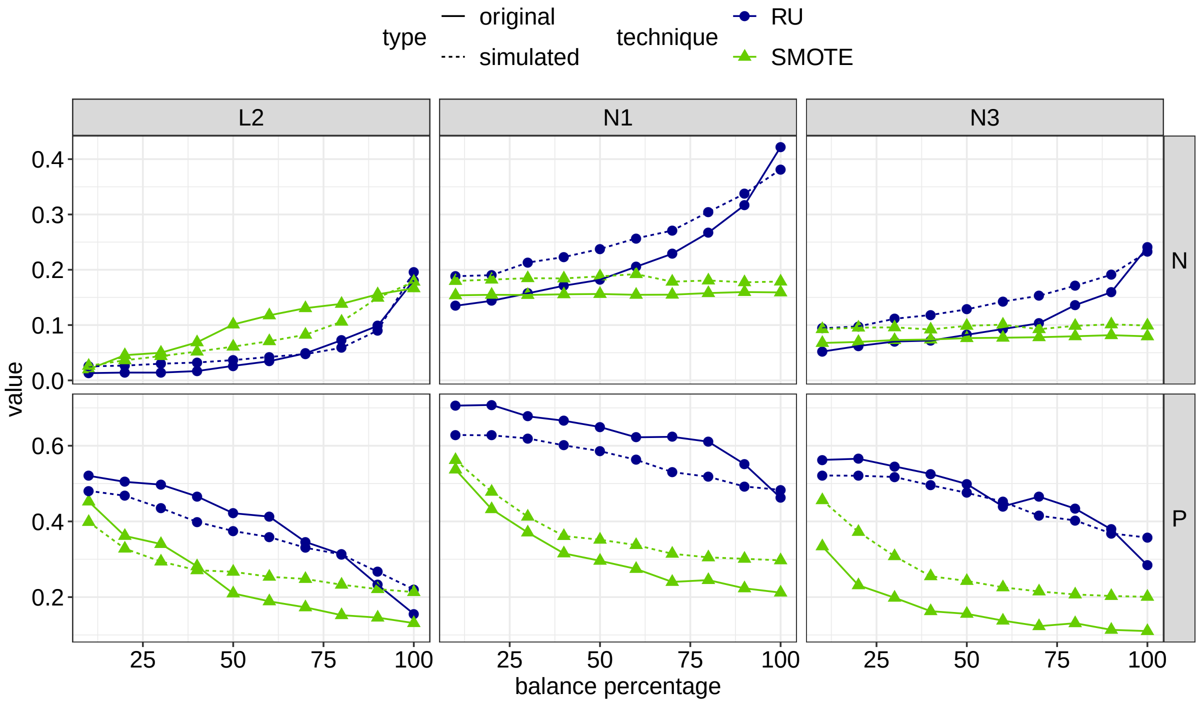
Source: Barella, Garcia and Carvalho (2020).

In Figure 32a all datasets are above the main diagonal, meaning that, for all datasets, calculating the original CMs took more time than extracting the standard meta-features. The extraction of the standard meta-features is the most time-consuming process of simulating the CMs after the models are built. In Figure 32b almost all datasets are above the main diagonal, meaning that calculate the original CMs took more time than extracting the simulated CMs. Thus, we show that a MtL approach using such meta-features is faster than calculating the CMs.

In Figure 33, we can see the mean values of the original and simulated CMs after applying SMOTE and RU with various sample sizes. The selected measures are L2, N1 and N3, the most imformative CMs (BARELLA; GARCIA; CARVALHO, 2019). The x-axis represents the sample size from 10% to 100%, e.q. how balanced the dataset is, and the y-axis represents the mean values of CMs. The figure shows the results for both classes, P and N, separately.

As the datasets get more balanced, the P class becomes less difficult, and the N class usually gets more difficult. While SMOTE decreases more the complexity than RU in the P class, RU tends to increase more the complexity of the N class. The main difference between original and simulates CMs occurs for N3 measures after applying SMOTE. In all other cases, the simulated CMs are similar to the original ones. Therefore, both original and simulated measures follow this pattern, giving evidence that the simulated CMs are as useful as the original CM to track data complexity when applying data balancing techniques.

Figure 33 – Mean values of original CMs and simulated CMs after applying SMOTE and RU with various sample sizes.



Source: Barella, Garcia and Carvalho (2020).

6.5 Conclusions

Measuring data complexity is useful for several ML applications, such as supporting the preprocessing techniques and estimating the expected difficulty of a classification problem. Although CMs are very important in these areas, they have a high computational cost that may prevent their popularization and efficient use. In this paper, we showed that a MtL approach is faster and yet effective to simulate them. For that, meta-models were induced based on standard meta-features, which have a lower computational cost. The main results indicate that the simulated CMs can predict the original CMs with low error and can be obtained at a lower computational cost. Moreover, the simulated CMs also tracks the data complexity when applying preprocessing techniques.

Future work shall look to increase the simulated CMs performance for the minority class, especially on the more imbalanced datasets. To improve the performance, we would like to investigate other meta-features, optimize the simulated CMs, evaluate other MtL approaches such as ranking, and investigate hyperparameter tuning for the classification algorithms. Additionally, we only considered binary datasets in this study. Multi-class datasets are more challenging and require further investigation.

6.6 Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES). The authors would like to thank CNPq, FAPESP (grant 2015/01382-0). The authors would like to thank CeMEAI-FAPESP (grant 2013/07375-0) for the computational resources.

CHAPTER 7

CONCLUSION

The nature of learning from imbalanced datasets is diverse and strongly related to other data intrinsic characteristics. For that reason, measuring class imbalance is not enough to explain and understand the problems encountered when applying ML to such datasets, while popular DCMs fail at this task. In this context, little attention has been paid to developing measures to assess imbalanced datasets characteristics.

Although several techniques for imbalanced datasets were proposed in the literature, no technique performs best in all classification tasks. This has recently lead the attention of researchers to build recommendation systems for such techniques. To the best of our knowledge, no work has approached recommending both pre-processing and algorithm-level techniques before.

This thesis investigated DCMs for imbalanced datasets. Based on empirical studies on artificial and real datasets, it can be concluded that the decomposed DCMs are useful in assessing the difficulty in imbalanced datasets. It was presented evidence of the descriptive ability of the decomposed DCMs, an example of an application using them as meta-features, and a way to reduce their computational cost.

An MtL system to recommend pre-processing and algorithm-level techniques was implemented. It used a set of well-known meta-features along with DCMs and decomposed DCMs. It performed better than a baseline considering predictive error. Using the system also reduces the computational cost compared to brutal force search in most cases. A faster estimation of the DCMs may decrease the system's computational cost, which was also addressed in this thesis.

7.1 Limitations and Future Work

Although the experiments were limited to small dimension datasets, it contained many datasets from different and diverse contexts. Still, high dimensional datasets may pose a problem in measuring the data complexity regarding computational cost. For example, the computational cost of measures based on NNs would increase intensely as the number of instances or features increases. Although models to estimate the decomposed DCMs were successfully implemented, further research is needed to determine if other solutions give more accurate results with lower computational cost, for example, techniques for estimating NNs (LIU; MOORE; GRAY, 2006).

This research clearly illustrates the decomposed DCMs applied to binary classification tasks, but it also raises questions about their use on multiclass ones. Some measures can not be directly applied to multiclass datasets, and it is not clear whether a decomposition one versus one or one versus all is the preferred approach. Still, the package we let available implements the latter.

The implemented recommendation system approached a diverse set of techniques for imbalanced datasets, but it did not recommend hyperparameters for any of the recommended elements. Classification algorithms and techniques for imbalanced datasets may need hyperparameter tuning to induce a model better. This problem is addressed as future work.

7.2 Main Contributions

This thesis successfully answered its research question: “How to define measures able to assess the complexity of imbalanced datasets and use these measures to recommend pre-processing and algorithm-level techniques with a good predictive performance for imbalanced datasets?”. During this research, several products were implemented to validate each specific hypothesis. All of these products were made available in three ready-to-use packages. In summary, the main contributions of this thesis are the following.

- A set of DCMs able to assess the difficulty of imbalanced datasets, even after applying pre-processing techniques
- A package in R implementing the decomposed DCMs [1]
- Evidence of the effectiveness of the decomposed DCMs as meta-features in an MtL system to recommend pre-processing and algorithm-level techniques
- A package in R implementing the MtL system to recommend techniques [2]
- A solution to the computational cost of the decomposed DCMs
- A package in R implementing the simulated decomposed DCMs [3]

[1] https://github.com/victorhb/ImbCoL
[2] https://github.com/victorhb/recommimb
[3] https://github.com/victorhb/SImbCoL

Moreover, the following research papers were written. Papers not indicated as *under review* or *yet to be submitted* are published.

- Barella, V. H., Garcia, L. P., de Souto, M. P., Lorena, A. C., & de Carvalho, A. (2018, July). Data complexity measures for imbalanced classification tasks. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
- Barella, V., Garcia, L., & de Carvalho, A. (2019, October). The Influence of Sampling on Imbalanced Data Classification. In 2019 8th Brazilian Conference on Intelligent Systems (BRACIS) (pp. 210-215). IEEE.
- Barella, V. H., Garcia, L. P., & de Carvalho, A. C. (2020, October). Simulating Complexity Measures on Imbalanced Datasets. In Brazilian Conference on Intelligent Systems (pp. 498-512). Springer, Cham.
- Barella, V. H., Garcia, L. P., de Souto, M. P., Lorena, A. C., & de Carvalho, A. Assessing the Data Complexity of Imbalanced Datasets. *Under review*
- Barella, V. H., Japkowicz, N., Garcia, L. P., & de Carvalho, A. Recommending Techniques for Imbalanced Datasets Using Meta-Learningand Data Complexity Measures. *Yet to be submitted*

Based on the results, data science practitioners should consider measuring the data complexity of imbalanced datasets, whether it is to interpret the data characteristics, select techniques, or develop new ones.

BIBLIOGRAPHY

ABDI, L.; HASHEMI, S. To combat multi-class imbalanced problems by means of over-sampling techniques. **IEEE Transactions on Knowledge & Data Engineering**, v. 28, n. 1, p. 238–251, 2016. Citations on pages 48, 67, 68, 71, 89, and 90.

ALCOBAçA, E.; SIQUEIRA, F.; RIVOLLI, A.; GARCIA, L. P. F.; OLIVA, J. T.; CARVALHO, A. C. P. L. F. d. MFE: Towards reproducible meta-feature extraction. **Journal of Machine Learning Research**, v. 21, n. 111, p. 1–5, 2020. Citation on page 99.

ANWAR, N.; JONES, G.; GANESH, S. Measurement of data complexity for classification problems with unbalanced data. **Statistical Analysis and Data Mining: The ASA Data Science Journal**, v. 7, n. 3, p. 194–211, 2014. Citations on pages 31, 62, and 63.

BARANDELA, R.; VALDOVINOS, R. M.; SÁNCHEZ, J. S. New applications of ensembles of classifiers. **Pattern Analysis & Applications**, Springer, v. 6, n. 3, p. 245–256, 2003. Citation on page 105.

BARELLA, V.; GARCIA, L.; CARVALHO, A. de. The influence of sampling on imbalanced data classification. In: IEEE. **2019 8th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2019. p. 210–215. Citations on pages 92, 94, 102, 118, and 129.

BARELLA, V. H.; GARCIA, L. P.; CARVALHO, A. C. de. Simulating complexity measures on imbalanced datasets. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2020. p. 498–512. Citations on pages 103, 114, 125, 127, 128, 129, and 130.

BARELLA, V. H.; GARCIA, L. P.; SOUTO, M. C. de; LORENA, A. C.; CARVALHO, A. C. de. Assessing the data complexity of imbalanced datasets. **Information Sciences**, Elsevier, 2020. Citations on pages 49, 51, 70, 72, 73, 75, 76, 77, 78, 80, 81, 99, 100, 101, and 102.

BARELLA, V. H.; GARCIA, L. P. F.; SOUTO, M. P. de; LORENA, A. C.; CARVALHO, A. de. Data complexity measures for imbalanced classification tasks. In: **International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2018. p. 1–8. Citations on pages 36, 39, 41, 43, 44, 48, 50, 72, 74, 86, 87, 90, 93, 99, 102, 112, 118, and 119.

BARUA, S.; ISLAM, M. M.; YAO, X.; MURASE, K. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 2, p. 405–425, 2014. Citations on pages 23, 48, 67, and 89.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. **ACM SIGKDD explorations newsletter**, v. 6, n. 1, p. 20–29, 2004. Citations on pages 22, 48, 86, and 98.

BENSUSAN, H.; GIRAUD-CARRIER, C.; KENNEDY, C. A higher-order approach to meta-learning. In: **10th International Conference Inductive Logic Programming (ILP)**. [S.l.: s.n.], 2000. p. 1–10. Citations on pages 102 and 123.

BORSOS, Z.; LEMNARU, C.; POTOLEA, R. Dealing with overlap and imbalance: a new metric and approach. **Pattern Analysis and Applications**, Springer, v. 21, n. 2, p. 381–395, 2018. Citations on pages 25 and 106.

BRAUSE, R.; LANGSDORF, T.; HEPP, M. Neural data mining for credit card fraud detection. In: **IEEE International Conference on Tools with Artificial Intelligence (ICTAI)**. [S.l.: s.n.], 1999. p. 103–106. Citation on page 30.

BRAZDIL, P.; GIRAUD-CARRIER, C.; SOARES, C.; VILALTA, R. **Metalearning - Applications to Data Mining**. 1. ed. [S.l.]: Springer, 2009. (Cognitive Technologies). Citations on pages 101, 118, and 123.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citation on page 105.

CANO, A.; ZAFRA, A.; VENTURA, S. Weighted data gravitation classification for standard and imbalanced data. **IEEE transactions on cybernetics**, v. 43, n. 6, p. 1672–1687, 2013. Citations on pages 65, 88, and 118.

CASTIELLO, C.; CASTELLANO, G.; FANELLI, A. M. Meta-data: Characterization of input features for meta-learning. In: **Modeling Decisions for Artificial Intelligence (MDAI)**. [S.l.: s.n.], 2005. v. 3558, p. 457–468. Citations on pages 101 and 122.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002. Citations on pages 23, 30, 48, 65, 66, 71, 86, 88, 89, 98, 103, 104, 118, and 125.

CHAWLA, N. V.; LAZAREVIC, A.; HALL, L. O.; BOWYER, K. W. Smoteboost: Improving prediction of the minority class in boosting. In: SPRINGER. **European conference on principles of data mining and knowledge discovery**. [S.l.], 2003. p. 107–119. Citation on page 105.

CHEN, Z.; YAN, Q.; HAN, H.; WANG, S.; PENG, L.; WANG, L.; YANG, B. Machine learning based mobile malware detection using highly imbalanced network traffic. **Information Sciences**, Elsevier, v. 433, p. 346–364, 2018. Citation on page 21.

CIESLAK, D. A.; HOENS, T. R.; CHAWLA, N. V.; KEGELMEYER, W. P. Hellinger distance decision trees are robust and skew-insensitive. **Data Mining and Knowledge Discovery**, v. 24, n. 1, p. 136–158, 2012. Citations on pages 65 and 88.

COSTA, A. J.; SANTOS, M. S.; SOARES, C.; ABREU, P. H. Analysis of imbalance strategies recommendation using a meta-learning approach. In: **7th ICML Workshop on Automated Machine Learning**. [S.l.: s.n.], 2020. Citations on pages 25, 98, and 106.

CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. [S.l.]: Cambridge university press, 2000. Citation on page 90.

DIAMANTINI, C.; POTENA, D. Bayes vector quantizer for class-imbalance problem. **IEEE Transactions on Knowledge and Data Engineering**, v. 21, n. 5, p. 638–651, 2009. Citations on pages 65 and 88.

DÍEZ-PASTOR, J. F.; RODRÍGUEZ, J. J.; GARCÍA-OSORIO, C. I.; KUNCHEVA, L. I. Diversity techniques improve the performance of the best imbalance learning ensembles. **Information Sciences**, v. 325, p. 98–117, 2015. Citations on pages 22, 31, 32, 48, 51, 68, and 88.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Http://archive.ics.uci.edu/ml. Citations on pages 100 and 123.

FERNANDES, E. R.; CARVALHO, A. C. de. Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning. **Information Sciences**, Elsevier, v. 494, p. 141–154, 2019. Citation on page 68.

FERNÁNDEZ, A.; GARCÍA, S.; GALAR, M.; PRATI, R. C.; KRAWCZYK, B.; HERRERA, F. **Learning from imbalanced data sets**. [S.l.]: Springer, 2018. Citations on pages 21, 22, 98, 103, 104, and 105.

FERNÁNDEZ, A.; JESUS, M. J. del; HERRERA, F. Addressing overlapping in classification with imbalanced datasets: A first multi-objective approach for feature and instance selection. In: **International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)**. [S.l.: s.n.], 2015. p. 36–44. Citations on pages 22, 31, 32, 48, 51, 68, and 88.

FERNáNDEZ, A.; GARCíA, S.; GALAR, M.; PRATI, R.; KRAWCZYK, B.; HERRERA, F. **Learning from Imbalanced Data Sets**. [S.l.]: Springer International Publishing, 2018. Citations on pages 48, 65, 86, 88, and 118.

FLACH, P. **Machine learning: the art and science of algorithms that make sense of data**. [S.l.]: Cambridge University Press, 2012. Citation on page 21.

FRENAY, B.; VERLEYSEN, M. Classification in the presence of label noise: a survey. **Neural Networks and Learning Systems, IEEE Transactions on**, v. 25, n. 5, p. 845 – 869, 2014. Citation on page 98.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997. Citation on page 105.

GARCIA, L. P.; RIVOLLI, A.; ALCOBAć, E.; LORENA, A. C.; CARVALHO, A. C. de. Boosting meta-learning with simulated data complexity measures. **Intelligent Data Analysis**, IOS Press, v. 24, n. 5, p. 1011–1028, 2020. Citations on pages 99 and 103.

GARCIA, L. P. F.; CARVALHO, A. C. P. L. F. de; LORENA, A. C. Effect of label noise in the complexity of classification problems. **Neurocomputing**, v. 160, p. 108–119, 2015. Citations on pages 31 and 48.

GARCIA, L. P. F.; LORENA, A. C. **ECoL: Complexity Measures for Classification Problems**. 2018. Https://CRAN.R-project.org/package=ECoL. Citations on pages 32, 50, 51, 102, and 119.

GARCIA, L. P. F.; LORENA, A. C.; SOUTO, M. P. de; HO, T. K. Classifier recommendation using data complexity measures. In: **24th International Conference on Pattern Recognition (ICPR)**. [S.l.: s.n.], 2018. v. 1, p. 874–879. Citation on page 48.

GONZALEZ-ABRIL, L.; NUÑEZ, H.; ANGULO, C.; VELASCO, F. GSVM: An SVM for handling imbalanced accuracy between classes inbi-classification problems. **Applied Soft Computing**, v. 17, p. 23–31, 2014. Citations on pages 65, 88, and 118.

HAIXIANG, G.; YIJING, L.; SHANG, J.; MINGYUN, G.; YUANYUE, H.; BING, G. Learning from class-imbalanced data: Review of methods and applications. **Expert Systems with Applications**, 2016. Citation on page 30.

HAN, H.; WANG, W.-Y.; MAO, B.-H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: **International Conference on Intelligent Computing (ICIC)**. [S.l.: s.n.], 2005. p. 878–887. Citations on pages 30, 65, 66, 71, 88, 89, 103, and 104.

HAYKIN, S. **Neural Networks – A Compreensive Foundation**. [S.l.]: Prentice-Hall, 1999. Citation on page 90.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: **International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2008. p. 1322–1328. Citations on pages 30, 65, 66, 71, 88, 89, 98, 103, 104, and 118.

HE, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on Knowledge & Data Engineering**, v. 21, n. 9, p. 1263–1284, 2008. Citations on pages 48, 65, 66, 86, 89, and 104.

HO, T. K.; BASU, M. Complexity measures of supervised classification problems. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 3, p. 289–300, 2002. Citations on pages 22, 30, 32, 48, 50, 60, 86, 87, 99, 100, 101, 102, 118, and 119.

HO, T. K.; BASU, M.; LAW, M. H. C. Measures of geometrical complexity in classification problems. In: **Data Complexity in Pattern Recognition**. [S.l.: s.n.], 2006. p. 1–23. Citations on pages 48, 50, 102, and 119.

JAPKOWICZ, N.; SHAH, M. **Evaluating learning algorithms: a classification perspective**. [S.l.]: Cambridge University Press, 2011. Citation on page 21.

JO, T.; JAPKOWICZ, N. Class imbalances versus small disjuncts. **ACM Sigkdd Explorations Newsletter**, ACM New York, NY, USA, v. 6, n. 1, p. 40–49, 2004. Citations on pages 22, 65, 66, 71, 88, 98, and 103.

KHOSHGOFTAAR, T.; HULSE, J. V.; NAPOLITANO, A. Comparing boosting and bagging techniques with noisy and imbalanced data. **Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on**, v. 41, n. 3, p. 552–568, May 2011. ISSN 1083-4427. Citation on page 22.

KOLACZYK, E. D. **Statistical Analysis of Network Data: Methods and Models**. [S.l.]: Springer Publishing Company, Incorporated, 2009. Citation on page 65.

KUBAT, M.; MATWIN, S. *et al.* Addressing the curse of imbalanced training sets: one-sided selection. In: CITESEER. **Icml**. [S.l.], 1997. v. 97, p. 179–186. Citations on pages 22, 30, 48, 65, 67, 71, 86, 88, and 103.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. **2008 Eighth IEEE International Conference on Data Mining**. [S.l.], 2008. p. 413–422. Citations on pages 103 and 105.

LIU, T.; MOORE, A. W.; GRAY, A. New algorithms for efficient high-dimensional nonparametric classification. **Journal of Machine Learning Research**, v. 7, n. Jun, p. 1135–1158, 2006. Citation on page 134.

LÓPEZ, V.; FERNÁNDEZ, A.; GARCÍA, S.; PALADE, V.; HERRERA, F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. **Information sciences**, Elsevier, v. 250, p. 113–141, 2013. Citation on page 98.

LORENA, A. C.; de Souto, M. C. P. On measuring the complexity of classification problems. In: **International Conference on Neural Information Processing**. [S.l.: s.n.], 2015. p. 158–167. Citations on pages 32, 48, 50, 102, and 119.

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 Computing Surveys (CSUR)**, v. 52, n. 5, 2019. Citations on pages 22, 48, 50, 51, 60, 99, 100, 101, 102, 118, and 119.

LU, Y.; CHEUNG, Y.-m.; TANG, Y. Y. Bayes imbalance impact index: A measure of class imbalanced data set for classification problem. **IEEE transactions on neural networks and learning systems**, IEEE, 2019. Citations on pages 62 and 64.

LUENGO, J.; FERNÁNDEZ, A.; GARCÍA, S.; HERRERA, F. Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling. **Soft Computing**, v. 15, n. 10, p. 1909–1936, 2011. Citations on pages 31, 32, 48, 51, 68, and 88.

LUENGO, J.; HERRERA, F. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. **Knowledge and Information Systems**, v. 42, n. 1, p. 147–180, 2015. Citations on pages 22, 31, 48, and 87.

MACIÀ, N.; BERNADÓ-MANSILLA, E. Towards UCI+: A mindful repository design. **Information Sciences**, v. 261, p. 237–262, 2014. Citations on pages 31 and 48.

MALDONADO, S.; WEBER, R.; FAMILI, F. Feature selection for high-dimensional class-imbalanced data sets using support vector machines. **Information Sciences**, v. 286, p. 228 – 246, 2014. ISSN 0020-0255. Available: <http://www.sciencedirect.com/science/article/pii/S0020025514007154>. Citation on page 22.

MANTOVANI, R. G.; ROSSI, A. L.; VANSCHOREN, J.; BISCHL, B.; CARVALHO, A. C. To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. In: **International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2015. p. 1–8. Citation on page 79.

MAZUROWSKI, M. A.; HABAS, P. A.; ZURADA, J. M.; LO, J. Y.; BAKER, J. A.; TOURASSI, G. D. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. **Neural networks**, Elsevier, v. 21, n. 2-3, p. 427–436, 2008. Citation on page 22.

MEYER, D.; DIMITRIADOU, E.; HORNIK, K.; WEINGESSEL, A.; LEISCH, F.; CHANG, C.-C.; LIN, C.-C.; MEYER, M. D. **Package 'e1071'**. 2017. Citation on page 38.

MITCHELL, T. M. **Machine Learning**. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077. Citations on pages 21 and 90.

MORAIS, R. F. de; MIRANDA, P. B.; SILVA, R. M. A meta-learning method to select under-sampling algorithms for imbalanced data sets. In: IEEE. **2016 5th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2016. p. 385–390. Citations on pages 25, 98, and 105.

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 98, 100, 101, 102, 118, and 123.

NG, W. W.; HU, J.; YEUNG, D. S.; YIN, S.; ROLI, F. Diversified sensitivity-based undersampling for imbalance classification problems. **IEEE transactions on cybernetics**, v. 45, n. 11, p. 2402–2412, 2015. Citations on pages 23, 67, and 89.

ORRIOLS-PUIG, A.; MACIá, N.; HO, T. K. **Documentation for the Data Complexity Library in C++**. [S.l.], 2010. Citations on pages 48, 50, 60, 102, and 119.

PENG, Y.; FLACH, P. A.; SOARES, C.; BRAZDIL, P. Improved dataset characterisation for meta-learning. In: **5th International Conference on Discovery Science (DS)**. [S.l.: s.n.], 2002. v. 2534, p. 141–152. Citations on pages 102 and 123.

PIMENTEL, B. A.; CARVALHO, A. C. P. L. F. de. A new data characterization for selecting clustering algorithms using meta-learning. **Information Sciences**, v. 477, p. 203–219, 2019. Citations on pages 102 and 123.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. Class imbalances versus class overlapping: an analysis of a learning system behavior. In: **Mexican International Conference on Artificial Intelligence (MICAI)**. [S.l.: s.n.], 2004. v. 4, p. 312–321. Citation on page 30.

QUINLAN, J. R. Induction of decision trees. **Machine Learning**, v. 1, n. 1, p. 81–106, 1986. Citation on page 90.

REIF, M. A comprehensive dataset for evaluating approaches of various meta-learning tasks. In: **1st International Conference on Pattern Recognition Applications and Methods**. [S.l.: s.n.], 2012. p. 273–276. Citations on pages 100, 101, and 122.

REIF, M.; SHAFAIT, F.; GOLDSTEIN, M.; BREUEL, T.; DENGEL, A. Automatic classifier selection for non-experts. **Pattern Analysis and Applications**, v. 17, n. 1, p. 83–96, 2014. Citations on pages 101, 102, 122, and 123.

RICE, J. R. The algorithm selection problem. **Advances in Computers**, v. 15, p. 65–118, 1976. Citations on pages 98, 99, and 122.

RIVOLLI, A.; GARCIA, L. P.; SOARES, C.; VANSCHOREN, J.; CARVALHO, A. C. de. Towards reproducible empirical research in meta-learning. **arXiv preprint arXiv:1808.10406**, p. 32–52, 2018. Citations on pages 99, 100, 101, and 122.

SÁEZ, J. A.; LUENGO, J.; STEFANOWSKI, J.; HERRERA, F. SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. **Information Sciences**, Elsevier, v. 291, p. 184–203, 2015. Citation on page 71.

SCHÖLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A. J.; WILLIAMSON, R. C. Estimating the support of a high-dimensional distribution. **Neural computation**, MIT Press, v. 13, n. 7, p. 1443–1471, 2001. Citation on page 105.

SEGRERA, S.; PINHO, J.; MORENO, M. N. Information-theoretic measures for meta-learning. In: **3rd Hybrid Artificial Intelligence Systems (HAIS)**. [S.l.: s.n.], 2008. p. 458–465. Citations on pages 101 and 123.

SEIFFERT, C.; KHOSHGOFTAAR, T. M.; HULSE, J. V.; NAPOLITANO, A. Rusboost: A hybrid approach to alleviating class imbalance. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 40, n. 1, p. 185–197, 2009. Citations on pages 24 and 105.

SINGH, D.; GOSAIN, A.; SAHA, A. Weighted k-nearest neighbor based data complexity metrics for imbalanced datasets. **Statistical Analysis and Data Mining: The ASA Data Science Journal**, Wiley Online Library, 2020. Citations on pages 62, 63, 64, and 65.

SMITH, F. W. Pattern classifier design by linear programming. **IEEE transactions on computers**, v. 100, n. 4, p. 367–372, 1968. Citation on page 60.

SMITH, M. R.; MARTINEZ, T.; GIRAUD-CARRIER, C. An instance level analysis of data complexity. **Machine learning**, v. 95, n. 2, p. 225–256, 2014. Citation on page 65.

SMITH-MILES, K. A. Cross-disciplinary perspectives on meta-learning for algorithm selection. **ACM Computing Surveys**, v. 41, n. 1, p. 1–25, 2008. Citations on pages 98, 100, 102, 118, 122, and 123.

SMOLYAKOV, D.; KOROTIN, A.; EROFEEV, P.; PAPANOV, A.; BURNAEV, E. Meta-learning for resampling recommendation systems. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Eleventh International Conference on Machine Vision (ICMV 2018)**. [S.l.], 2019. v. 11041, p. 110411S. Citations on pages 25, 98, and 105.

SOARES, C.; PETRAK, J.; BRAZDIL, P. Sampling-based relative landmarks: Systematically test-driving algorithms before choosing. In: **10th Portuguese Conference on Artificial Intelligence (EPIA)**. [S.l.: s.n.], 2001. p. 88–95. Citations on pages 100 and 122.

SUN, Y.; KAMEL, M. S.; WONG, A. K.; WANG, Y. Cost-sensitive boosting for classification of imbalanced data. **Pattern Recognition**, v. 40, n. 12, p. 3358–3378, 2007. Citation on page 30.

SáEZ, J.; LUENGO, J.; STEFANOWSKI, J.; HERRERA, F. Managing borderline and noisy examples in imbalanced classification by combining smote with ensemble filtering. In: CORCHADO, E.; LOZANO, J.; QUINTIáN, H.; YIN, H. (Ed.). **Intelligent Data Engineering and Automated Learning – IDEAL 2014**. Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8669). p. 61–68. ISBN 978-3-319-10839-1. Available: <http://dx.doi.org/10.1007/978-3-319-10840-7_8>. Citation on page 22.

TAVALLAEE, M.; STAKHANOVA, N.; GHORBANI, A. A. Toward credible evaluation of anomaly-based intrusion-detection methods. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 40, n. 5, p. 516–524, 2010. Citation on page 21.

TOMEK, I. Two modifications of CNN. **IEEE Trans. Systems, Man and Cybernetics**, v. 6, p. 769–772, 1976. Citation on page 67.

VANSCHOREN, J. Meta-Learning: A Survey. **arXiv:1810.03548 [cs, stat]**, Oct. 2018. ArXiv: 1810.03548. Citation on page 100.

VANSCHOREN, J.; BLOCKEEL, H.; PFAHRINGER, B.; HOLMES, G. Experiment databases. **Machine Learning**, v. 87, n. 2, p. 127–158, 2012. Citations on pages 102 and 123.

VANSCHOREN, J.; van Rijn, J. N.; BISCHL, B.; TORGO, L. OpenML: networked science in machine learning. **SIGKDD Explorations**, v. 15, n. 2, p. 49–60, 2013. Citations on pages 68, 90, 100, 123, and 125.

VEROPOULOS, K.; CAMPBELL, C.; CRISTIANINI, N. *et al.* Controlling the sensitivity of support vector machines. In: **Proceedings of the international joint conference on AI**. [S.l.: s.n.], 1999. v. 55, p. 60. Citations on pages 24, 30, 98, 103, and 105.

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. **International Journal of Data Mining and Bioinformatics (IJDMB)**, v. 14, n. 2, p. 101–119, 2016. Citations on pages 102 and 123.

WANG, S.; YAO, X. Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE. **2009 IEEE Symposium on Computational Intelligence and Data Mining**. [S.l.], 2009. p. 324–331. Citations on pages 98, 103, and 105.

YANG, Z.; TANG, W.; SHINTEMIROV, A.; WU, Q. Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 39, n. 6, p. 597–610, 2009. Citations on pages 21 and 22.

YU, H.; NI, J.; ZHAO, J. ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. **Neurocomputing**, v. 101, p. 309–318, 2013. Citations on pages 23, 67, and 89.

APPENDIX A

ASSESSING THE DATA COMPLEXITY OF IMBALANCED DATASETS

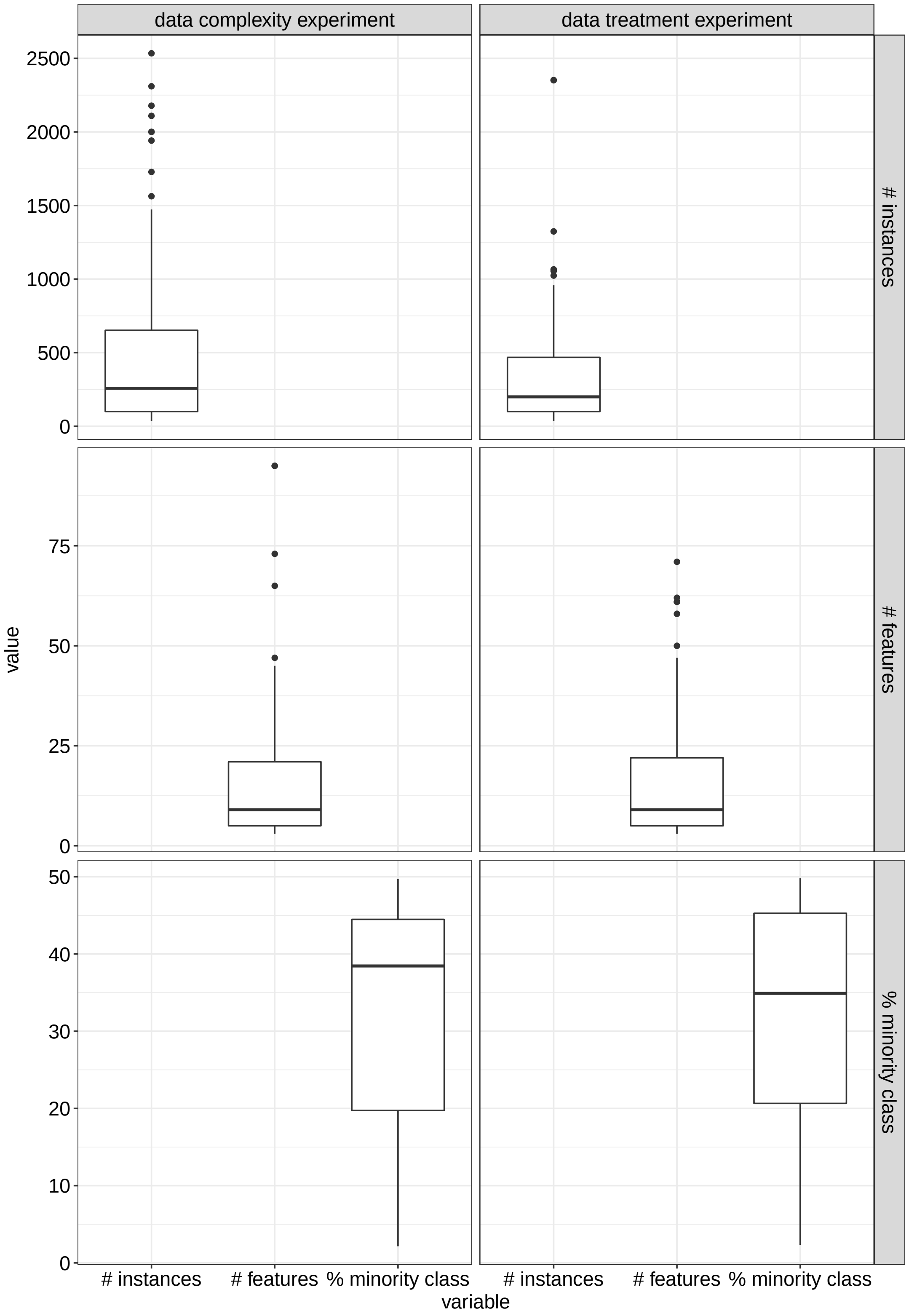
Table 24 – Information about the 102 datasets used in the experiment to evaluate the data complexity measures

OpenML ID	Number of Instances	Number of Features	% Minority Class	OpenML ID	Number of Instances	Number of Features	% Minority Class
31	1000	21	30	**927**	42	17	40.48
43	306	4	26.47	**928**	46	5	45.65
444	132	4	46.21	**931**	662	4	47.43
463	180	33	13.89	**934**	1156	6	22.15
467	52	10	48.08	**938**	42	11	45.24
472	87	4	40.23	**945**	76	7	47.37
714	125	5	39.2	**949**	559	5	14.31
717	508	11	43.7	**950**	559	5	3.4
724	468	4	44.44	**958**	2310	20	14.29
729	44	4	38.64	**962**	2000	7	10
733	209	7	26.79	**964**	36	23	33.33
736	111	4	47.75	**983**	1473	10	42.7
747	167	5	22.75	**987**	500	24	16
748	163	6	28.83	**988**	67	16	38.81
753	194	33	46.39	**991**	1728	7	29.98
758	67	16	26.87	**994**	846	19	25.77
764	450	4	12.22	**1009**	63	32	39.68
767	475	4	12.84	**1014**	797	5	19.45
770	625	7	49.6	**1016**	990	14	9.09
772	2178	4	44.49	**1020**	2000	65	10
777	47	8	42.55	**1025**	400	6	22.5
778	252	15	49.21	**1026**	155	9	31.61
780	51	7	41.18	**1045**	145	95	5.52
782	120	3	47.5	**1050**	1563	38	10.24
785	45	47	48.89	**1055**	89	9	22.47
787	50	6	48	**1061**	107	30	18.69
790	55	3	43.64	**1064**	101	30	14.85
791	43	3	39.53	**1066**	145	95	41.38
800	74	28	41.89	**1067**	2109	22	15.46
801	185	4	47.03	**1073**	274	9	48.91
811	264	3	38.26	**1075**	130	9	8.46
818	310	9	46.77	**1443**	661	38	7.87
825	506	21	44.07	**1444**	1043	38	12.18
826	576	12	41.49	**1446**	296	38	12.84
827	662	4	49.7	**1450**	125	40	35.2
841	950	10	48.63	**1451**	705	38	8.65
848	38	6	26.32	**1452**	745	37	2.15
859	74	10	41.89	**1462**	1372	5	44.46
860	380	3	48.68	**1464**	748	5	23.8
875	100	4	19	**1473**	100	10	12
882	60	16	48.33	**1487**	2534	73	6.31
885	131	4	36.64	**1488**	195	23	24.62
890	108	8	29.63	**1495**	250	7	42.8
891	93	7	38.71	**1504**	1941	34	34.67
893	73	6	45.21	**1506**	470	17	14.89
895	222	3	39.64	**1600**	267	45	20.6
900	400	7	41.25	**23499**	277	10	29.24
907	400	8	48.5	**40669**	160	7	43.75
915	315	14	42.22	**40705**	959	45	36.08
921	132	4	34.85	**40710**	303	14	45.54
925	323	5	45.82	**40981**	690	15	44.49

Table 25 – Information about the 102 datasets used in the experiment to evaluate the data complexity measures

OpenML ID	**Number of Instances**	**Number of Features**	**% Minority Class**	**OpenML ID**	**Number of Instances**	**Number of Features**	**% Minority Class**
37	768	9	34.9	**946**	88	3	48.86
40	208	61	46.63	**947**	559	5	4.29
50	958	10	34.66	**951**	559	5	2.33
53	270	14	44.44	**955**	151	6	34.44
59	351	35	35.9	**965**	101	18	40.59
311	937	50	4.38	**969**	150	5	33.33
336	267	23	20.6	**970**	841	71	37.69
448	120	4	35	**973**	178	14	39.89
450	264	5	7.2	**974**	132	5	38.64
459	83	4	44.58	**996**	214	10	35.51
461	100	7	27	**997**	625	5	46.08
465	97	11	24.74	**1004**	600	62	16.67
479	92	11	20.65	**1006**	148	19	45.27
713	52	4	46.15	**1011**	336	8	42.56
719	137	8	31.39	**1012**	194	30	35.57
721	200	11	48.5	**1013**	138	3	6.52
731	96	5	48.96	**1015**	72	4	16.67
741	1024	3	49.71	**1048**	369	9	44.72
745	159	16	33.96	**1054**	161	40	32.3
750	500	8	49.2	**1059**	121	30	7.44
765	475	4	13.47	**1060**	63	30	12.7
771	108	5	44.44	**1062**	36	30	22.22
774	662	4	47.89	**1063**	522	22	20.5
788	186	61	41.4	**1065**	458	40	9.39
795	662	4	49.4	**1071**	403	38	7.69
796	209	8	25.36	**1121**	294	12	28.57
798	106	58	22.64	**1167**	320	9	33.44
804	70	8	48.57	**1412**	226	24	15.49
814	468	3	45.3	**1441**	123	40	13.01
815	52	10	46.15	**1442**	253	38	10.67
817	48	5	47.92	**1447**	327	38	12.84
820	235	13	39.57	**1448**	194	40	18.56
835	48	5	43.75	**1449**	253	38	10.67
836	34	9	44.12	**1463**	100	6	32
853	506	14	41.3	**1467**	540	21	8.52
857	40	8	35	**1480**	583	11	28.64
862	87	11	48.28	**1490**	182	13	28.57
864	60	8	45	**1494**	1055	42	33.74
865	100	4	7	**1498**	462	10	34.63
867	130	3	19.23	**1510**	569	31	37.26
874	50	6	42	**1511**	440	9	32.27
886	500	8	49.8	**1524**	310	7	32.26
887	61	3	47.54	**1556**	120	7	49.17
892	50	8	48	**4329**	470	17	14.89
902	147	7	46.94	**40660**	42	12	30.95
905	39	4	30.77	**40680**	1324	11	22.05
906	400	8	48.25	**40683**	88	9	27.27
908	400	8	48	**40702**	1066	11	17.07
909	400	8	49.25	**40999**	2351	47	44.02
941	189	10	47.62	**41007**	2352	47	40.35
942	50	5	48				

Figure 34 – Comparison of the characteristics of both groups of datasets



Source: Elaborated by the author.

ICMC USP
SÃO CARLOS