Intelligent Systems Group

informatika fakultatea

facultad de informática

eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Konputazio Zientziak eta Adimen Artifizialaren Saila

Departamento de Ciencias de la Computación e Inteligencia Artificial

Theoretical and Methodological Advances in Semi-supervised Learning and the Class-Imbalance Problem

by

Jonathan Ortigosa-Hernández

Supervised by Iñaki Inza and Jose A. Lozano

Dissertation submitted to the Department of Computer Science and Artificial Intelligence of the University of the Basque Country (UPV/EHU) as partial fulfilment of the requirements for the PhD degree in Computer Science

Donostia - San Sebastián, Tuesday 21st August, 2018

To my father for instructing me the faculty of wonder, and

To my mother for teaching me that persistence and perfectionism are two qualities necessary for a great achievement.

This dissertation would not have been possible without your support. Thanks!

A mi padre por mostrarme que el cuestionamiento es un hábito muy constructivo, y

a mi madre por enseñarme que la persistencia y el perfeccionismo son dos medios necesarios para conseguir un resultado de calidad.

Esta tesis os la debo a vosotros. ¡Gracias!

Statistical thinking will one day be as necessary for efficient citizenship as the ability to read and write!

- H.G. Wells, *Mankind in the Making*

Acknowledgments

This research project was carried out during the years 2009–2018 at the Intelligent Systems Group, a research group belonging to the university of the Basque Country. Since many people helped me during this endeavour, I wish to drop a few lines to thank them for their support.

First of all, I want to express my deep gratitude to my mentors, Iñaki Inza and José A. Lozano, for their dedication, keen interest, timely advice, meticulous scrutiny, scholarly instruction and scientific perspective. They are mainly responsible for the kind of professional working person I currently am.

I would also like to thank to my colleagues of the Intelligent System Group: Alexander Mendiburu, Aritz Pérez, Borja Calvo, Carlos Echegoyen, Carlos Pérez, Dinora Morales, Ekhiñe Irurozki, Eneko Mateo, Guzmán Santafé, Izaskun Ibarbia, Javier Navaridas, Jerónimo Hernández, José Antonio Fernandes, José Antonio Pascual, José Luis Flores, Josu Ceberio, Juan Diego Rodríguez, Leticia Hernando, Ramón Sagarna, Roberto Santana, Rubén Armañanzas, Tania Lorido, Unai López, and Usue Mori. To all of them, sincere thanks for the incomparable working atmosphere, occasional advice, continuous support and the after-hours entertainment. In particular, I am especially grateful to Aritz Pérez, Ekhiñe Irurozki, Juan Diego Rodríguez and Leticia Hernando for their valuable research advice.

In like manner, I owe a debt of gratitude to Zhi-Hua Zhou for allowing me to complete a stay as a visitor in his research group, LAMDA, at Nanjing University and to his colleagues, who gave me the opportunity to enjoy such an enlightening experience.

This dissertation would not have been possible without financial support. It has be partially supported by the university of the Basque Country, the Basque Government, the Spanish Ministry of Science and Innovation, and the European Commission.

I would also like to acknowledge the Management and Human Resources departments, and colleagues of the Maluuba and Gestamp Automoción companies for facilitating the reconciliation of this PhD project with my work duties while combining both academia and businesses. Special thanks to Tania Herrador, Jon Armendariz, Miguel Muñoz, and Ainoa Cerezuelo.

Thanks also to the specialised designer Iker Teruelo for illustrating this dissertation and to the English teacher John Kennedy for proofreading all my work.

Above all, I would like to give thanks to the people who did not scientifically contribute to the dissertation but were my companions in this prolonged journey. If my mentors are responsible for shaping my intellectual virtues and forming adequate work habits, my family is accountable for setting up the pillars on which my working persona is based. I would like to give special thanks to my parents for unconditionally loving me, caring for me, educating me in the moral values which guide my daily decisions, trusting me when I would lose hope, staying awake the nights before important dates, being with me when I fell sick, crying with me in the bad times, etc. I cannot thank them enough for everything they have done for me. Without them, I would be nowhere. *Por eso, mamá y papá, esta tesis os la quiero dedicar a vosotros.*

Last but not least, I also want to dedicate this work to my friends and beloved ones. I only list the closest and uncited above ones but if you are reading this, you also have also the right to be included in the list. Ainara Ramos, Álex Fernández, Borja Rivas, Borja Mariscal, Dánae Zarzuelo, Iris Moreno, Jokin Ayerregaray, José Cereijo, Laura Carboneres, Leire Burgos, Leyre González, Luca Gómara, María Montero, Miriam Alzúa, Sara Gutiérrez, Sergio Fernández, and Unai Murillo, thanks for putting up with me all these years. This work is also yours.

Contents

0 Preface 1

0.1 Overview of the Dissertation 3

Part I Introduction to the Main Contributions on Classification

1 A Brief Introduction to Contemporary Classification 7

1.1 Main Types of Classification Problems 8

1.2 Learning of Classifiers 10

1.3 Evaluation of Classifiers 26

1.4 Threats when Obtaining Acceptable Classifiers 30

Main Contributions to Classification in a Nutshell 37

2.1 Semi-supervised Learning of Multi-dimensional Classifiers 37

2.2 A Study on Semi-supervised Learning of Multi-class Classifiers 42

2.3 A Study on the Competitiveness of Classifiers for Unbalanced Problems 45

2.4 A Measure for the Class-Imbalance Extent of Multi-class Problems 49

General Conclusions, Future Work, and Published Material . . . 53

3.1 Conclusions 53

3.2 Future Work 56

3.3 List of Publications 58

Part II Selected Refereed Publications

Approaching Sentiment Analysis by Using Semi-supervised Learning of Multi-dimensional Classifiers 63

Semi-supervised Multi-class Classification Problems with Scarcity of Labelled Data: A Theoretical Study 97

Towards Competitive Classifiers for Unbalanced Classification Problems: A Study on the Performance Scores 139

Measuring the Class-imbalance Extent of Multi-class Problems 181

References 191

0

Preface

After more than 70 years of life, *machine learning* [1] is at the edge of becoming a classical field of study. Evolved from the fields of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data [2]. As early as 1959, computer gaming pioneer Arthur Samuel defined it as “the field of study that gives computers the ability to learn without being explicitly programmed” [3].

In those early days, symbolic approaches [4], which were broadly used in artificial intelligence, dominated the field. Not until the 1990s did machine learning become reorganised as a separate field with its own personality, and it started to flourish benefiting from the increasing availability of digitised information, and the possibility to distribute that via the Internet. Those novel approaches shifted focus away from the rules induced by the symbolic approaches towards methods and models borrowed from statistics and probability theory [5], which are still alive and kicking.

Nowadays, probably due to the fact that fashionable terms for naming machine learning, such as big data [3] or data science [6], can be found every other week in the public media, the field is becoming increasingly ubiquitous with numerous applications in companies, transforming the way they carry out their businesses. For instance, machine learning along with several other IT technologies, under the catchy pseudonym of industrie 4.0 [7], is currently revolutionising the manufacturing industry.

Thus, in this historical context of expansion, social acclamation and ubiquity, where, as it is said, being a data scientist (a practitioner of machine learning) is the sexiest job of the century [8], this dissertation has the aim of theoretically and methodologically make a humble contribution to the much-publicised machine learning field, which is eager for novel solutions and techniques.

Specifically, my contributions are focussed on the problem of classification. Classification is defined as a statistical method used to build predicative models capable of separating and classifying new data objects [9]. Nowadays, this methodology can be found in many appliances. For example, in email servers, which are capable of filtering spam emails from the normal emails. However, how can an email server learn to differentiate between junk and real emails without being explicitly programmed to?

Suppose we have already been provided with a collection of emails, say l of them, some of them junk and some of them real, and from these emails we want to build ways to filter out spam. We refer to this collection as the training set, which is, in general, a labelled set of data whose sole purpose is to build a classifier. Now, from our training data, we need to determine which variables, called features, we want to measure in order to assess whether future emails are spam or not. This process is called feature engineering and its resulting features can be ordinal or nominal random variables: an example of a discrete feature in our spam classifier is whether the sender of an email comes from a `.eus` address or not, and an example of a continuous feature may be the frequency of a certain word, like “free”, “visa”, or “winner”, in a particular email. If n features are found, then our training dataset can be viewed as an $l \times n$ matrix where each email is represented by a row vector of features. Then, a spam classifier, which is a function mapping a vector of features into a vector of class variables (here, of size 1), can be directly learnt by just providing this matrix along with the already annotated class of each email to a classical supervised learning algorithm [9]. However, note that every classifier comes with an associated error: as you have probably noticed, sometimes, spam detectors filter out some normal emails or even let some junk mail enter your mailbox. Thus, since we are interested in minimising the associated error of the resulting classifier; selecting a representative and sufficient training dataset with enough examples of each class, obtaining informative features, and choosing an adequate supervised learning algorithm with proper parameters are usually key concepts in solving a supervised classification problem.

The above exposed supervised learning procedure has been widely applied and accepted in the machine learning community since it provides positive results. Unfortunately, it has its limitations. In this dissertation, I study two of these limitations and contribute to alleviate their negative impact in the performance of the resulting classifiers:

First, *having a sufficient labelled training set of training data is not always possible*. For example, suppose our required collection of emails is relatively large and individually annotating all of them is intractable. In that case, a set of techniques named **semi-supervised learning** [10] appears as an exciting direction in the machine learning community. These techniques aim at learning competitive classifiers from those learnt in supervised learning. However, instead of learning from a huge labelled dataset, they use training datasets composed of a small number of labelled examples and a huge number of examples whose class variables are unknown.

Secondly, *common supervised learning algorithms do not get along with atypical objects*. In other words, when a training dataset has very few annotated examples of some particular classes, the learning algorithm usually becomes biased towards the most probable classes ignoring those least probable classes under the assumption that it is dealing with either outliers or noisy data. Returning to the spam filter example, imagine we live in a utopian world where junk mail is rare. Then, our collection of size l of emails would have only a tiny portion of junk mail, let us say a proportion of $1 : 10,000$. In that case, the most probable scenario is that a classifier learnt using a supervised learning approach will nearly always let the junk mail enter in your mailbox. This problem is called **the class-imbalance problem** [11] and is currently puzzling not only the practitioners of machine learning, but also the machine learning research community.

I hope you fancy reading the proposed new approaches, ideas and contributions regarding these fascinating classification scenarios as much as I have delighted in exploring them during these last few years.

0.1 Overview of the Dissertation

This document is divided into two separate parts: a self-explanatory introduction to the contributions proposed in this dissertation and a collection of scientific publications supporting those contributions.

Part I introduces the main contributions of this PhD work to the machine learning field and is composed of three chapters: Chapter 1 contains a brief self-explanatory notational introduction to statistical classification. Then, Chapter 2 sums up the main four contributions of this dissertation, their origins and conclusions, and the potential future research lines generated from them. These are as follows:

1. Semi-supervised learning approaches to learn multi-dimensional classifiers.
2. A theoretical study on the probability of error of the optimal semi-supervised multi-class learning technique.
3. A framework to theoretically studying the competitiveness of state-of-the-art learning techniques and the adequateness of performance scores in the class-imbalance domain.
4. A robust measure for characterising the class-imbalance extent of multi-class problems.

Finally, some general conclusions and remarks on potential future work are drawn in Chapter 3.

Part II encompasses a collection of four of the published scientific articles in order to give support for the previously mentioned main contributions. This part is divided into Chapter 4, Chapter 5, Chapter 6, and Chapter 7, and they correspond to the contributions 1-4, respectively.

Part I

Introduction to the Main Contributions on Classification

1

A Brief Introduction to Contemporary Classification

This chapter introduces the general notation and framework used throughout the introductory part of this dissertation. Although this notation is based on the published work, it has been defined in order to cover all the presented contributions, which were made in different time periods and to address different kinds of problems. Thus, slight notational differences exist between this introductory part of the global PhD work and the published works. Fortunately, each published article contains its own notational introductory section.

In machine learning and statistics, classification stands for the statistical problem of identifying the values $\mathbf{c} = (c_1, \ldots, c_M)$ of a categorical vector of class variables $\mathbf{C} = (C_1, \ldots, C_M)$ given a set of features $\mathbf{X} = (X_1, \ldots, X_N)$ of a new unlabelled observation $\mathbf{x} = (x_1, \ldots, x_N)$, usually on the basis of a training set of data $\mathcal{D}$ containing observations [12][13]. Here, M represents the number of target variables and N the number of predictive features.

Each class variable, C_j, is a discrete random variable of cardinality $K_j \geq 2$, which takes discrete values from the range $\{c_{(j,1)}, \ldots, c_{(j,K_j)}\}$. Moreover, let

$$K = \prod_{j=1}^{M} K_j \text{ and } \kappa = \max_j K_j$$

be the cardinality of the Cartesian product of all class variables and the maximum cardinality of the class variables, respectively, so that the random variable, $\mathbf{C}$, representing the joint probability of class variables, may also be a discrete random variable with values in the range $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$[1]. Each feature X_i may be a discrete or continuous random variable.

An example of classification could be assigning one or more diagnoses to a given patient who has a described set of characteristics or features (gender,

[1] For simplicity of notation, the class subscript will not be used for the particular and common case of having just one class variable ($M = 1$, the uni-dimensional classical classification). The class variable will be written as C, its realisation as c, and its discrete values as $c_1, \ldots, c_K$, being $K (= \kappa$, in this case) also its cardinality.

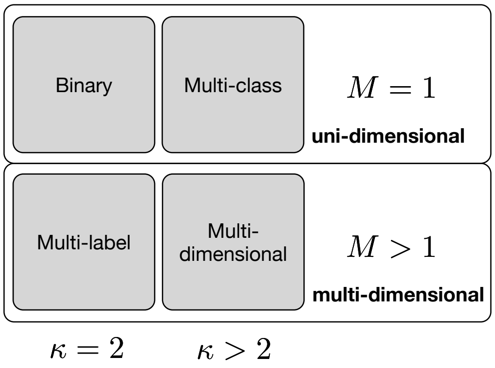


Fig. 1.1: Types of classification problems depending on the shape of the target variables.

blood pressure, glucose level, presence or absence of certain symptoms, etc.), or the spam filtering problem discussed in the Preface.

1.1 Main Types of Classification Problems

Depending on the layout of the random variable representing the vector of class variables $\mathbf{C} = (C_1, \ldots, C_M)$, several major types of classification problems can be defined:

Firstly, although classical classification tasks focus on the prediction of a single class variable C, many real-world domains consider a vector $\mathbf{C}$ composed of more than one class variable. Therefore, it is reasonable to initially split the classification problems into uni-dimensional classification problems ($M = 1$) and multi-dimensional classification problems ($M > 1$). Despite the fact that multi-dimensional problems may include uni-dimensional problems, the latter have been studied longer and, therefore, they can be solved, in spite of its debatable simplicity, by far more specific techniques. Thus, it is worth giving some consideration to the uni-dimensional case in this categorisation.

Secondly, another key differential characteristic regards the number of values that the class variables can take, whether they are all binary ($\kappa = 2$) or whether they contain some multinomial class variables ($\kappa > 2$). Moreover, in this differentiation, the amount of learning techniques used to solve each kind of classification problem is unbalanced; the amount of learning techniques proposed by the community for binary class variables greatly outnumbers the quantity of techniques capable of dealing with multinomial class variables.

Thus, combining these differentiations, four major types of classification problems can be found in the literature. The following paragraphs, which are summarised in Figure 1.1, introduce these major types, shortened by the complexity of their classification task:

The **binary classification problem** ($M = 1, \kappa = 2$) is the best understood and the most studied classification problem in the literature due to its simplicity. Only one class variable C is to be identified from the features and it can take only two different values; C is either equal to c_1 or to c_2. Several examples of binary classification problems can be found in the literature: (i) intrusion detection systems [14] for computer security violations, where a single transaction is classified either as normal or as a threat, (ii) sarcasm recognition [15], where sentences are identified as sarcastic or non-sarcastic, or even the identification of human genes associated with a certain disease [16].

The **multi-class classification framework** ($M = 1, \kappa > 2$), instead, groups all the uni-dimensional classification problems where the binary constraint is not applied, i.e. the single target variable C takes $K > 2$ values. Although one digit is usually enough to bound most multi-class classification problems, e.g. Anderson's Iris data set [17], in this major framework, we can also find highly multi-class problems with a large set of possible class values ($K >> 2$): from classifying a description of a flower as one of the over $300,000$ known flowering plants [18] to classifying a whistled tune as one of the over 30 million recorded songs [19].

A **Multi-label classification problem** ($M > 1, \kappa = 2$) is the multi-dimensionalisation of a binary classification problem [20]: several binary class variables which may be statistically dependent, $M > 1$ and $\kappa = 2$, must be simultaneously inferred from the same set of features. Formally, each C_j is either equal to $c_{(j,1)}$ or $c_{(j,2)}$. This kind of problem can be found in many applications [13]: a text document or a semantic scene can be assigned to multiple topics, a gene can have multiple biological functions, a patient may suffer from multiple diseases, a patient may become resistant to multiple drugs for HIV treatment, a physical device can break down due to multiple components failing, etc.

Multi-dimensional classification ($M > 1, \kappa > 2$) encompasses the classification problems where a set of probably dependent class variables has to be simultaneously identified from the same set of predictive features, $M > 1$, and their cardinalities may take any value equal or greater than 2, $\kappa > 2$. Examples of this kind of classification problems can be found in [21], [22] or [23], where several intricate defects can appear in a stainless steel plate, several problems in the fish recruitment forecasting field, or different attitudes of the author (subjectivity, sentiment polarity level, will to influence) are characterised from a given online text, respectively.

Roughly, the theoretical studies and methodological proposals for learning a classifier in the multi-class framework can be applied for binary problems, the ones used for multi-label problems can also be used for binary problems, and multi-dimensional research can be directly applied to either binary, multi-class and multi-label problems. Moreover, by parameterising multi-class problems as $(M = 1, \kappa \geq 2)$, multi-label as $(M \geq 1, \kappa = 2)$, and multi-dimensional as $(M \geq 1, \kappa \geq 2)$, multi-class and multi-label may encompass binary problems, and the multi-dimensional classification framework may include the whole classification spectrum. The remaining potential appliances are, in general, futile; they can only be assured for a limited set of specific scenarios. The interested reader can find further information on the unfeasibility of using approaches for multi-class and multi-label classification to tackle multi-dimensional classification problems in [23].

Due to the fact that the given definition of multi-dimensional classification problems includes the whole classification spectrum, in the introductory part of this dissertation, I will use it to describe the mathematical framework encompassing the contributions introduced.

1.2 Learning of Classifiers

Formally, let us assume we are provided with a M-dimensional classification problem γ with a generative model $\rho(\mathbf{x}, \mathbf{c})$ which not only specifies how data is generated from the symptoms (features) but also provides prior information about the distribution of the different class variables. This generative model is given by the following generalised joint probability density function:

$$\rho(\mathbf{x}, \mathbf{c}) = p(\mathbf{c})\rho(\mathbf{x}|\mathbf{c}). \tag{1.1}$$

Here, $p(\mathbf{c})$ is a discrete distribution symbolising the joint probability distribution of the vector of class variables $\mathbf{C}$ and $\rho(\mathbf{x}|\mathbf{c})$ is the conditional distribution of the feature space. These generative probability distributions may be parametrised by certain model parameters in some of the publications. They can be sometimes written as $\rho(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})$ and $\rho(\mathbf{x}|\mathbf{c}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the set of model parameters of the conditional probability distributions.

Then, solving a M-dimensional classification problem is equivalent to defining, learning or inferring a function $\Psi(\mathbf{x}) = \hat{\mathbf{c}}_{\Psi}$ able to make predictions from features. This function is known as a classifier and maps a given vector of observations $\mathbf{x}$, drawn from the generative function of eq. (1.1), into an instantiation $\mathbf{c}$ of the vector of the class variables, written as $\hat{\mathbf{c}}_{\Psi}$, i.e.

$$\begin{aligned} \Psi : x_1 \times \ldots \times x_N &\mapsto \{1, \ldots, K_1\} \times \ldots \times \{1, \ldots, K_M\}. \\ \mathbf{x} &\mapsto \mathbf{c} \end{aligned} \tag{1.2}$$

When the generative model $\rho(\mathbf{x}|\mathbf{c})$ is known and computations are possible, the generative probability distributions can be directly used to define a rule capable of deciding to which configuration of class values an unseen instance belongs to. However, the usage of these distributions in a decision rule depends on how costly it is to make wrong predictions using the features [24]. Examples of classification or decision problems with different costs can be easily found in the literature: For instance, in the spam filtering example;

it is more costly to filter out an important email than to let a junk mail enter your mailbox. Another example is the Pascal's wager [25], marked as the first formal use of decision theory [26]. There, Pascal claimed that as long as there is a positive probability of the existence of God, disbelieving in God always comes with a higher cost than believing in God, since in the event of the existence of God, a belief (prediction of existence) means an eternity in heaven and a disbelief (prediction of non-existence) means an eternity in hell. In the event of non-existence, both decisions of belief and disbelief have very small losses and gains, respectively. These costs, which are associated which every correct and incorrect prediction made by the classifier, are usually defined in a loss function [27]. Formally, this loss function, $L(\Psi(\mathbf{x}), \mathbf{c}_i) = \ell$, is the cost associated to the decision $\Psi(\mathbf{x}) = \hat{\mathbf{c}}_\Psi$ when the true class vector is $\mathbf{c}_i \neq \hat{\mathbf{c}}_\Psi$. As previously mentioned, it has a deep influence on the learning of the classifier, and, thus, in its resulting performance.

If we are dealing with a classification problem where the costs of misclassifying any class variable are symmetrical and equal, and there is no cost for a correct classification, we are assuming the well-known 0-1 loss function ($\ell = 0$ if $\hat{\mathbf{c}}_\Psi = \mathbf{c}_i$, and $\ell = 1$ otherwise, [27]). In a such situation, the Bayes decision rule [28] can be directly used to classify new instances as it inherently minimises this loss function. Formally, it is defined as follows:

Definition 1. *Assuming the generative model, $\rho(\mathbf{x}, \mathbf{c})$, to be known, the* ***Bayes decision rule*** *(BDR) is given by*

$$\hat{\mathbf{c}}_B = \arg\max_i p(\mathbf{c}_i)\rho(\mathbf{x}|\mathbf{c}_i). \tag{1.3}$$

Here, $\hat{\mathbf{c}}_B$ is the categorical class vector assigned by the BDR to the observation $\mathbf{x}$. This rule has a corresponding probability of error

$$e_B = 1 - \sum_{i=1}^{K} p(\mathbf{c}_i) \int\limits_{\Omega_i} \rho(\mathbf{x}|\mathbf{c}_i) d\mathbf{x} \tag{1.4}$$

which is called the Bayes error and is the highest lower bound of the probability of error of any classifier as proved in [28] and in [13]. In equation 1.4,

$$\Omega_i = \{\mathbf{x} : p(\mathbf{c}_i)\rho(\mathbf{x}|\mathbf{c}_i) - \max_{i' \neq i} p(\mathbf{c}_{i'})\rho(\mathbf{x}|\mathbf{c}_{i'}) > 0\} \tag{1.5}$$

represents the region where $p(\mathbf{c}_i)\rho(\mathbf{x}|\mathbf{c}_i)$ is maximum and, so, the instances are assigned to the vector of class variables $\mathbf{c}_i$ by the BDR, for all i.

The BDR is the most adequate classifier when the total number of correct classifications is to be maximised. Hence, it has enjoyed an excellent academical position among the traditional machine learning techniques [29]. However, as previously put forward, in real-life problems, some types of misclassifications are more crucial – they come with higher misclassification costs – than others. Note that the BDR does not guarantee the correct prediction of the classes with a low probability of occurrence since the prediction of the most probable class values produces a greater overall number of correct classifications than correctly predicting the least probable values. Due to this, other classifiers appear to overcome the limitations of the BDR, such as the other omniscient protagonist of this dissertation; the EDR.

Definition 2. *Assuming the generative model to be known, the **equiprobable Bayes decision rule** (EDR) is given by*

$$\hat{\mathbf{c}}_E = \arg\max_i \rho(\mathbf{x}|\mathbf{c}_i). \tag{1.6}$$

Here, $\hat{\mathbf{c}}_E$ is the categorical class vector assigned by the EDR to the observation $\mathbf{x}$. This rule has an associated probability of error

$$e_E = 1 - \sum_{i=1}^{K} \int_{A_i} \rho(\mathbf{x}|\mathbf{c}_i) d\mathbf{x} \tag{1.7}$$

which corresponds to the arithmetical mean of the error in not classifying each configuration of class vector as so. This error is the highest lower bound for this arithmetical mean of any classifier as proved in this dissertation [30]. There,

$$A_i = \{\mathbf{x} : \rho(\mathbf{x}|\mathbf{c}_i) - \max_{i' \neq i} \rho(\mathbf{x}|\mathbf{c}_{i'}) > 0\} \tag{1.8}$$

is the region where $\rho(\mathbf{x}|\mathbf{c}_i)$ is maximum and, so, the instances are assigned to the vector of class variables $\mathbf{c}_i$ by the EDR, for all i.

In order to clarify the behavioural difference between both decision rules (BDR and EDR), let us consider the example illustrated in Figure 1.2: imagine we face a uni-dimensional binary classification problem with a generative model given the following mixture joint density distribution [31]:

$$\rho(x, c) = p(c_1)\rho(x|c_1)\mathbb{1}(c = c_1) + p(c_2)\rho(x|c_2)\mathbb{1}(c = c_2). \tag{1.9}$$

There, $\mathbb{1}(E)$ is the indicator function with a value of 1 if the event E is true, and 0 otherwise. This model is parametrised as follows: $p(c_1) = 0.85$, $p(c_2) = 0.15$, and the conditional distributions follow two different normal distributions; $\rho(x|c_1) \sim N(3, 0.75)$ and $\rho(x|c_2) \sim N(5, 0.75)$. Figure 1.2 plots the generative mixture model of the example; while the upper chart only shows the mixture density distribution of both conditional feature probability distributions – $\rho(x|c_i)$ –, the lower chart shows the mixture density distribution of the generative model $\rho(x, c)$, i.e. the conditional feature probability distributions multiplied by their class probabilities, – $p(c_i)\rho(x|c_i)$ –. Both views are displayed in order to show the different behaviour of both decision rules.

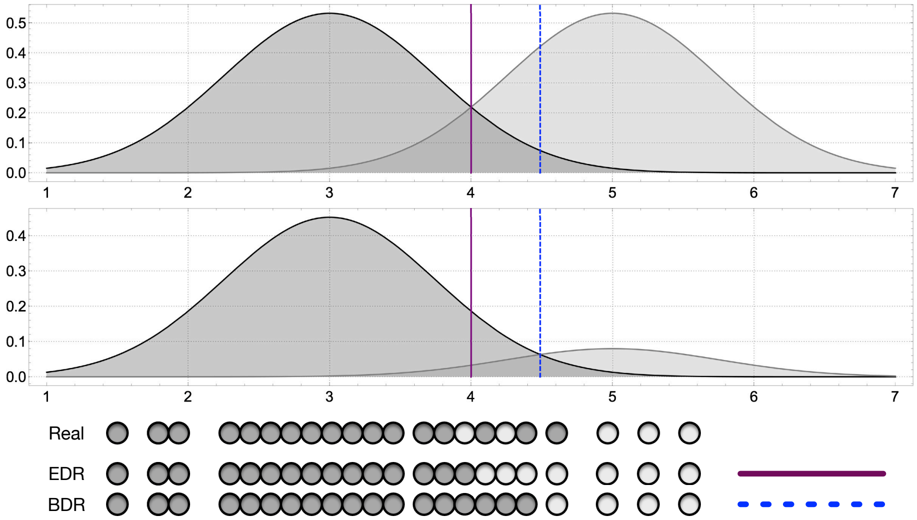


Fig. 1.2: Example of the decision regions provided by both the EDR and the BDR to classify a set of observations from a given binary toy problem $p(x, c)$ parametrised as: $p(c_1) = 0.85$, $p(c_2) = 0.15$, $p(x|c_1) \sim N(3, 0.75)$, and $p(x|c_2) \sim N(5, 0.75)$.

Here, each decision rule can be simplified to a single value d in the abscissa, splitting the feature space into two decision regions: $(-\infty, d)$ for the points to be classified as c_1 and $[d, \infty)$, the other, for the points to be classified as c_2. The limiting points of the BDR and EDR, which are drawn with a blue dashed line and a continuous purple line in the figure, are $d = 4.5$ and $d = 4$, respectively. It is trivial to obtain those values from both Definition 1 and Definition 2, respectively. As can be seen, the EDR divides the feature space by density dominance in the upper chart and the BDR behaves equivalently in the lower chart. By means of just this perception, it can be easily deduced that the choice of one of these omniscient theoretical classifiers will impact the performance of the classification: imagine we are supplied with a testing sample of the generative problem composed of 22 instances – 17 of c_1 and 5 of c_2 – marked as Real in the figure. Note that the values x of these instances are given by their position with respect to the abscissa and their classes by their color; dark grey corresponds to c_1 and white to c_2. The EDR will classify those instances as the sample row marked as EDR and the BDR as the sample row marked as BDR. Then, it can be perceived that BDR obtains a greater overall performance by correctly classifying 19 out of 22. The EDR, instead, aims at balancing the number of correct classifications for both classes simultaneously. This results in a degradation of the overall performance as it misclassifies more instances of the class c_1. Favourably, a better performance is obtained for the minority class value c_2 using this rule, 4 matches instead of 3.

In short, this example shows us that the BDR fits in with problems where the overall number of correct classifications has to be maximised. However, it might ignore minority classes. On the contrary, the EDR is a proper classifier for atypical events since it balances the number of correct classifications among all classes including classes with low probability. Problems such as the stated junk mail filter may benefit from this property. As a drawback, it tends to increase the number of misclassifications in the most probable classes.

Many other decision rules can be defined in order to create classifiers which favour certain classifications over others. The Bayes decision theory [26] is the field studying this type of all-knowing classifiers. Authors skilled in this field may argue that, both the BDR and the EDR are versions of the same decision rule, called the (log-)likelihood ratio test [31]. This assertion remains within this field due to the fact that it usually narrows the classification spectrum to just uni-dimensional binary problems as most of the calculi becomes intractable when the uni-dimensional binary constraint is relaxed. In this dissertation, I make use of two different definitions, since the likelihood ratio test loses the comprising property for similar decision rules but with different costs when $K > 2$ or $M > 1$. Interested readers in the (Bayes) decision theory, other existing loss functions, and decision rules can find an interesting introduction in [32].

Unfortunately, the Bayes decision theory assumes a knowledge over the faced classification problem which in most of the real-world cases is not often achievable: the generative model, along with the real class distribution, is usually unknown. Fortunately, a random sample of the distribution of the generative model, $\mathcal{D}$, is often available. Thus, practitioners must infer a classifier Ψ using a deterministic learning algorithm $\mathbb{A}$ over the training dataset $\mathcal{D}$, i.e. $\mathbb{A}(\mathcal{D}) = \Psi$, in order to classify future unseen unlabelled instances. These learning algorithms are usually defined to asymptotically (as $|\mathcal{D}| \to \infty$) converge to the behaviour of a decision rule, such as the BDR or the EDR, by minimising a certain loss function [28]. Thus, idealistically, the resulting classifiers will have a similar, but not greater, expected performance. However, finite samples usually introduce some bias in the learning task decreasing the classification performance. On the other side, we have the theoretical baseline classifiers, which established the lowest expected performance of a classifier to be considered competitive. The most used baseline classifier, which uses no training data ($\mathcal{D} = \emptyset$), is the random guessing of the class variables:

Definition 3. *The classifier representing the random guessing of K different configurations of M class variables is known as the **uniformly random classifier** (RAND) and it is given by*

$$\hat{\mathbf{c}}_R = \mathit{Unif}\{1, K\}. \tag{1.10}$$

where $\mathit{Unif}\{1, K\}$ is the discrete uniformly random function which assigns a categorical class configuration $\mathbf{c}_i$ to an example $\mathbf{x}$ with probability $1/K$. Both the probability of error and the arithmetical mean of the error in classifying *each configuration of class vector of this classifier coincides and are equal to*

$$\frac{K-1}{K}. \tag{1.11}$$

Having introduced the theoretical performance bounds for the learnt classifiers, the following sections present a review of the most common types of training datasets used in the literature and the most well-known learning algorithms used to deal with each type of training datasets. Moreover, in Section 1.4, several characteristics of the finite training datasets which hamper the classification performance are discussed.

1.2.1 Structure of the Available Training Dataset

Learning involves the ability to generalise from past experience in order to deal with new situations that are related to this experience [33]. In statistical classification and when the generative model is unknown, this experience comes in the form of a training dataset $\mathcal{D}$, which may have different shapes. Depending on this, different learning approaches are used. Formally, let $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$ be defined as the available training dataset which may be composed of the following two different sets:

- Let $\mathcal{L} = \{(\mathbf{x}^{(1)}, \mathbf{c}^{(1)}), \ldots, (\mathbf{x}^{(l)}, \mathbf{c}^{(l)})\} = \{(\mathbf{x}^{(n)}, \mathbf{c}^{(n)})\}_{n=1}^{l}$ be defined as a **supervised training dataset** of size l drawn from the generative function. There, let the class labels $\{\mathbf{c}^{(n)}\}_{n=1}^{l}$ be i.i.d. random values drawn from $p(\mathbf{c})$ and let each observation $\{\mathbf{x}^{(n)}\}_{n=1}^{l} \in \mathcal{L}$ be also an i.i.d. random value but drawn from $\rho(\mathbf{x}|\mathbf{c})$. In other words, a supervised training dataset is a set of observations where each observation not only contains the features but also the real classes of the observation.
- Let $\mathcal{U} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(u)}\} = \{\mathbf{x}^{(m)}\}_{m=1}^{u}$ be defined as an **unsupervised training dataset** of size u drawn from the generative function, where each observation $\{\mathbf{x}^{(m)}\}_{m=1}^{u} \in \mathcal{U}$ is also an i.i.d. random value but drawn from $\rho(\mathbf{x})$. In this case, the values of the classes for the observations are not given, only features are available.

While other richer and more complete ways exist to taxonomise the spectrum of learning scenarios [34]; by means of my definition, three different kinds of training datasets, and, thus, three classical major learning scenarios, which ideally fit and cover the current PhD work, can be defined:

1. The most extensive work carried out in the machine learning community assumes a labelled training dataset ($\mathcal{D} = \mathcal{L}$) to be available [35]. Under this assumption, the goal is to infer the function $\Psi(\mathbf{x})$ as defined in equation 1.2. This problem is known as **supervised learning** and the used learning algorithm must be able to generalise from $\mathcal{D}$ to unseen situations in a "reasonable" way [33].

2. On the opposite side of the spectrum, an unlabelled dataset may be provided. Here, the only possible goal is to find interesting coherent structures in the data $\mathcal{D} = \mathcal{U}$. In other words, the problem of **unsupervised learning** is fundamentally that of estimating a density $\rho(\mathbf{x}) = \sum_{i=1}^{K} \rho(\mathbf{x}|\mathbf{c}_i)p(\mathbf{c}_i)$ which is likely to have generated the training dataset [10]. The density is written as a marginal density distribution of the generative model due to the fact that the broadly used clustering technique [35, 12] aims at not only determining the number of classes present in the density, but also at discovering the feature distributions of these classes. However, as proved later in this dissertation, having no labels in the training datasets produces classifiers no better than the RAND [36]. Other forms of unsupervised learning are, for instance, outlier detection or unsupervised dimensionality reduction. Since, apart from clustering, no other unsupervised learning directly relates to classification, this major learning technique will not be further discussed in this dissertation.
3. Somewhat in the middle, the **semi-supervised learning** scenario arises [10, 37, 38]. This problem assumes that the available training dataset is composed of both labelled and unlabelled examples ($\mathcal{D} = \mathcal{L} \cup \mathcal{U}$). An important motivation for dealing with this kind of datasets is that, in some applications, gathering labels is relatively expensive or time-consuming, compared to the cost of obtaining an unlabelled example. Consider, for instance, the example of building a webpage classifier exposed in [39]. Whilst downloading billions of unlabelled webpages is straightforward, reading them to assign labels is time-consuming. So, why not use unlabelled observations to improve the performance of supervised learning methods that only use a small labelled set, $\mathcal{L}$, to train a classifier?

The following sections exhaustively review both supervised and semi-supervised learning scenarios and their most important learning paradigms.

1.2.2 Major supervised learning paradigms

Roughly, the supervised learning process is carried out by means of an optimisation algorithm provided with fully annotated training data, $\mathcal{D} = \mathcal{L}$, drawn from eq. (1.1). It attempts to infer a mapping function Ψ that minimises a certain loss function [27]. Here, most of the traditional learning algorithms use the 0-1 loss or a surrogate loss which, by providing an upper bound for it, is also expected to minimise the 0-1 loss [40]. Thus, these learning algorithms usually have an asymptotic behaviour to the BDR. According to [41], there are six major supervised learning paradigms: (i) decision trees, (ii) rules extracted from decision trees, (iii) rule-based machine learning, (iv) statistical learning, (v) connectionism, and (vi) probabilistic approaches. I, however, perceive rules extracted from decision trees and rule-based machine learning as similar paradigms to decision trees. So, therefore, in this introductory part, I only tackle the other remaining five approaches.

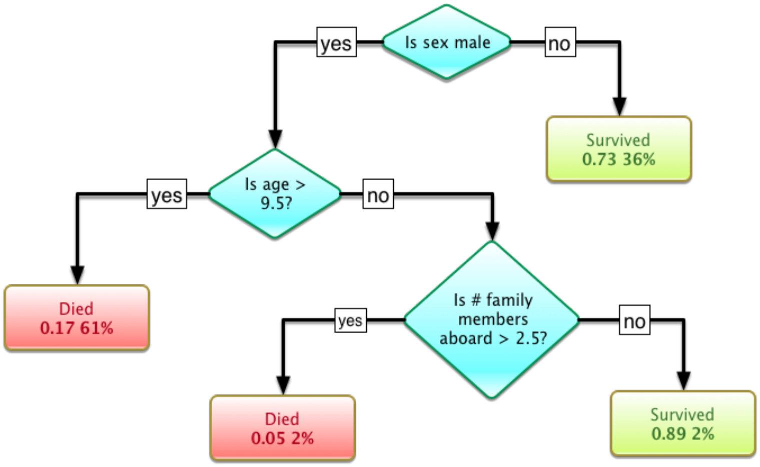


Fig. 1.3: Using just the gender, the age and the number of siblings and parents combined as features, a classifier can be learnt in order to predict whether the person survived or died in the Titanic accident.

The most common approaches to supervised learning are the paradigms which produce a rule, Ψ, in the form of a tree-like model known as a **decision tree** in order to discriminate among the values of the classes of the objects. C4.5 [42] is the most utilised learning algorithm for decision trees. The nodes in a decision tree correspond to selected object features, and the edges correspond to predetermined alternative values for these attributes. Leaves of the tree correspond to sets of objects with an identical classification vector of classes [43]. Figure 1.3 shows an example of a decision tree learnt from a supervised dataset composed of the list of RMS Titanic passengers and their main features. Rules extracted from decision trees (e.g. C4.5Rules) and **rule-based machine learning** (e.g. RIPPER [44]), instead, aim at producing classifiers which typically take the form of a set of nested if-then expressions, i.e. {IF 'condition', THEN 'result'}. It is trivial that the decision tree presented in Figure 1.3 can be straightforwardly transformed into the set of rules proposed in Algorithm 1. A deep coverage of decision trees and rule/based machine learning can be found in [43], the book used to write the previous lines.

Statistical learning [45] is a paradigm for machine learning drawing from the fields of statistics and functional analysis. Probably, the most used methods within this paradigm are support vector machines (SVM) classifier, its multiple variants and similar kernel-based learning methods. Interested readers of this learning strategy may refer to [46]. Briefly, support vector machines find linear boundaries in their input space. However, not all feature spaces for binary problems are linearly separable [47]. Fortunately, as happens

Algorithm 1 Determining whether the person survived or died in the Titanic accident

```
if $Sex = Male$ then
  if $Age > 9.5$ then
    Died
  else
    if $FamilyMembers > 2.5$ then
      Died
    else
      Survived
    end if
  end if
else
  Survived
end if
```

with other linear methods, we can make the classifier more flexible by enlarging the feature space using basis expansions to a larger dimension, infinite in some cases. Generally linear boundaries – hyperplanes – in the enlarged space achieve better training-class separation, and translate to nonlinear boundaries in the original space. This procedure is illustrated in Figure 1.4.

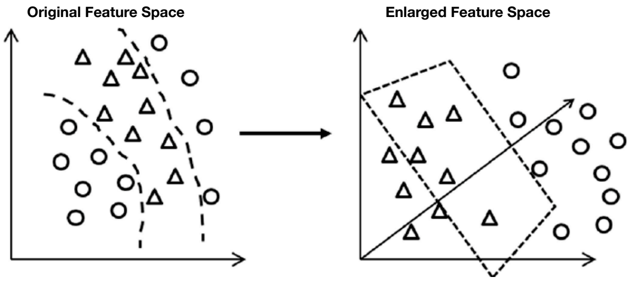


Fig. 1.4: How a paradigm such as SVM achieves a linear separability of the space in an enlarged feature space. Image extracted from [48].

An important drawback of these classifiers based on the linear separability of the feature space is that they are originally designed to solve uni-dimensional binary classification problems. However, steps towards covering uni-dimensional multi-class problems are being proposed [49].

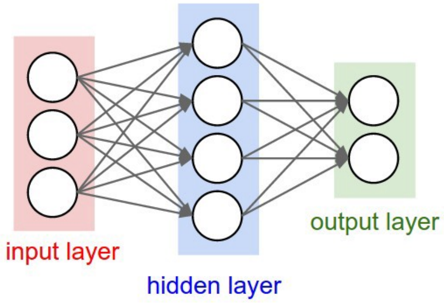


Fig. 1.5: Structure of an artificial neural network.

Connectionism covers a set of approaches in the field of artificial intelligence that attempts to represent mental or behavioural phenomena as emergent processes of interconnected networks of simple units [50]. **Artificial neural networks** [51, 52] are, currently and by far, the most popular connectionist model used. They produce classification models vaguely inspired by the biological neural networks that constitute animal brains in order to categorise instances into classes. The typical structure of a neural network classifier can be viewed in Figure 1.5. There, each circular node represents an artificial neuron and an arrow represents a connection of one artificial neuron to the input of another. The neurons, which are sorted in layers, are capable of receiving, processing and transmitting a signal. As can be seen, to perform a classification, three kinds of layers are required:

- The *input layer* serves as an input interface for the whole network. It feeds the next (hidden) layer with the features of a new unseen instance.
- After a learning process which typically deals with the modification of weights and activation thresholds by using a training dataset $\mathcal{D}$, the *hidden layer* encapsulates several functions and weights over the inputs, i.e. the model parameters which may or not activate an output signal on the basis of the activation thresholds. Therefore, these neurons may produce different signals and transmit them to the following layer. This hidden layer may be present or absent in the network. Also, there might be several hidden layers consecutively connected.
- Finally, the interface which has no successor is called the *output layer*. This layer collects all the signals from the previous (hidden) layer and returns the final prediction of the classification model.

Deep artificial neural networks, which are neural networks with a very large number of hidden layers have attracted a great attention in the past few years. Although powerful computer hardware such as GPU (Graphical Processing Units) existed [53], not until 2012, when significant impacts in image and text recognition fields were shown, can we talk of a real **deep learning** revolution [54, 55, 56]. In Figure 1.6, we can see a schematic example of a deep neural

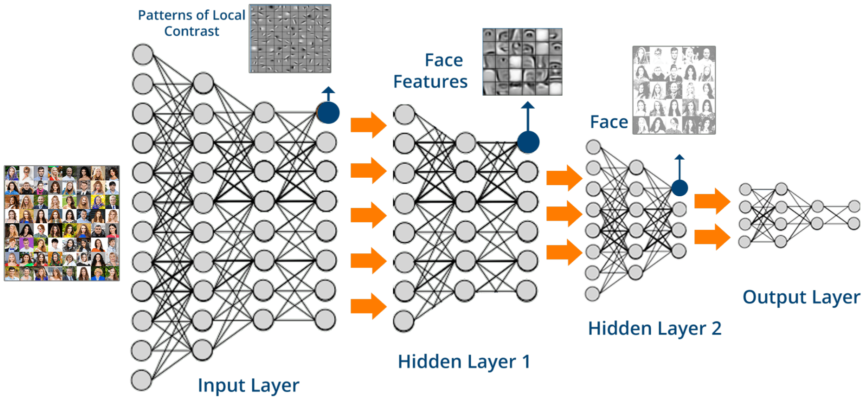


Fig. 1.6: A deep artificial neural network to face recognition from the online book *Neural Networks and Deep Learning*.

network architecture with its multiple hidden layers to classify images, in this case to recognise different faces.

To the best of my knowledge, neither of the above discussed paradigms is able to directly tackle *multi-dimensional classification problems*. To do so, they either deal with each class variable in isolation and solve as many classification problems as class variables are present, or treat each configuration of the vector of class variables as a different class value facing a highly multi-class problem. Both approaches have important drawbacks: while the former leaves out the dependencies among the class variables, the latter may suffer an unfeasible number of combinations. Research on **probabilistic approaches**, instead, recently, has found a reliable and efficient way to deal with multiple class variables [57, 13]. These approaches make use of an induction algorithm over the annotated training data in order to learn a probability distribution $\hat{\rho}(\mathbf{x}, \mathbf{c})$ or $\hat{\rho}(\mathbf{c}|\mathbf{x})$. Note that the former distribution is a direct estimation of the generative model defined in equation 1.1. Afterwards, probabilistic classifiers predict the values of the class variables of new unlabelled instances based on a classification rule over the previous probability distribution. An example of classification rule is the joint classification rule [58] for multi-dimensional problems:

$$\hat{\mathbf{c}} = \arg\max_{i} \hat{\rho}(\mathbf{c}_i|\mathbf{x}). \tag{1.12}$$

This rule, which obtains better results than several other already proposed rules [58], returns the most probable combination of values for the vector of class values.

The taxonomy of the probabilistic approaches depends on the assumed family of distributions of the classification rule. Within these approaches, Bayesian networks (Naïve Bayes, in particular) are the most well-known due to the fact that they are powerful tools for knowledge representation and inference under uncertainty conditions [59]. Moreover, efficient algorithms to infer knowledge from data are available in the literature [60]. Since these formalisms have a great presence in this dissertation, I devote the following section to a detailed description of how they are used as a learning approach.

1.2.2.1 Multi-dimensional Bayesian Network Classifiers

Bayesian networks have been extensively used as classifiers [61, 62] and have become a classical and well-known classification paradigm. In [13], they are defined as:

Definition 4. *A* ***Bayesian network*** *over a finite set* $\mathbf{V} = \{Z_1, \ldots, Z_n\}$, $n \geq 1$, *of nominal random variables is a pair* $\mathbb{B} = (\mathcal{S}, \boldsymbol{\theta})$ *where* $\mathcal{S}$ *is a directed acyclic graph (DAGs) whose vertices correspond to the random variables and* $\boldsymbol{\theta}$ *is a set of parameters* $\theta_{z|\mathbf{pa}(z)} = p(z|\mathbf{pa}(z))$, *where* $\mathbf{pa}(z)$ *is a value of the set of variables* $\mathbf{Pa}(Z)$, *parents of the* Z *variable in the graphical structure* S. *Thus, the network* $\mathbb{B}$ *defines a joint probability distribution*[2] $p_{\mathbb{B}}$ *over* $\mathbf{Z}$ *given by*

$$p_{\mathbb{B}}(z_1, \ldots, z_n) = \prod_{i=1}^{n} \theta_{z_i|\mathbf{pa}(z_i)} \tag{1.13}$$

Conventionally, a Bayesian network classifier is a Bayesian network specially designed to solve classification problems in which instances described by a set of discrete features have to be assigned to a class value. In my broader vision of the classification spectrum, where an instance may be simultaneously assigned to several class variables, the term multi-dimensional Bayesian network classifier [57][63] is used, instead, to denote the generalisation of the classical Bayesian network classifiers. These classifiers model the relationships between the variables by means of directed acyclic graphs (DAG) over the class variables and over the feature variables separately, and, then, connect both sets of variables by means of a bi-partite directed graph.

Definition 5. *A* ***multi-dimensional Bayesian network classifier*** *is a Bayesian network* $\mathbb{B} = (\mathcal{S}, \boldsymbol{\theta})$, *where the DAG structure* $\mathcal{S} = (\mathbf{V}, \mathbf{A})$ *has the set* $\mathbf{V}$ *of discrete random variables partitioned into the sets* $\mathbf{V}_C = \{C_1, \ldots, C_M\}$,

[2] Usually, Bayesian networks assume nominal or discrete random variables. For this reason, I refer to the probabilities as $p(\cdot)$ instead of $\rho(\cdot)$. For numerical random variables, other probabilistic graphical models are used.

$M \geq 1$, of class variables, the set $\mathbf{V}_F = \{X_1, \ldots, X_N\}$ $(N \geq 1)$ of features, and the set of arcs $\mathbf{A}$ can be partitioned into the following three sets:

- *$\mathbf{A}_{CF} \subseteq \mathbf{V}_C \times \mathbf{V}_F$ is composed of the arcs between the class variables and the feature variables, so we can define the feature selection subgraph of S as $S_{CF} = (\mathbf{V}, \mathbf{A}_{CF})$. This subgraph represents the selection of features that seems relevant for classification given the class variables.*
- *$\mathbf{A}_C \subseteq \mathbf{V}_C \times \mathbf{V}_C$ is composed of the arcs between the class variables, so we can define the class subgraph of S induced by $\mathbf{V}_C$ as $S_C = (\mathbf{V}_C, \mathbf{A}_C)$.*
- *$\mathbf{A}_F \subseteq \mathbf{V}_F \times \mathbf{V}_F$ is composed of the arcs between the feature variables, so we can define the feature subgraph of S induced by $\mathbf{V}_F$ as $S_F = (\mathbf{V}_F, \mathbf{A}_F)$.*

Then, the joint probability distribution over $\mathbf{V}$ is defined as

$$p_{\mathbb{B}}(\mathbf{x}, \mathbf{c}) = \prod_{j=1}^{M} \theta_{c_j|\mathbf{pa}(c_j)} \prod_{i=1}^{N} \theta_{x_i|\mathbf{pa}(x_i)}, \tag{1.14}$$

and the corresponding classifier is defined by the following classification rule as

$$\hat{\mathbf{c}}_{\mathbb{B}} = \arg\max_i p_{\mathbb{B}}(\mathbf{c}_i|\mathbf{x}) = \arg\max_i p_{\mathbb{B}}(\mathbf{c}_i) p_{\mathbb{B}}(\mathbf{x}|\mathbf{c}_i). \tag{1.15}$$

Figure 1.7 shows a multi-dimensional class Bayesian network classifier with 3 class variables and 5 features, and its partition into the three subgraphs.

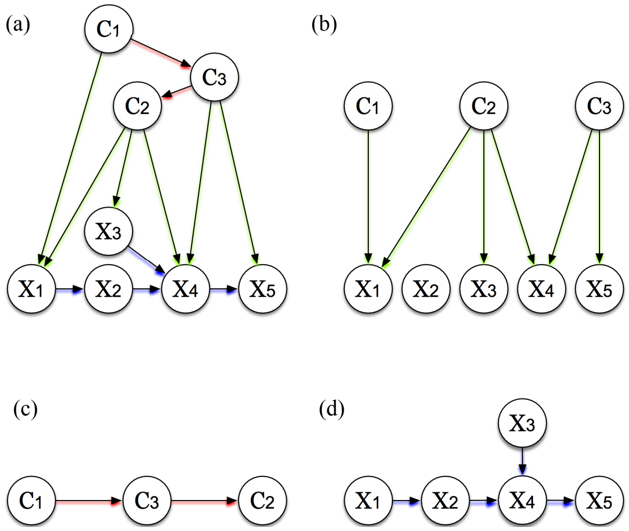


Fig. 1.7: A multi-dimensional Bayesian network classifier and its division: (a) Whole graph, (b) feature selection subgraph, (c) class subgraph, and (d) feature subgraph.

Depending on the structure of the three subgraphs, several sub-families of multi-dimensional class network classifiers, which are consistent with their broadly used uni-dimensional counterparts, are proposed in the state-of-the-art literature [57, 58]:

- *Multi-dimensional naïve Bayes classifier (MDnB)*: the class subgraph and the feature subgraph are empty and the feature selection subgraph is complete. This is the multi-dimensionalisation of the well-known naïve Bayes classifier [4].
- *Multi-dimensional tree-augmented Bayesian network classifier (MDTAN)*: both the class subgraph and the feature subgraph are directed trees. It could be viewed as the multi-dimensional version of the (uni-dimensional) tree-augmented Bayesian network classifier (TAN) proposed in [64].
- *Multi-dimensional J/K dependences Bayesian classifier (MD J/K)*: This structure is the multi-dimensional generalisation of the well-known K-DB [65] classifier. It allows each class variable C_i to have a maximum of J dependences with other class variables C_j, and each predictive variable X_i to have, apart from the class variables, a maximum of K parents.

Whilst induction algorithms to learn $p_{\mathbb{B}}(\mathbf{x}, \mathbf{c})$ for the first two structures are already proposed in the literature [57], the MD J/K (already defined in literature [58]) lacked that characteristic until the proposed manuscript [23]. Moreover, the presented work also addresses the computational burden of the algorithms proposed in [57] by allowing our induction algorithm to efficiently perform a structural learning for MDTAN classifiers. Thus, in this dissertation, the main sub-families of multi-dimensional class Bayesian network classifiers were equipped with efficient learning algorithms.

1.2.3 The semi-supervised learning problem

Semi-supervised learning is a weakly supervised problem [34] concerned with using unlabelled examples, that is, examples for which we know the features but not their corresponding categories, to improve the performance of supervised learning methods that only use a limited labeled set to train a model [39]. There are mainly two different approaches which can tackle this problem [10]:

1. The usage of unsupervised techniques – mainly clustering – over $\mathcal{U}$ to estimate the density $\rho(\mathbf{x}) = \sum_{i=1}^{K} \rho(\mathbf{x}|\mathbf{c}_i)p(\mathbf{c}_i)$ which is likely to have generated the training dataset to, afterwards, use the annotated training dataset, $\mathcal{L}$, to label the regions of the density.
2. Use $\mathcal{L}$ to learn a classifier in a supervised manner and, then, use the unlabelled data to improve that initial classifier.

In this dissertation, both approaches were dealt with. Whilst the former is assumed in my theoretical paper about semi-supervised learning [36] included in Chapter 5, the latter is used to solve a real-world semi-supervised learning problem in the included manuscript [23], which is shown in Chapter 4.

Theoretically, the problem has been extensively studied in the literature. Works such as [31, 38, 66, 67, 68, 69] address and give answers to several interesting questions: (i) *How many unlabelled examples is each labelled instance worth in terms of performance?* (ii) *How the probability of error of the optimal procedure which may be proposed in this framework varies on the values of l and u of labelled and unlabelled examples, respectively?* (iii) *What is the real contribution of the unlabelled training subset in the parameter estimation of the model for the classifier?* (iv) *Does unlabelled data always help?* (v) *What is the relationship between the learnt model misspecification with respect to the generative model and the performance degradation perceived in the classifier?* (vi) *What is the convergence rate to the minimum error of both labelled and unlabelled subsets?* However, all these previous works only focus on binary classification problems. In the literature, only [70] faces the whole uni-dimensional classification range. Due to this, in this dissertation, I devote some research to proving that some answers provided by the previous works do not straightforwardly generalise to the multi-class case. Moreover, in Chapter 5, I contribute to the literature with an optimal semi-supervised procedure for the whole uni-dimensional classification problem and a study of the convergence of probability of error for the optimal classifier. Thus, fundamental limits in the performance of any multi-class semi-supervised classifier may be established.

Methodologically, a set of diverse paradigms to semi-supervised learning has been proposed in the literature due to the high demand for solutions to a multitude of practical problems where it is relatively expensive to produce labeled data [10, 71], e.g. the automatic classification of web pages or automatic medical diagnoses.

Unlike what happens in supervised learning, the proposal of semi-supervised learning paradigms is still a subject undergoing intense study. While semi-supervised learning may seem to be obviously helpful in any scenario, the fact that these methods – $\mathbb{A}_{\mathrm{SSL}}(\mathcal{L} \cup \mathcal{U})$ – can actually lead a worse performance than their supervised counterparts – $\mathbb{A}_{\mathrm{SUP}}(\mathcal{L})$ – has been both widely observed and described [72]. Due to this undesirable characteristic, a lot of effort is being made in proposing "safe" semi-supervised learning algorithms [73, 74].

Regarding the taxonomy of the semi-supervised learning setting, the paradigms may be arranged in the following order: from (i) the techniques where the improvement of performance is achieved by trial-and-error processes, followed by (ii) the techniques where a set of hard-to-check assumptions must hold to achieve improvements, and ending in (iii) the so-called robust or safe semi-supervised techniques which ensure performance gain without relying on assumptions that are not intrinsic to the classifier in hand.

Early semi-supervised learning proposals were trial-and-error processes such as **self-training** or **expectation-maximisation (EM) algorithm based approaches** [38, 75]. With the main advantage that these approaches can be applied to any supervised learning algorithm without making any assumptions on the data, they work as follows: First, the parameters of the classifier are estimated only on the labelled data. Using this trained classifier, either the whole unlabelled subset, $\mathcal{U}$, is labelled (or just the most confident unlabelled examples) so that they can be added to the annotated training dataset, $\mathcal{L}$. Then, the classifier parameters are re-estimated using this new labelled dataset in order to get a new classifier which is used to re-label the subset $\mathcal{U}$. This process is iteratively applied until a stop criterion is met, such as the predicted labels of the unlabelled subset do not suffer any change. Although they show practical success in some applications [23, 75], a trial-and-error methodology with different supervised learning approaches is, in some cases, computationally impracticable.

So, to avoid the time-consuming task of finding competitive semi-supervised classifiers, some methods were proposed on the claim of David MacKay *"You cannot do inference without making assumptions"* [76]. Thus, these methods leverage the unlabelled data by introducing some assumptions linking the features and the values of the class variables [73]. The most common assumptions are the following:

- *Smoothness or continuity assumption:* If two instances are close in a high-density region, they are likely to share the same values for the class variables.
- *Cluster assumption:* The data tend to form a finite number of clusters, and points in the same cluster are more likely to share a label.
- *Low density separation assumption:* The decision boundaries should lie in low-density regions.
- *Manifold assumption:* The data lie approximately on a manifold of a much lower dimension than the input space.

The most popular methods based on assumptions are **transductive support vector machines (TSVM)** and **graph-based methods** [10]. While the former directly implements the low-density separation assumption, the latter approach is directly built on the manifold assumption. Although it is claimed that this approach ensures an improved performance when the assumptions are met [77], these assumptions are, in practise, hard to check when the labelled training subset is limited. If these assumptions are not met, the use of unlabelled data cannot guarantee any significant advantages over learning a purely supervised learning problem [78]. Thus, the trial-and-error property of the previous paradigm returns.

Current research [73, 74] attempts to guard against the possibility of deterioration in performance by not introducing additional assumptions. This paradigm, called **robust or safe semi-supervised learning**, is still an on-going research topic and it could revolutionise the semi-supervised learning field in the near future.

Lastly, since, in Chapter 4, I propose a new methodology to solve a real-world semi-supervised learning problem [23], I would like to add a few lines on my proposal. At the time I faced that problem, no other work on the literature had dealt with a multi-dimensional classification problem in a semi-supervised manner. Moreover, to the best of my knowledge, no safe semi-supervised learning paradigm had been proposed. Thus, I use the natural evolution of the field in the uni-dimensional framework by proposing an EM algorithm based approach with its trial-and-error best classifier search disadvantage.

1.3 Evaluation of Classifiers

Classifiers often produce misclassifications. Thus, once a classifier is proposed, its associated discerning skill needs to be measured. When the generative function is assumed to be known, a common tool used for visualising the performance of a classifier is the true confusion matrix of a given classifier, Ψ [79]. It is a square matrix of size K containing the mathematical expectations of classifying, with a classifier Ψ, an example of the configuration of classes $\mathbf{c}_i$ (rows) as the values $\mathbf{c}_j$ (columns).

As previously mentioned, when the generative model is known, decision rules using the generative model can be defined. In such situations, the *true confusion matrix*, $\mathbf{A}_\Psi = [a_{i,j}]_{1\leq i,j\leq K}$, can be directly calculated by taking expectations:

$$a_{i,j} = \mathbb{E}_{\mathbf{x}|\mathbf{c}=\mathbf{c}_i}[\hat{\mathbf{c}}_\Psi = \mathbf{c}_j]. \tag{1.16}$$

There, $\mathbb{E}[\cdot]$ stands for the mathematical expectation. For instance, each element of the true confusion matrix for the BDR applied to a multi-dimensional problem with generative model $\rho(\mathbf{x}, \mathbf{c})$ is calculated as

$$a_{i,j} = p(\mathbf{c}_i) \int\limits_{\Omega_j} \rho(\mathbf{x}|\mathbf{c}_i) d\mathbf{x}. \tag{1.17}$$

When $\rho(\mathbf{x}, \mathbf{c})$ is unknown, an estimation of the true confusion matrix is utilised instead. It is usually referred to as the empirical confusion matrix, or simply as the confusion matrix. To perform such an estimation, labelled data is required. They usually come in the form of an independent testing dataset, $\mathcal{D}_T$. Similarly to the training dataset, $\mathcal{D}_T$ is also assumed to be sampled from the generative model. Note that both subsets ($\mathcal{D}$ and $\mathcal{D}_T$) must be independent sets in order to ensure a fair performance assessment. When data is limited, there are supervised and semi-supervised methods[3] in the literature, such as

[3] Unsupervised learning is left out in this section due to the fact that I assume the availability of labelled data to evaluate the trained models to use the aforementioned assessment methods.

(repeated) k-fold cross-validation, leave-one-out, etc [80], in order to fairly and efficiently exploit all the available labelled data to both train and test the classifier.

Concerning the estimation of the confusion matrix for a learning algorithm in a dataset, it is performed in the following manner: The classifier Ψ, which has previously been trained using $\mathcal{D}$, is used to classify all the labelled testing data $\mathcal{D}_T$. Afterwards, each element $a_{i,j}$ of the confusion matrix is filled by just assigning the number of instances with real values $\mathbf{c}_i$ which are classified as values $\mathbf{c}_i$ when the classifier Ψ is used. Formally and being $\mathbb{1}(E)$ the indicator function, the elements are calculated as:

$$a_{i,j} = \sum_{(\mathbf{x},\mathbf{c_i}) \in \mathcal{D}_T} \mathbb{1}(\hat{\mathbf{c}}_\Psi = \mathbf{c}_j). \tag{1.18}$$

As stated in [79], the confusion matrix is one of the most informative performance summaries that a learning system can rely on. Among other information, it contains how accurate the classifier is on each configuration of classes, and the way it tends to get confused among configurations. However, it is often tedious not only when determining the overall behaviour of a classifier from the confusion matrix, but also when comparing several classifiers. Therefore, quantity measures which summarise the confusion matrix are often preferred. These measures are known as numerical performance scores [80]. Since *the best behaviour* may vary from one kind of problem to another, there are many different and diverse performance scores in the community. This diversity may, at times, obscure important information on the hypotheses or algorithms under consideration [81]. Thus, it is fundamental to check in advance the adequateness of a numerical performance score for assessing a determined classification problem so that the validity of the obtained results can be ensured.

Moreover, in the literature, there are also other methods such as the graphical performance scores [82] which, instead, produce a visual inspection output. This output is capable of capturing the whole multi-objective nature of the classifier and producing a richer, but more subjective – many simultaneous behaviours must be taken into account by the practitioner –, assessment [80]. Unfortunately, as can easily be seen, these scores also possess the main drawbacks of the confusion matrix. Thus, they are also usually summarised into a single value in order to make a better comparison between classifiers.

1.3.1 Performance Assessment in the Uni-dimensional Classification Setting

In the uni-dimensional classification problem framework, both numerical and graphical performance scores have been broadly used. The **numerical performance scores**, see Table 1.1, can be mainly divided into two different groups: local scores, which only focus on the behaviour of one target class

	Name	Notation	Formula
Local scores	Precision	$\mathcal{P}^i$	$a_{i,i}\left(\sum_{j=1}^{K} a_{j,i}\right)^{-1}$
	Recall	$\mathcal{R}^i$	$a_{i,i}\left(\sum_{j=1}^{K} a_{i,j}\right)^{-1}$
	F-score	$\mathcal{F}_{\beta}^{i}$	$\frac{(\beta^2+1)\mathcal{R}^i\mathcal{P}^i}{\beta^2\mathcal{R}^i+\mathcal{P}^i}$
Global scores	Classification accuracy	$\mathcal{A}cc$	$\sum_{i=1}^{K} a_{i,i}\left(\sum_{i=1}^{K}\sum_{j=1}^{K} a_{i,j}\right)^{-1}$
	Arithmetic mean among the recalls (*a-mean*)	$\mathcal{A}$	$\sum_{i=1}^{K}\frac{1}{K}\mathcal{R}^i$
	Geometric mean among the recalls (*g-mean*)	$\mathcal{G}$	$\sqrt[K]{\prod_{i=1}^{K}\mathcal{R}^i}$
	Harmonic mean among the recalls (*h-mean*)	$\mathcal{H}$	$K\left(\sum_{i=1}^{K}\frac{1}{\mathcal{R}^i}\right)^{-1}$
	Maximum value among the recalls	max	$\max_i \mathcal{R}^i$
	Minimum value among the recalls	min	$\min_i \mathcal{R}^i$

Table 1.1: Numerical performance scores for the uni-dimensional framework (by convention $0/0 = 1$).

value, and global scores, which summarise the performance of the classifier taking into account its behaviour in all the values for a unique class variable.

The most used *local performance scores* are precision $\mathcal{P}^i$ and recall $\mathcal{R}^i$. Whilst $\mathcal{P}^i$ assesses to what extent the predictions of a certain class c_i are correct, $\mathcal{R}^i$ assesses to what extent all examples of a certain class c_i are classified as so. Introduced in the information retrieval field [83], another widely used local score is the $\mathcal{F}$-score [84]. This score weights the two previously defined local performances for a given class value and depends on a parameter, β, provided by the practitioner. When $\beta = 1$, the obtained $\mathcal{F}$-score is known as the $\mathcal{F}_1$-score and corresponds to the harmonic mean between the precision and recall terms. This setting of the score is usually the one used in the literature [85, 86]. Unfortunately, for their local property, these scores lose the global picture of the performance of the classifier; they are more useful when applied to most, or even all, of the values of the class variable [81]. Therefore, most of the global scores are just functions which, by taking local performance scores applied to some/all classes, summarise the behaviour of the classifier according to a determined subjective criterion.

The most ubiquitous *global performance scores* are the classification accuracy, $\mathcal{A}cc$ and the classification error ϵ, which are equivalent: $\mathcal{A}cc = 1 - \epsilon$. Whilst the former gives the ratio of the number of correct classifications over the total number of cases in the testing dataset, the latter focuses on the misclassifications. Note that $\mathcal{A}cc$ can also be seen as a weighted arithmetic mean over the K recalls. In this case, each recall $\mathcal{R}^i$ is weighted on the number of examples of the class value c_i in the testing dataset. Although they are a proper choice for maximising the overall number of correct classifications, the performance on underrepresented classes has very little impact on the global measure when compared to the overrepresented classes [81][86]. In those cases, their use is pointless and the use of unweighted averages over the recalls is preferred [30]. With this approach, all classes, over and underrepresented, share a common consideration in the global score. The most-used scores in the unbalanced scenario are the Pythagorean means – arithmetic ($\mathcal{A}$), geometric ($\mathcal{G}$), and harmonic ($\mathcal{H}$) means – over the recalls of the K-classes. In the literature [80, 87, 88], they are referred to as a-mean, g-mean, and h-mean, respectively. Other works, e.g. [30], also consider the maximum recall, max, and the minimum recall, min (among the classes), as global scores.

In this chapter, which focuses on classification, the class variables are assumed to be categorical random variables, i.e. they have no logical order. When this assumption is relaxed and the target variables are nominal or ordinal values, then other numerical performance scores arise. In Chapter 4, several numerical performance scores for ordinal class variables are introduced and used due to necessities of the real-world problem confronted.

Regarding the **graphical performance scores**, a well-known score for binary classification problems is the Receiver Operating Curve ($\mathcal{ROC}$) [82]. It allows the visualisation of the trade-off between the proportion of positives, c_1, that are correctly identified and the proportion of negative events, c_2, wrongly categorised as positive, proving that any classifier cannot increase the number of true positives without also increasing the false positives [29, 89]. The area under the $\mathcal{ROC}$ curve ($\mathcal{AUC}$) provides a single measure (numerical score) of a classifier's performance [90] – see Figure 1.8 – which properly fits for the binary setting. Multi-class problems, instead, introduce the issue of combining multiple pairwise discriminability values for which there are several approaches in the literature [91, 92].

1.3.2 Evaluation of Multi-dimensional Classification Approaches

By means of the definition of the confusion matrix used in eq. (1.16) and eq. (1.17), the use of either uni-dimensional numerical performance score of Table 1.1 to assess multi-dimensional classifiers seems straightforward. Unfortunately, in the published literature, only the classification accuracy [22, 23, 58, 93] – out of the 6 global scores – presented is utilised. In those works, it is called **classification joint accuracy**. An interesting potential future research line, which I in-depth discuss in the future work section (Section

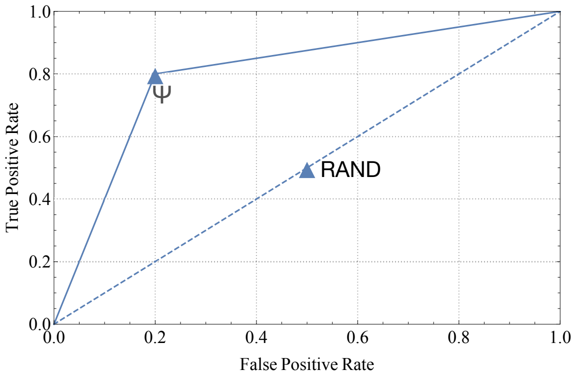


Fig. 1.8: Example of a ROC plot. Two curves for two classifiers are plotted: the dashed line represents the random classifier, RAND, whilst the solid line is a classifier, Ψ, behaving better than the random classifier.

3.1), could be the study of how different uni-dimensional numerical performance scores describe the behaviour of a multi-dimensional classifier.

Given the shape of the confusion matrix in the uni-dimensional case for a single class variable, a totally different approach to assess the classifier is to extract M independent confusion matrices, one for each class variable. Then, by summarising each confusion matrix with a numerical performance score and averaging the results, a global performance evaluation of the multi-dimensional classifier can be extracted. This methodology has also been used in the literature [23], going by the name of **classification mean accuracy**. It obtains the classification accuracy of the classifier for each class variable and, then, calculates the arithmetic mean of all the accuracies.

Finally, as put forward in the previous subsection, the categorical property of the class variables can also be relaxed in this setting. In Chapter 4, due to the requirements of the classification problem being dealt with, I also define, describe and make use of a numerical multi-dimensional global performance score for ordinal class variables.

1.4 Threats when Obtaining Acceptable Classifiers

As stated in [94]: "A training supervised or semi-supervised dataset $\mathcal{D}$ differs from a random labelling of the classes in the difficulty of training a competitive classifier capable of assigning values for the class variables to future data from the same generative model. A random labelled dataset is improbable to be the source of a classifier since not much can be learned from it about the unseen points. Similarly, in real-world classification problems such learning can usually be done with various degrees of difficulty which may produce weak solutions unable to successfully predict future instances."

A classification problem can be difficult for different reasons. Thus, in this dissertation, I refer to the common situations which degrade the discerning skill of the trained classifiers as threats to obtain acceptable classifiers. These, which have been long studied in the literature [89, 95], are mainly the following: (i) training set size [96], (ii) noise in the data [97], (iii) dispersity of some classes and the presence of small disjuncts [98], degrees of overlapping among the classes [99], and degree of unbalance among the values of the class variables [100].

The threats, which have strong interdependences among them [89], are briefly described in the following subsections. Unfortunately, since complex classification settings lead to intricate casuistry which may hinder the description, when approach each threat, I will refer only to (uni-dimensional) binary classification problems for the sake of clarity.

1.4.1 Training Set Size

In practise, the number of labelled examples to train a classifier is frequently limited because labelling is either costly or time-consuming. As the estimation of the parameters of the classifiers is dependent on the labelled examples, in supervised learning, the scarcity of data directly translates into poor estimations and systems which cannot accurately classify future instances; the induction algorithms do not have enough data to make a consistent generalisation [33] about the inner distribution of the samples. Potential solutions to this problem may be (i) gathering huge unlabelled datasets so that semi-supervised learning techniques [10] may be applied, or (ii) obtaining more labelled data to train the classifier but only the crucial instances. More information about the former approach can be found in Section 1.2.3. Regarding the latter, there are estimations of the required number of examples to obtain valuable estimations for certain well-known probability distributions [96] and estimations of the required class distribution [101] for certain supervised learning paradigms.

1.4.2 Presence of Noise in the Data

The data which is corrupted or distorted is called noise. This property is known to affect the way any data mining system behaves [102]. Fortunately, both the inductive bias [33] of the learning algorithm and the class frequencies of the training dataset play important roles enabling the proposal of mitigating approaches.

According to [33], the inductive or learning bias of a learning algorithm is the set of assumptions that the learner uses to predict unseen examples which are not encountered in the training dataset. This bias prevents overfitting – classifiers whose models are too well trained on the training data but fail to make predictions on new data – and fosters generalisation. In the event of having all the class values appropriately represented in the training dataset common learning biases are currently able to handle noisy data. When facing high levels of noise, [89] claims that the most robust classifiers are Bayesian classifiers and SVM. In these cases, rule-based learning approaches are not recommended as their performance degrades quicker in the degree of noisy examples.

The most problematic issue, here, happens when noise is present for the underrepresented class values, in areas of low density or in small training datasets. As long as there are more underrepresented data than noise, approaches such as undersampling techniques [103] may be applied to alleviate the problem [104]. Alternatively, the learning bias must be adjusted to enable the inclusion of both real data and noise in the learning process or either let the inductive bias eliminate both noisy and true real data.

Finally, note that the noise can also be filtered or even corrected if it is introduced in the training dataset by an external source. This last approach is the most recommended to overcome this problematic issue.

1.4.3 Presence of Disperse Small Disjuncts

Small disjuncts are really small clusters of data in the training dataset which may be mistaken as noise in the learning process [105]. Thus, they can be eliminated by the inductive bias of the learning algorithm. When a class value is composed of multiple and disperse small disjuncts, and these are removed in the learning process, the overall performance will show a significant drop.

To deal with this complex case, we can consider small disjuncts as "necessary noise" and make use of the same approaches of Section 1.4.2 but reversely [105]. Thus, the classification problem can be approached by using sensitive to noise solutions such as rule-induction techniques. Also, the induction bias of the learning algorithm can be modified to overfit on small clusters. Finally, and among other approaches, the most effective approach is to obtain additional training data [89].

1.4.4 Class-overlapping

The previous threats are only dependent on the training dataset, so any improvement on data might alleviate the problem. However there are threats which only depends on the nature of the classification problem being dealt with, i.e. they are dependent on the generative function $\rho(\mathbf{x}, \mathbf{c})$. The following last two threats lie within this setting. First, I focus on class-overlapping, a problem which originates in the local probability distribution of the feature space $\rho(\mathbf{x}|c)$ [95], and it is regarded as one of the toughest pervasive problems in classification [106].

Class overlap arises when objects belonging to different classes share the same feature representation, i.e. $\exists \mathbf{x}$ s.t. $\rho(\mathbf{x}|c_1)\rho(\mathbf{x}|c_2) > 0$, and it has a direct relationship with the irreductible Bayes error – eq. 1.4 –. The value of e_B depends on the proportion of size of the feature space which is overlapped and

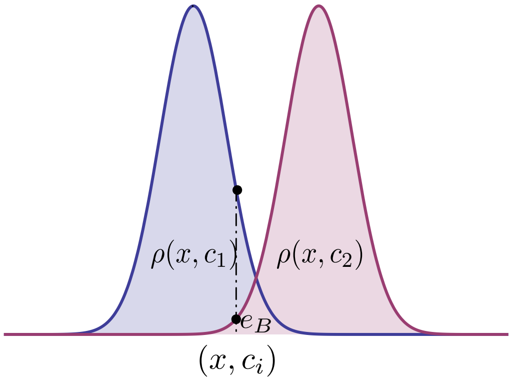


Fig. 1.9: The Gaussian conditional features distributions $\rho(\mathbf{x}|c)$ for a uni-dimensional binary classification problem showing class-overlapping.

it is equal to 0 if and only if the classifier can be modelled using a deterministic algorithm. When the uncertainty of the generative model increases ($e_B > 0$), the achievable performance improvement of a given trained classifier, Ψ, over a baseline classifier, such as RAND, gets narrower. Thus, the classification problem becomes complex.

Figure 1.9 shows a uni-dimensional problem showing class-overlapping. As it can be seen, there is ambiguity about the real class value of any example originated in the overlapped region. Fortunately, class-overlapping has only a real dramatic effect when the overlapped regions are large and contain similar number of training data for each class. In other words, both conditional feature distributions are similar, $\rho(\mathbf{x}|c_1) \sim \rho(\mathbf{x}|c_2)$, thus making the task of differentiation between c_1 and c_2 very difficult or, even, impossible. In the unlikely scenario of having all the feature space overlapped and both classes sharing a similar class probability, any classifier will show a similar performance to RAND.

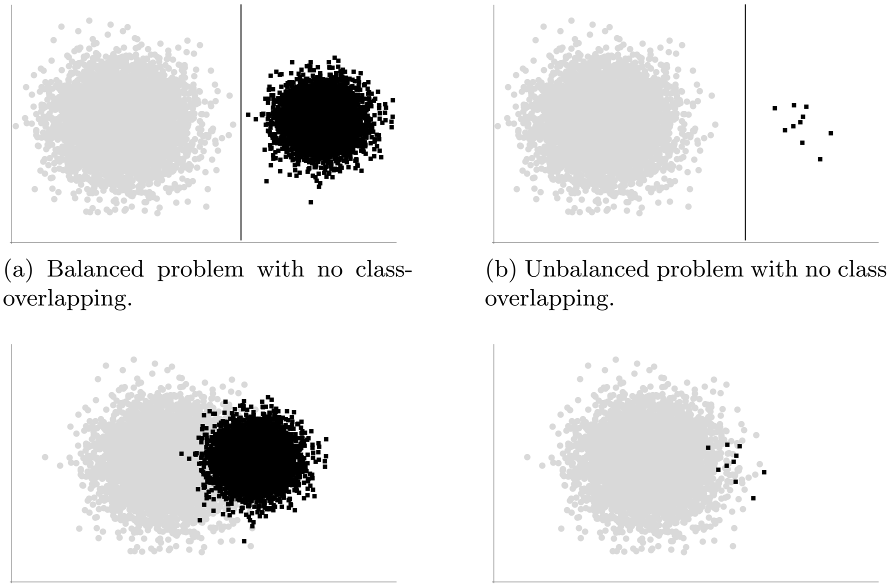
There are some methodological proposals to alleviate this problem [89]. Unfortunately, since it is an intrinsic property of the dealt classification problems, there is little room to overcome this threat. In this case, neither a larger dataset is able to overcome the degradation. Finally, in practical applications, exploring alternative class definitions, i.e. using different feature sets for the objects to be classified, could be an approach to solve this complexity [107].

1.4.5 Class-imbalance

The class-imbalance problem is also dependent on the generative function but, unlike class-overlapping, it arises from the class probability distribution, $p(c)$, and it is able to compromise the performance of the vast majority of standard learning algorithms [11]. Traditional learning algorithms assume a 0-1 or a surrogate loss function and, thus, they expect balanced class distribution – $\forall i, p(c_i) \sim 1/K$ – or equal misclassification costs [28]. In other words, they inherently maximise classification accuracy and have an asymptotic behaviour close to the BDR favouring an overall number of correct classifications (keep in mind the example of Figure 1.2). Therefore, when they are presented with complex training datasets sampled from skewed class distributions – $\exists i$ s.t. $p(c_i) \sim 0$ –, these algorithms fail to properly represent the distributive characteristics of the data and resultantly provide dummy classifiers, always classifying new data as the most probable value. Here, the class values with the lowest probabilities are ignored. Furthermore, it also is worth pointing out that the latter class values are commonly the ones having the highest interest from a learning point of view. This implies a great classification cost when they are not correctly predicted [89].

Whether a skewed class distribution produces a class-imbalance problem depends on the previously described class-overlapping issue. Figure 1.10 shows their relationship by plotting four binary classification scenarios sharing two different degrees of class-imbalance and two different degrees of class-overlapping. For each problem, the data has been created by sampling two bivariate Gaussian distributions. To simulate no class-overlapping, I choose two Gaussians whose means are far from each other (Figures 1.10a and 1.10b), and to simulate the opposite, I shorten the distance between these means (Figures 1.10c and 1.10d). Regarding the class-imbalance setting, it is simulated by sampling $1,000$ instances of each class for balanced problems (Figures 1.10a and 1.10c), and by sampling $1,000$ instances for the majority class and only 10 samples for the minority, for the case of unbalanced problems (Figures 1.10b and 1.10d). By just having an overall look at Figure 1.10, it can be easily noticed how the combination of both factors (class-imbalance and class-overlapping) has a straight effect on discriminating among the classes and, therefore, on the performance of the trained classifier. When there is no class-overlapping ($e_B = 0$), as shown in the first row of the figures, the class distribution does not hinder the predictive power of the resulting classifier. In both scenarios, a simple and perfect discriminant linear classifier can be easily drawn. This classifier is represented by a straight continuous line in the figure. However, in the second row, the situation is completely different. When both classes are balanced (Figure 1.10c), a classifier with a tolerable recall for both classes can be learned. Unfortunately, in the unbalanced scenario (Figure 1.10d), the lack of enough examples for the minority class hinders the process of discriminating it; any intuitively chosen classifier will be incompetent, i.e. it will have a low recall for the minority class. This example concurs with the literature [99], where it is stated that (i) only when the class-overlapping is non-zero, the influence of the class distribution on the competitiveness of the inferred classifier is noticeable, and that (ii) the influence of class-overlapping into the learning process is even stronger than class-imbalance.

In an attempt to overcome this threat, hundreds of methodological solutions to the class-imbalance problem are currently proposed. They usually



(a) Balanced problem with no class-overlapping.

(b) Unbalanced problem with no class-overlapping.

(c) Balanced problem with class-overlapping.

(d) Unbalanced problem with class-overlapping.

Fig. 1.10: Relation between class-overlapping and class-imbalance.

attempt to find an adequate balance between the prediction powers of the trained classifier for the most and the least probable class values. According to [103], the proposed solutions can be mainly categorised into the following three major groups:

1. The development of *inbuilt mechanisms* [108], which change the classifications strategies of the classifier to impose bias towards the minority classes.
2. The usage of *data sampling methods* [109], which modify the estimated class distribution (from the training dataset) in the learning algorithm so that it can learn from a balance scenario.
3. The adoption of *cost-sensitive learning methods* [110], which assume higher misclassifications costs for examples of the minority classes.

To the best of my knowledge, all literature is focused on unbalanced unidimensional classification problems. However, within that setting, both the multi-class framework [111] and the theoretical understandings of this intricate issue [99, 101, 107] are commonly left aside. Few works on the class-imbalance literature can be found dealing with those cases. Here, it is worth mentioning that, when the binary class constraint of a single target variable is relaxed, the class-imbalance casuistry gets more complex, and, thus, new theoretical and methodological challenges that are not observed in binary problems arise [111]. Whilst in an unbalanced binary problem there is only one majority class value – the most probable – and one minority class values – the least probable –; in the multi-class framework, an unbalanced class distribution may show multiple majority class values or multiple minority class values. In this thesis, based on the class distribution, I formally catalogue the multi-class classification problems into the following groups:

Definition 6. *A K-class classification problem, γ, is balanced if it exhibits a uniform distribution between its classes. Otherwise, it is considered to be unbalanced. Formally,*

$$\gamma \text{ is balanced } \iff \forall i, p(c_i) = \frac{1}{K}, \text{ i.e. } p(c) = \mathbf{e}, \tag{1.19}$$

where $\mathbf{e}$ is the special case of having K equiprobable class values.

Definition 7. *A multi-class classification problem ($K > 2$) shows a multi-majority class-imbalance if most of the class values have a higher or equal probability than equiprobability, i.e.*

$$\gamma \text{ is multi-majority } \iff \sum_{i=1}^{K} \mathbb{1}\left(p(c_i) \geq \frac{1}{K}\right) \geq \frac{K}{2}. \tag{1.20}$$

Definition 8. *An unbalanced classification problem, γ, with $K > 2$, shows a multi-minority class-imbalance when most of the class probabilities are below the equiprobability. Formally,*

$$\gamma \text{ is multi-minority } \iff \sum_{i=1}^{K} \mathbb{1}\left(p(c_i) < \frac{1}{K}\right) > \frac{K}{2}. \tag{1.21}$$

In this dissertation, assuming the whole uni-dimensional spectrum (binary + multi-class), I pursue a theoretical understanding and formalisation of the class-imbalance problem and I also propose a methodological advance. Whilst, in Chapter 6, I shed some light, through a novel theoretical framework, on the adequateness of the most commonly used performance scores to assess unbalanced multi-class problems as well as on the scores maximised in the most popular methodological approaches; in Chapter 7, a class-imbalance metric capable of suitably characterising the imbalance degree of a given multi-class problem is proposed.

2

Main Contributions to Classification in a Nutshell

This dissertation explores two different and intricate scenarios within the classification field: the *semi-supervised learning framework* and *the class-imbalance problem*. Since each scenario has been explored from both theoretical and methodological perspectives, four contributions, sorted in chronological order, are summarised in this chapter:

- **two theoretical studies** exploring both the semi-supervised learning (Section 2.2, which digests Chapter 5) and the class-imbalance (Section 2.3, summarising Chapter 6) scenarios, and
- **two different methodological advancements** to the problem-solving process. Section 2.1 – summary of Chapter 4 – corresponds to the semi-supervised learning related contribution and Section 2.4 sums up the advances proposed for the class-imbalance problem of Chapter 7.

2.1 Semi-supervised Learning of Multi-dimensional Classifiers

Encouraged by the requirements of a real-world problem in the context of Sentiment Analysis [112, 113, 114, 115], the first contributions enriched the state-of-the-art literature with not only **a methodological extension of the semi-supervised learning framework to the whole multi-dimensional classification spectrum**, but also **a new supervised multi-dimensional Bayesian network classifier learning algorithm**.

Sentiment Analysis is a broad area defined as the computational study of opinion, sentiments and emotions expressed in text [112]. It mainly originated to meet the need for organisations to automatically find the opinions or sentiments of the general public about their products and services, as expressed on the Internet. Within this domain, Socialware©, one of the most relevant companies in mobilised opinion analysis in Europe, has been working on an application slightly different to the traditional approaches. In this case,

Class-variable	Cardinality	Values
Sentiment Polarity	5	{very negative, negative, neutral, positive, very positive}
Subjectivity	2	{objective,subjective}
Will to Influence	4	{none, question, complaint/recommendation, appeal}

Table 2.1: Target class variables of the faced multi-dimensional problem.

	Feature	Description
1	**First Persons**	Verbs in the first person.
2	**Second Persons**	Verbs in the second person.
3	**Third Persons**	Verbs in the third person.
4	**Relational Forms**	Phatic expressions, i.e. expressions performing a social task.
5	**Agreement Expressions**	Expressions that show agreement or disagreement.
6	**Request**	Sentences that express a certain degree of request.
7	**Imperatives**	Imperative verbs in the second person.
8	**Exhorts and Advice**	Exhortative verbs, e.g. recommend, advise, prevent, etc.
9	**Sufficiency Expressions**	Expressions used to corroborate other sentences of the text.
10	**Prediction Verbs**	Verbs in the future.
11	**Authority**	Expressions that denote high degree of authority.
12	**Questions**	Direct and indirect questions.
13	**Positive Adjectives**	Positive adjectives.
14	**Negative Adjectives**	Negative adjectives.

Table 2.2: Features of the faced multi-dimensional problem.

instead of studying a certain aspect[1] of the text in isolation, the company sought to characterise the costumers of a set of Spanish companies through three different, but related, dimensions: their *sentiment polarity* towards the product, the *subjectivity* of their online posts, and their *will to influence* on other customers through their posts in several forums (see Table 2.1). As can be noticed, I was dealing with a multi-dimensional classification problem.

Then, each post had to be represented as a set of features. Here, I relied on the high levels of experience of the company on manually dealing with this problem and, after properly testing them against the state-of-the-art alternatives, I opted to use their 14 morphological features (Table 2.2).

In the end, the provided dataset consisted of $2,542$ posts written in Spanish. Since labelling is a laborious and time-consuming task, only 150 pieces of text were manually labelled by an expert in Socialware© according to the three different class variables exposed. The remaining $2,392$ posts were left as unlabelled instances. Therefore, this multi-dimensional classification problem had to be approached using semi-supervised learning techniques.

At the time I faced the problem, no other work on the literature had dealt with a multi-dimensional classification problem in a semi-supervised manner. Thus, I made use of the previous natural evolution of the semi-supervised field in the uni-dimensional framework by proposing an EM algorithm based approach to deal with multiple, and possibly related, class variables. This approach has the main advantage that it can be directly applied to any supervised learning algorithm. However, it is computationally costly. On that

[1] E.g. *subjectivity classification* [116], *sentiment polarity classification* [113], *authorship identification* [117], or *affect analysis* [118].

account, my first objective was to study the feasibility of using the available learning algorithms [57] for the sub-families of multi-dimensional class network classifiers in this context.

The real costly task in learning Bayesian networks from a dataset is the structural learning process. Estimating the parameters for a given fixed network is, instead, quite straightforward. So, I focused on analysing the former and, after a review of the multi-dimensional family of Bayesian network classifiers (MDnB, MDTAN, and MD J/K), the following conclusions were found:

1. As the MDnB has a fixed structure, here, it raised no computational problem.
2. The proposed learning algorithm for MDTAN [57] follows a computationally intensive wrapper method for both structural and parameter learning tasks by maximising the joint accuracy of a given dataset. Fortunately, the limited sizes of the dataset in question and multi-dimensional problem allowed the use of this early learning algorithm.
3. The structure MD J/K has been already defined in the literature [58]. However, to the best of my knowledge, it possessed no associated learning algorithm.

Because of this, a supervised filter learning algorithm for multi-dimensional J/K dependences Bayesian network classifiers which uses statistical testing over mutual information measures were proposed. Its 'filter' nature had a favourable impact in reducing the computation time of the proposals. Another advantage of the suggested algorithm is that, by setting both J and K to 1, a MDTAN structure can be learnt in a shorter time than using the original algorithm.

Algorithm 2 My version of the EM Algorithm [119]

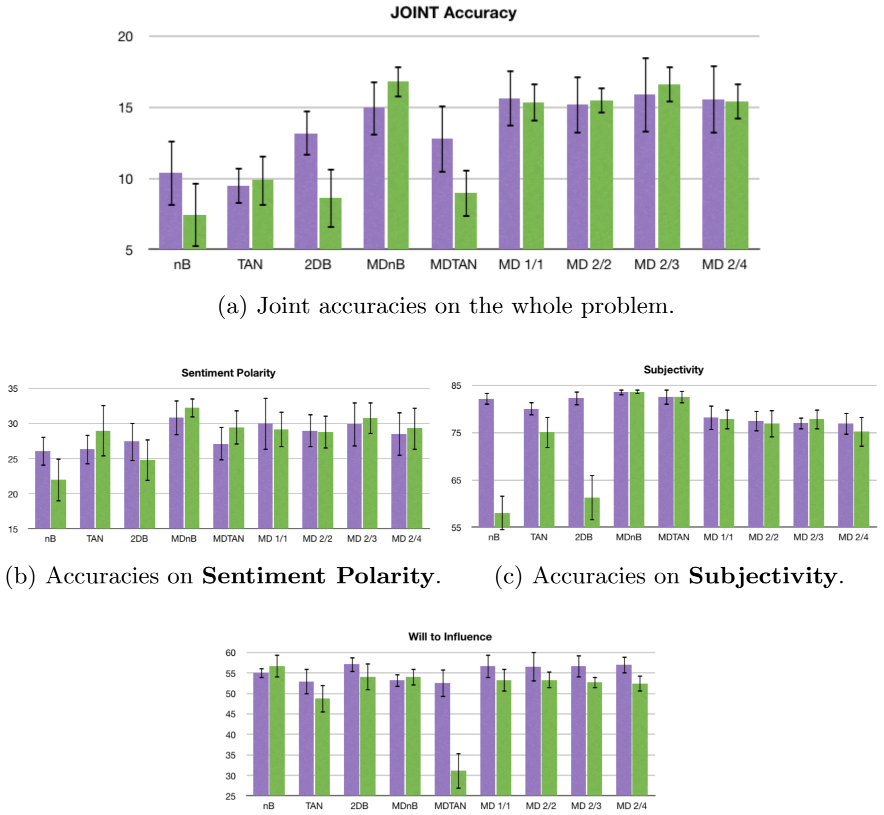
```
Require: A training dataset $\mathcal{D}$ with both labelled and unlabelled data and
    an initial model $p_{\mathbb{B}}^{(K=0)}(\mathbf{x}, \mathbf{c})$ with a fixed structure and with an initial set
    of parameters $\mathbf{\Theta}^{(K=0)}$.
while the model $p_{\mathbb{B}}^{(K)}(\mathbf{x}, \mathbf{c})$ does not converge do
    E-STEP Use the current model $p_{\mathbb{B}}^{(K)}(\mathbf{x}, \mathbf{c})$ to estimate the probability
        of each configuration of class variables for each unlabelled instance.
    M-STEP Learn a new model $p_{\mathbb{B}}^{(K+1)}(\mathbf{x}, \mathbf{c})$ with structure and parameters
        $\mathbf{\Theta}^{(K)}$, given the estimated probabilities in the E-STEP.
end while
Ensure: A Bayesian network classifier $\psi_{\mathbb{B}}$, which takes an unlabelled instance
    and predicts the class variables.
```

Afterwards, I proceeded with the extension of the previous family of learning algorithms to the semi-supervised learning framework and the eventual solution of the faced classification problem. The typical aim of the EM algorithm in the semi-supervised framework is to find the parameters of a given structure for the model $p_{\mathbb{B}}(\mathbf{x}, \mathbf{c})$ that maximises the likelihood of the data, using both labelled and unlabelled instances. However, in [72], it is stated that if the correct structure of the real distribution of the data is obtained, unlabelled data improve the performance of the trained Bayesian network classifier; otherwise, unlabelled data can actually degrade performance. For this reason, it seemed more appropriate to perform both structural and parametric searches in order to find the real generative model. The proposed version of the EM algorithm for multi-dimensional classification can be found in Algorithm 2. Note that this proposal is closer to the Bayesian structural EM algorithm [120] rather than the original formulation [119]. At that point, I tested my battery of proposed semi-supervised learning algorithms (EM plus the family of Bayesian network classifiers) over a set of designed artificial datasets. The experimentation, which can be found in the appendices of Chapter 4, concludes by claiming that more accurate multi-dimensional classifiers can be found using these semi-supervised learning approaches.

Finally, in order to solve the real world problem being addressed using the proposed methodologies, an exhaustive experimentation where two major hypotheses were checked, was performed:

1. The uni-dimensional counterparts of the studied multi-dimensional Bayesian network classifiers yield suboptimal solutions to this problem, i.e. the explicit use of the relationships among the class variables can be beneficial to improve the recognition rates.
2. When there is a scarcity of labelled data, multi-dimensional techniques can work effectively with huge sets of unlabelled data in order to improve the classification rates.

Then, the given semi-supervised dataset was used to learn three different (one for each class variable) uni-dimensional Bayesian network classifiers – naïve Bayes (nB), tree-augmented network classifier (TAN), and a 2-dependence Bayesian classifier (2DB) – and three sub-families of the multi-dimensional Bayesian network classifiers – MDnB, MDTAN [57], and MD2/K with $K = 2, 3$, and 4 –. In order to compare both learning frameworks, all the structures were learnt in both scenarios. As stated in the introduction, the uni-dimensional approaches cannot be straightforwardly applied to deal with multi-dimensional problems, so, in these cases, the dataset was divided into three uni-dimensional independent datasets. Figure 2.1 shows the obtained results; whilst Figure 2.1a sums up the joint accuracy of each learning algorithm, Figures 2.1b, 2.1c, and 2.1d show the accuracies obtained in each class variable. The results confirmed both hypotheses and depicted the semi-supervised version of the MDnB as the global best solution for the given real-world problem. Probably due to the fact that, when applying rational criteria in determining the predictive features of a problem (i.e. the expert list of



(a) Joint accuracies on the whole problem.

(b) Accuracies on **Sentiment Polarity**.

(c) Accuracies on **Subjectivity**.

(d) Accuracies on **Will to Influence**.

Fig. 2.1: Estimated performance scores on the faced problems using both uni-dimensional (nB, TAN, 2DB) and multi-dimensional classifiers (MDnB, MDTAN, MD 1/1, MD 2/2, MD 2/3, MD 2/4) in both supervised (purple) and semi-supervised learning (green) scenarios (20 × 5cv)

14 features), the resulting features are usually probabilistically independent given the class variables [121].

The solution of this Sentiment Analysis application throws interesting methodological future work paths. Among them, ways to optimise the semi-supervised learning algorithms to avoid computational burden and research towards robust semi-supervised learning algorithms [73, 74] for multi-dimensional classifiers are key. Regarding the text classification domain, two interesting research paths remain open:

1. Affect Analysis [118] aims at extracting a large number of potential emotions from text [122], e.g. happiness, sadness, anger, hate, violence, ex-

citement, fear, etc. Some of them are not mutually exclusive and certain emotions may be correlated. Thus, multi-dimensional classification solutions, both in supervised and semi-supervised frameworks, seem to be a perfect match for this problem.

2. Here, I have proven that the same corpus can be used to identify several target variables in the same classification task. Therefore, why not use the proposed methodology to take advantage of the possible existing relationships between the class variables to improve the recognition rates in the characterisation of textual information?

2.2 A Study on Semi-supervised Learning of Multi-class Classifiers

Over the course of this research project, I noticed that the majority of the theoretical works on semi-supervised learning (see Table I of [36]) solely focused on standard binary problems. Furthermore, most of them do not straightforwardly generalise to the multi-class setting. Thus, in an attempt to overcome this limitation, here, in order to highlight the blocking issues to generalise their framework, I particularly focus on the conclusions of the key research problem proposed in [31, 66], where the fundamental limits for the probability of error of binary semi-supervised problems are drawn. Afterwards, I was able to propose new general strategies to adapt their research by, specifically, **contributing with an optimal semi-supervised procedure for the whole uni-dimensional problem and using it to study the convergence of its optimal probability of error when multi-class labelled data are scarce.** By doing so, **fundamental limits in performance of any uni-dimensional classifier learnt in the semi-supervised scenario** can be established.

Before studying this key scenario, I deal with the case of dealing with no labelled data at all. Under the assumption of having a mixture density [123] as generative model, even with an infinite amount of unlabelled data to estimate the correct model density, $\rho(\mathbf{x})$, the generative model is only identifiable up to a permutation, π, of its single-components. In other words, I am able to determine the whole distribution of class values but there is not a single clue to the real mapping between them and the true class values of the generative model. Here, I also prove that the probability of committing a classification error, $P_e(l, u)$, where l is the number of labelled data in the training subset and u the number of unlabelled data, when there is no labelled data coincides with that of the RAND classifier:

$$P_e(0, u) = \frac{(K-1)}{K}, \ \forall u \geq 0. \tag{2.1}$$

Subsequently, I centre on the pioneering work of Castelli and Cover [31] for binary problems with l and u greater than 0. Under the assumptions

Algorithm 3 Optimal theoretical procedure for SSL binary problems [31] [66]

```
LEARNING TASK:
    • Stage 1 Use unlabelled set $\mathcal{U}$ to obtain $\rho(\mathbf{x})$ and, by identifiability, a
      permutation $\pi$ of its components ($\rho(\mathbf{x}|c_{\pi(1)})$, $\rho(\mathbf{x}|c_{\pi(2)})$, $p(c_{\pi(1)})$, and
      $p(c_{\pi(2)})$).
    • Stage 2 By means of the likelihood ratio test and the labelled set $\mathcal{L}$,
      determine the correspondence between the real classes and the current
      mixture components:
      $\hat{\pi}(1) = 1$ and $\hat{\pi}(2) = 2$, or $\hat{\pi}(2) = 1$ and $\hat{\pi}(1) = 2$.
CLASSIFICATION TASK:
    • Stage 3 Assign the sample $\mathbf{x}^{(0)}$ to the class induced by the BDR using
      the learned model.
```

of (i) learning the correct model density, $\rho(\mathbf{x})$, using an infinite amount of unlabelled examples, (ii) having the unlabelled samples distributed according to a identifiable [123] mixture density $\rho(\mathbf{x}) = p(c_1)\rho(\mathbf{x}|c_1) + p(c_2)\rho(\mathbf{x}|c_2)$, and (iii) having $p(c_1), \rho(\mathbf{x}|c_1), p(c_2)$ and $\rho(\mathbf{x}|c_2)$ unknown, the authors define the three-stage optimal procedure of Algorithm 3. It consists of two major parts: (1) the learning task with two stages, where a model using both $\mathcal{U}$ – Stage 1 – and $\mathcal{L}$ – Stage 2 –, and (2) the classification task or Stage 3, where the unseen instances are classified according to the previously learnt model. This optimal procedure achieves the highest lower bound of the probability of error of any semi-supervised classifier, since all the three stages are optimal. Unfortunately, this optimal procedure cannot be transferred to the multi-class scenario. Although both stage 1 and 3 can be straightforwardly used to deal with multi-class classification problems, the optimality of Stage 2 can only be guaranteed for $K = 2$ due to the fact that the likelihood ratio test is only optimal for two simple hypotheses [66].

Based on the principle of maximum likelihood to determine the real permutation, π, between the components learnt using the unlabelled records and the real class values, I propose an optimal learning strategy for Stage 2 called PC_{SSL} (Permutation of Components in Semi-Supervised Learning). It returns, in polynomial time on the number of class values K, the correspondence π with the highest likelihood function $L(\pi; \mathcal{L})$. As proved, PC_{SSL} is optimal for the multi-class scenario. Therefore, the optimal theoretical procedure for semi-supervised learning in the multi-class framework which can be used to study the fundamental limits of the proposals in this setting is now as shown in Algorithm 4.

Additionally, since semi-supervised learning is applied to domains where labelled data are very difficult to obtain, the minimum required amount of labelled records to be used in Stage 2 of Algorithm 4 is studied. Whilst just

Algorithm 4 Optimal theoretical procedure for SSL multi-class problems

```
1: LEARNING TASK:
    • Stage 1 Use unlabelled set $\mathcal{U}$ to obtain $f(\mathbf{x})$ and, by identifiability, a
      permutation of its components $(f_{\pi(j)}(\cdot)$, and $\eta_{\pi(j)}, j = 1, ..., K)$.
    • Stage 2 Use the labelled set $\mathcal{L}$ to determine the correspondence between
      the classes and the mixture components, i.e. the permutation of the
      components $\hat{\pi}$, by means of my proposal PC$_{\mathrm{SSL}}$:
      $\hat{\pi} = \arg\max_{\pi} L(\pi; \mathcal{L}) = \arg\max_{\pi} \prod_{i=1}^{K} \prod_{(\mathbf{x},c_i)\in\mathcal{L}} p(c_i)\rho(\mathbf{x}|c_{\pi^{-1}(i)})$
2: CLASSIFICATION TASK:
    • Stage 3 Assign the sample $\mathbf{x}^{(0)}$ to the class induced by the BDR using
      the learned model.
```

one labelled datum is needed in Algorithm 3 for binary problems [31], in the multi-class framework, at least $(K-1)$ examples of different class values are required. Thus, I am required to calculate l_k as the expected minimum number of instances needed to have a labelled set with $(K-1)$ different class values among them. The minimum value for l_k is set for balanced K-class classification problems and it is given by the following formula:

$$l_K = \begin{cases} 1 & \text{if } K = 2 \\ \displaystyle\sum_{j=2}^{K-1} (-1)^j \binom{K}{j} (j-1) \times \left(\frac{K-j}{K}\right)^{(K-2)} \left(K - 1 + \frac{(K-j)}{j}\right) & \text{if } K > 2 \end{cases} \tag{2.2}$$

When the class probabilities differ from equiprobability, l_k exponentially increases in the variance between the probabilities as shown in the published material.

Finally, I focus on how labelled records reduce the probability of committing a classification error when having a huge amount of unlabelled records. Under the assumption of an infinite amount of unlabelled examples, in binary problems, Algorithm 3 is used to prove that labelled records exponentially reduce the probability of error $P_e(l,\infty)$ to the Bayes error e_B [31]. Formally,

$$P_e(l,\infty) - e_B = exp\{-lZ + o(l)\},$$

where $Z = -\log\left\{2\sqrt{p(c_1)p(c_1)} \int \sqrt{\rho(\mathbf{x}|c_1)\rho(\mathbf{x}|c_2)}\right\}$ is the Bhattacharyya distance between the densities $\rho(\mathbf{x}|c_1)$ and $\rho(\mathbf{x}|c_2)$ multiplied by a term equal to $\log(2\sqrt{p(c_1)p(c_2)})$. When the binary constraint is relaxed, the casuistry of when an error is committed in Algorithm 4 becomes more complex. Thus, different scenarios must be created and studied. After some algebra, I reach the conclusion that, when there is no class-overlapping among the class values

($e_B = 0$), the probability of error converges to 0 exponentially fast in the sense that

$$o\left(\left(\frac{K-1}{K}\right)^l\right). \tag{2.3}$$

However, when class-overlapping is present, the calculation of the convergence rate for $P_e(l,\infty)$ requires extra assumptions. Under the assumption of facing balanced multi-class classification problems, i.e. all priors are equiprobable, $\forall i, p(c_i) = 1/K$, I found that the optimal probability of error of multi-class problems may decrease to e_B exponentially fast in the number of labelled data l. Formally,

$$P_e(l,\infty) - e_B \leq 2\exp\left\{\frac{-l\lambda^2}{2K}\right\}, \tag{2.4}$$

where $\lambda \in (0,1]$ depends on the degree of intersection among the components of the mixture distribution and is defined by

$$\lambda = \frac{1}{K}\min_j\left\{\int_{R_j}\rho(\mathbf{x}|c_i)d\mathbf{x} - \max_{z\neq i}\int_{R_j}\rho(\mathbf{x}|c_z)d\mathbf{x}\right\}. \tag{2.5}$$

As seen in this upper bound, the threat of class-overlapping, discussed in Section 1.4, has a great influence in the performance of the resulting classifier.

In short, this work provides a general extension of previous key research in semi-supervised learning to the whole uni-dimensional domain and proves that, in this domain, the optimal probability of error decreases exponentially fast in the number of labelled data of the training dataset. Besides, some remarks on the impact of my proposal on the problem-solving of real-world problems are included in the manuscript – see Chapter 5 –. As future work, two interesting paths can be directly taken from the conclusions of this research. First, in this research, I assume that there are an infinite amount of unlabelled records but, which real number corresponds, in practise, to this number? Secondly, I propose sample complexities for l when I determine the minimum number of labelled data required. However, how do the complexity and dimensionality of the feature space impact on this matter?

2.3 A Study on the Competitiveness of Classifiers for Unbalanced Problems

Many classification problems show significant differences among the probabilities of the classes. This situation is known as the class-imbalance problem [29, 100] and it is widely considered to be a major obstacle to build competitive classifiers. Although a great methodological effort has been invested in proposing adequate algorithms and competitive solutions, the machine learning literature still lacks a solid theoretical foundation regarding the class-imbalance problem [105, 107]. Therefore, this research focuses on **pursuing a theoretical understanding of this scenario through a novel framework which enables the analysis on the most commonly used scores to assess the performance of classifiers**. Precisely, I aim at shedding some light on the theoretical foundation of the class-imbalance problem by providing a framework capable of answering the following three key questions which should certainly have an impact on future research:

Which performance scores are adequate to determine the competitiveness of a classifier in unbalanced uni-dimensional domains? I seek to determine which performance scores succeed in expressing the long studied performance detriment [124, 125] resulting from learning, in a classical manner[2], well-defined categories in moderate unbalanced scenarios. To accomplish this task, I define a novel controlled framework where the other factors hindering the performance of the classifiers can be cancelled so that the contribution of the class distribution to the performance scores can be legitimately quantified in isolation. Specifically, I focus on the numerical performance scores which can be expressed as Hölder [126] (a.k.a. power or generalised) means among the recalls of each class value [30]. Formally:

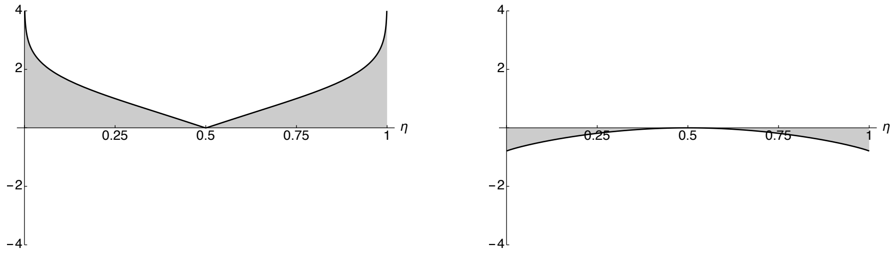
Let $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_K\}$ be the recalls obtained for a classifier, $\{\zeta_1, \zeta_2, \ldots, \zeta_K\}$ a series of non-negative weights s.t. $\sum_i \zeta_i = 1$ and $p \in \mathbb{R} \cup \{+\infty, -\infty\}$ an affinely extended real number, then, a Hölder mean is a mean of the form:

$$M_p = \left(\sum_{i=1}^{n} \zeta_i \mathcal{R}_i^p \right)^{\frac{1}{p}} . \tag{2.6}$$

Note that all the global scores defined in Table 1.1 can be included in these means. Now, these numerical performance scores can be defined as general functions $\mathcal{S}_\Psi(\boldsymbol{\eta}, \boldsymbol{\theta}) : (\Psi, \boldsymbol{\eta}, \boldsymbol{\theta}) \mapsto \mathbb{R}$, where Ψ is the used classifier, $\boldsymbol{\eta} = \{\eta_1, \eta_2, \ldots, \eta_K\}$, where $\eta_i = p(c_i)$ is the vector containing the class probabilities, and θ represents the model parameters of the generative function. Consequently, under the assumption of knowing the generative model, the BDR classifier was used to define a valuable measure in my proposed controlled framework:

*Let $\boldsymbol{\Theta}$ represent the space of parameters for a fixed family of distributions over the feature space for classification problems with K classes. Let $\mathcal{S}_B(\boldsymbol{\eta}, \boldsymbol{\theta})$ be the value of a performance score, $\mathcal{S}$, assessing the behaviour of the BDR on a K-class classification problem, γ, with a class distribution vector equal to $\boldsymbol{\eta}$ and parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Also, let $\mathcal{S}_B(\boldsymbol{e}, \boldsymbol{\theta})$ be the value of S evaluating the BDR inferred from the balanced version of γ. Therefore, the **influence***

[2] Minimising the 0-1 loss function. In this research, I assume the knowledge of the generative model so that the BDR can be used.



(a) An appropriate score ($\mathcal{G}$).

(b) An inappropriate score ($\mathcal{A}$cc).

Fig. 2.2: Examples of the influence function for two opposite scores.

function, $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$, *of the* K*-class distribution* $\boldsymbol{\eta}$ *on the performance score* $\mathcal{S}$ *using the* BDR *(*$\Psi = B$*) as a classifier is defined as follows:*

$$\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta}) = \int\limits_{\boldsymbol{\Theta}} [\mathcal{S}_B(\boldsymbol{e}, \boldsymbol{\theta}) - \mathcal{S}_B(\boldsymbol{\eta}, \boldsymbol{\theta})] d\boldsymbol{\theta}. \tag{2.7}$$

This function possesses the virtue of isolating the impact that the class distribution has on the performance of the classifiers (here, the BDR) when measured using the performance score, $\mathcal{S}$; it returns the average influence of the class distribution for every possible set of parameter values. Positive values of $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ denote that the performance score $\mathcal{S}$ obtains, on average, higher values for the BDR when it is used in a balanced scenario rather than when it is inferred from a class distribution, $\boldsymbol{\eta}$. Negative values mean the opposite; the BDR achieves, in general, worse values for $\mathcal{S}$ in the balanced scenario.

By exploiting prior research [14, 81, 107] on the expected behaviour of the BDR when facing a skewed class distribution, the shape that an adequate performance score should have for this influence function and, eventually, discern among the adequate and inadequate scores for unbalanced domains were determined:
A performance score $\mathcal{S}$ *is successful in being adequate to determine the competitiveness of a classifier* Ψ *in the class-imbalance scenario if, assuming a scenario where a classifier is inferred by directly minimising a* 0-1 *loss (maximising the classification accuracy),*

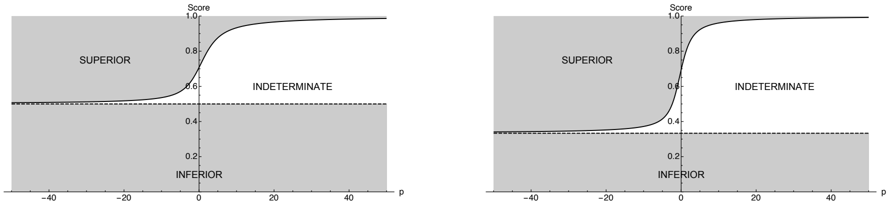
1. *its influence function is positive for almost any* $\boldsymbol{\eta} \neq \mathbf{e}$*, and*
2. *it shows a negative correlation to the minority class probability, i.e.* $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ *grows as the Euclidean distance between* $\boldsymbol{\eta}$ *and* $\mathbf{e}$ *gets larger.*

Figure 2.2 shows, in the binary scenario, an example of an adequate performance score whose influence function follows the previous properties (Figure 2.2a) and an example of an inadequate performance score (Figure 2.2b). Subsequently, with this study, I find evidence to support that numerical scores are sufficient for this intricate domain and that the performance scores which are unweighted Hölder means with $p \leq 1$ (a-mean, g-mean, h-mean, and min) among the recalls are the most appropriate to evaluate the competitiveness of classifiers in unbalanced problems. In these cases, misclassifying the least probable classes is highly penalised and this penalisation is increased as the value of p decreases.

Which performance scores are maximised in the most common learning solutions designed to deal with skewed classes? In this study, a review of the state-of-the-art is performed in order to devise what decision rules are behind the successful proposals of the literature, their competitiveness, and their inherently maximised scores. Particularly, I theoretically scrutinise data sampling [103] and cost-sensitive learning [127], which are the approaches that currently dominate the research efforts [11]. I conclude that most of the learning solutions proposed within those approaches are designed to asymptotically converge to the EDR. Moreover, I mathematically prove that the EDR is an optimal classifier for the unweighted Hölder mean with $p = 1$ (a-mean), and determine the advantages of the asymptotical usage of the EDR to deal with practical class-imbalance problems. However, the optimality for unweighted Hölder means with $p \neq 1$ does not hold. Afterwards, the implications of proposing approaches to solve unbalanced problems by maximising other $p \neq 1$ are discussed.

Can bounds guaranteeing the competitiveness of a classifier be provided for certain adequate performance scores? Yes, I finalise this research by fulfilling the necessity of the community for a standardised set of evaluation practices for proper comparisons among classifiers facing unbalanced data [11]. To be precise, I perform this task by providing two different practical bounds for the performance scores expressed as unweighted Hölder means among the recalls with $p \in \mathbb{R} \cup \{+\infty, -\infty\}$; a bound for the lowest value of the performance score ensuring a competitive solution for unbalanced problems and a bound for the highest value of the score indicating an incompetent solution (a classifier which is not better than the random classifier in at least one class value). Here, the conclusions are also consistent with the first study, using Hölder means with $p \leq 1$ is presumed to be adequate for determining the competitiveness of a classifier. In fact, both bounds coincide in $p = -\infty$. In Figure 2.3, the regions created using both bounds can be viewed for $p \in [-50, 50]$ in both binary and ternary problems.

The principal limitations of this theoretical research are the assumptions made; (i) the generative model is known, (ii) the concepts are well-defined, and (iii) the performance scores are restricted to numerical scores sharing the form of a Hölder mean. Thus, as future work, I strongly believe that the research lines should start by lessening those assumptions. In my humble opinion, the first step should be the use of several practical classifiers as representative classifiers rather than the BDR in a scenario where the generative is unknown.



(a) Binary classifier.

(b) Ternary classifier.

Fig. 2.3: Limiting values for the unweighted Hölder means ensuring that a given classifier is superior/inferior to the random guessing.

In short, with this set of theoretical contributions, I believe that, in the uni-dimensional classification setting, the current distance towards the ideal of defining more adequate learning systems capable of dealing with skewed class distributions in the multi-class scenario has been shortened.

2.4 A Measure for the Class-Imbalance Extent of Multi-class Problems

If it can be measured, it can be managed [128]. For that reason, several authors [108, 109, 111, 129] measure the class-imbalance degree of their experimental databases so that a better understanding of the unbalanced classification problems being addressed might be achieved. However, to the best of my knowledge, they use either tedious or suboptimal[3] measures when the binary class constraint is relaxed in the uni-dimensional classification framework. Therefore, **the last of the presented contributions is devoted to properly measuring the problematic class-imbalance extent in both binary and multi-class uni-dimensional problems.**

To characterise the class-imbalance extent of a classification problem, practitioners purposely use the distribution of class values available in the training set, $\hat{p}(c)$, as an estimation of the real class skewness of the generative model, $p(c)$. For the sake of clarity, since all the summaries can be used independently of the knowledge of the generative model, in this section, I also use $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_K)$, but denote indistinctly $p(c)$ and $\hat{p}(c)$.

Here, the most straightforward measures are either writing down the empirical class distribution [109] or to directly transcribing the occurrences of all class values of the dataset [111]. Although they are very informative for showing the degree of skewness at a glance, these measures become tedious when K gets large. Thus, authors usually opt to calculate a summary of the empiri-

[3] In terms of correlation with the performance detriment produced by the class-imbalance situation.

cal class distribution which is called *imbalance-ratio* (IR) [41]. It is calculated dividing the maximum statistic, η_i, by the minimum,

$$IR(\boldsymbol{\eta}) = \frac{\max_i \eta_i}{\min_j \eta_j}. \tag{2.8}$$

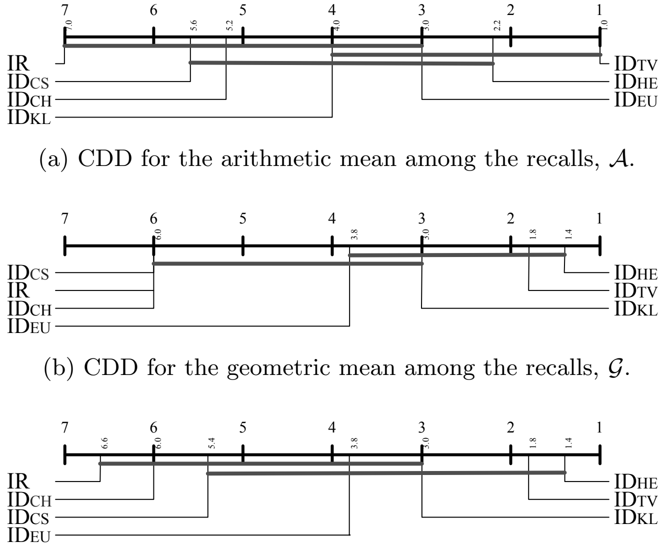
$IR : \boldsymbol{\eta} \mapsto \mathbb{R}$ is an injective function and, thus, an appropriate summary of the class-imbalance extent for binary problems. However, when K outnumbers 2, the injection is lost – as shown in the toy problem presented in [130]. There, diverse problems with different class-imbalance extents, indicating different complexities, share the same imbalance ratio. This may imply that IR, in the multi-class setting, is not correlated with the hindrance produced by skewed class distributions. So, in order to bridge this gap, I propose *imbalance-degree* (ID) as a new summary which is capable of properly shortening class distribution of both binary and multi-class classification problems into a single value. Formally, the ID of a class distribution, $\boldsymbol{\eta}$, is given by

$$ID(\boldsymbol{\eta}) = \frac{d_{\varDelta}(\boldsymbol{\eta}, \mathbf{e})}{d_{\varDelta}(\boldsymbol{\iota}_m, \mathbf{e})} + (m - 1), \tag{2.9}$$

where m is the number of minority classes, $d_{\varDelta}$ is the chosen distance/similarity function to instantiate ID, $\mathbf{e}$ is a balanced multi-class distribution with K-class values, and $\boldsymbol{\iota}_m$ is the distribution showing exactly m minority classes with the highest distance to $\mathbf{e}$. The proposed distance/dissimilarity summary has the following interesting properties:

1. By means of a single real value in the range $[0, K)$, it not only summarises the class distribution of a given problem, but also inherently expresses the number of majority and minority classes.
2. Depending on the requirements of the experimental setup and the degree of sensitivity sought, this measure can be instantiated with any common distance/dissimilarity function. In this research, I test, from the metrics in the vector space [131]; Manhattan, Euclidean (EU) and Chebyshev (CH) distances, and, from measures for probability distributions, the following f-divergences [132]; Kullback-Leibler (KL) divergence [133], Hellinger (HE) and total variation (TV) distances [134], and χ^2-divergence (CS) [135].
3. A unique mapping between the multi-class distributions and the numerical value of imbalance-degree is ensured for problems showing different numbers of majority and minority classes. Therefore, diverse problems cannot share a common numerical value as happens with imbalance-ratio.

Although the previous properties highlight an advantage of ID over IR, some experimentation is required in order to determine which summary is more appropriate to evidence the hindrance that skewed multi-class distributions cause in the learning processes. By means of the supervised learning



(a) CDD for the arithmetic mean among the recalls, $\mathcal{A}$.

(b) CDD for the geometric mean among the recalls, $\mathcal{G}$.

(c) CDD for minimum recall obtained, *min*.

Fig. 2.4: Pearson correlation ranking between the performance of the supervised learning paradigms (see Section 1.2.2) on the studied datasets and the summaries, $\alpha = 0.05$. The dissimilarity functions used to instantiate ID_{Δ} are specified in the subscript $\Delta = \{\text{EU,CH,KL,HE,TV,CS}\}$.

paradigms introduced in Section 1.2, I estimate their performances[4] in a key database [136] composed of 15 unbalanced datasets. Then, by just calculating the correlation between each summary and the resulting performances scores, the appropriateness of either ID or IR can be easily checked. The experimental results clearly indicate ID as a proper summary choice and IR as a suboptimal measure for multi-class problems; IR shows the worst Pearson correlations and significant differences (see Figure 2.4, where the ranking among the summaries is shown) are found between IR and ID. Finally, the experimentation was concluded by arguing that instantiating ID using either Hellinger or total variation distances produces significant robust summaries of the class-imbalance extent.

This contribution may be extended in several ways, being the most direct to perform a more exhaustive analysis over a larger number of distance/similarity functions, over a larger set of unbalanced problems or, even, introducing other measures for the other hindering aspects [89] presented in Section 1.4 which may harm the performance of the trained classifiers.

[4] Using the recommended a-mean, g-mean and minimum recall performance scores to assess unbalanced classification problems [30].

3

General Conclusions, Future Work, and Published Material

In this chapter, both general conclusions of this PhD work and potential future paths to extend the contributions are presented. Furthermore, the publications arising from this dissertation and included in Part II are also listed.

3.1 Conclusions

Real-world problems are complex and challenging. They can come in all sizes, shapes and varieties. However, when researching challenging situations on classification, the theoretical and methodological efforts of the literature tend to be biased towards simple classification schemas such as uni-dimensional binary problems. Instead of simplifying the statements of real-world problems so that the literature may be directly applied, this PhD dissertation opts to relax the common assumption of simple classification tasks. This enables me to push the existing classification schema boundaries further and to be able to directly apply the philosophy of taking on the real-world classification problems as they come. Specifically, here, I focus on generalising, from a both theoretical and methodological point of view, two current distinguished and defying situations called the semi-supervised learning paradigm and the class-imbalance problem.

Semi-supervised learning [10] is concerned with using unlabelled examples[1] in the learning task so that the performance of supervised learning methods that only use a limited labelled set to train the model can be improved [39]. Nowadays, this technique is still a subject undergoing intense study due to the fact that, in some applications, gathering labels for the training dataset is relatively expensive or time-consuming compared to the cost of obtaining unlabelled examples. Unfortunately, hardly any theoretical or methodological solution of the literature assumes classification schemes beyond uni-dimensional binary problems. Thus, in this dissertation, I devote a

[1] examples for which the features are known but not their corresponding categories

large research period to studying the semi-supervised learning paradigm but applied to more complex classification scenarios such as multi-class and multi-dimensional classification problems. The achieved contributions in that time span are listed as follows:

1. An EM algorithm [119] based semi-supervised approach capable of dealing with multiple class variables. This methodology enabled the extension of the semi-supervised learning framework to the whole multi-dimensional classification spectrum [57].
2. It is shown that, as happens in uni-dimensional classification [72], when using the generative structure to train a multi-dimensional classifier in a semi-supervised manner, the unlabelled data always helps. When there is a mismatch between the generative and the assumed structures, performance degradation of the trained classifier occurs.
3. A competent solution for a real-world multi-dimensional classification problem in the context of Sentiment Analysis [112] by means of the previous methodological advances. Specifically, a classifier capable of characterising the comments of the customers of several Spanish companies through three different, but related, dimensions was engineered. The sentiment of the users towards the product, the subjectivity of their online post, and their will to influence on other customers was extracted.
4. It is also proven that the probability of committing a classification error using a training dataset with no labelled data and any affinely extended real number of unlabelled data coincides with that of using the RAND classifier.
5. An optimal theoretical procedure for semi-supervised learning in the multi-class framework which allows the study of the fundamental limits of the methodological proposals in this setting. It is a generalisation of the pioneering optimal procedure for binary problems proposed in [31].
6. Theoretically, the minimum number of labelled examples required to train a multi-class classifier in a semi-supervised framework was also investigated.
7. It is proven, by means of the proposed optimal procedure, that the optimal probability of error in the semi-supervised framework might decrease to the Bayes error exponentially fast, but no faster, and that the class-overlapping [95, 106] had a direct influence on this convergence.

The second scenario, the **class-imbalance problem** [29, 100] arises from the class probability distribution, $p(\mathbf{c})$, of the generative model and it is able to compromise the performance of the majority part of standard learning algorithms [11]. Traditional learning algorithms assume a 0-1 loss function and, thus, they expect balanced class distribution or equal misclassification costs. So, when they are presented with complex training datasets sampled from skewed class distributions, these algorithms fail to properly learn the characteristics of the generative model and resultantly provide dummy classifiers always classifying incoming data as the most probable values [107]. Nowadays, it is a hot topic in the literature [89] and it is the subject of many papers, workshops, special sessions, and dissertations. Unsuitably, the class-imbalance literature is highly biased towards binary classification problems. Thus, in this dissertation, I theoretically and methodological contribute to that literature but assuming the whole uni-dimensional classification spectrum (binary + multi-class). Precisely, the presented contributions regarding the class-imbalance problem in this PhD work are the following:

1. A novel controlled theoretical framework where the other interdependent factors (see Section 1.4) threatening the performance of the trained classifier can be marginalised. By doing so, the contribution of the class distribution to the detriment of the performance of the classifier can be legitimately quantified in isolation.
2. Under the assumption of knowing the generative model, the BDR was used to define, in the suggested controlled framework, a valuable measure for the influence of the class-distribution in the score to assess the performance of a classifier.
3. Evidence to support that numerical scores are sufficient to adequately assess problems suffering from class skewness was found during this research project. Previous work [11] argued the opposite.
4. It is claimed that the performance scores which are unweighted Hölder means [126] with $p \leq 1$ (a-mean, g-mean, h-mean, etc.) among the recalls are the most appropriate to evaluate the competitiveness of classifiers in unbalanced problems.
5. I discovered that most of the learning solutions proposed in the literature under the approaches of data sampling [109] and cost-sensitive learning [110] were designed to asymptotically converge to the EDR.
6. The EDR is an optimal classifier for the unweighted Hölder mean with $p = 1$ (a-mean) as proven in this research work.
7. The necessity in the literature for a standardised set of evaluation practices for proper comparisons among classifiers facing unbalanced data [41] was fulfilled. This was achieved by providing two practical bounds for common performance scores ensuring both competent and incompetent classifiers.
8. A new summary of any binary and multi-class class distribution, which is capable of properly measuring the class-imbalance extent and which highly correlates with the detriment produced by class skewness, was proposed.

Finally, there is a contribution of this dissertation that, although it was key in the designed multi-dimensional semi-supervised methodological proposal, cannot be added to either of the previous lists. I also contributed to the supervised learning scenario by completing the family of multi-dimensional Bayesian network induction algorithms [57]. Specifically, I proposed a computational efficient supervised filter algorithm for MD J/K [58] networks which makes use of statistical testing over mutual information measures. Furthermore, this algorithm can also learn MDTAN structures in a much shorter time than using the original learning algorithm proposed in [57].

3.2 Future Work

The potential future extensions of this PhD work are briefly discussed in the following paragraphs.

In this dissertation, a particular real-world problem was solved by proposing a methodology which generalises **the semi-supervised learning framework** to the fresh multi-dimensional classification framework. Since it was an early exploration of this classification schema, there is, undoubtedly, a lot of work to be done here.

- Firstly, the collection of multi-dimensional classifiers is limited [23, 57, 58]. To the best of my knowledge, there are still no learning algorithms capable of dealing with ordinal features without nominalising them. A variety of these classifiers could significantly enrich the literature. Furthermore, I would like to point out the necessity of proposing computational efficient learning algorithms not only to ensure the scalability in the number of features and class variables, but also to fight the computational burden of semi-supervised learning techniques such as EM algorithm [119] based solutions.
- Secondly, different and diverse performance scores for multi-dimensional classification problems are required. When facing a new real-world uni-dimensional problem, there is always a positive probability of encountering at least one of the threats described in Section 1.4. When dealing with multi-dimensional problem, just based on the number of class variables, the probability of facing a threat is greater. Therefore, just having only a couple of subjective performance scores [23] to assess a multi-dimensional classifier seems to be a crucial limitation in this field. Consequently, I strongly believe that it is imperative to perform an exhaustive analysis on which uni-dimensional performance scores are adequate for being generalised to the multi-dimensional setting.
- Thirdly, in situations where labelled data are costly, the proposed methodology can be extended to deal with unlabelled records for which not all labels are missing.
- Fourthly, as advanced in the introduction, the natural evolution of the semi-supervised field (Section 1.2.3) can be used in the multi-dimensional framework. In this path, the foundation stone was laid with the described methodological proposal [23]. Consequently, learning algorithms leveraging the unlabelled data by introducing some assumptions linking the features and the class values might be proposed [73] in the literature.

In the literature review of the theoretical studies on semi-supervised learning, I was only able to find one study [70] dealing with the multi-class setting. However, it follows a different path than mine. Thus, instead of taking one giant leap by assuming the whole classification schema, I limited my theoretical scope to the uni-dimensional spectrum in order to settle this spectrum in the literature. I still believe that taking such an assumption is burdensome, so my suggested future research lines focus on populating the literature with valuable answers to theoretical semi-supervised questions in the multi-class scenario. Specifically, the following three different issues are key:

- Is there any number beyond which any extra additions of unlabelled data does not decrease the probability of error?
- In this dissertation, l_k was calculated as the expected minimum number of labelled data required in semi-supervised learning for uni-dimensional domains assuming that the complexity and dimensionality of the feature space do not affect its calculation. What happens if this assumption is relaxed?
- The pubished proposals follow a correct model assumption to study the optimal probability of error. When this assumption does not hold, how does the optimal probability of error varies in the number of labelled data, l? Does it still exponentially decrease in l?
- Finally, how do skewed class distributions affect the semi-supervised learning techniques? The optimal procedure for binary problems proposed in [31] can easily be modified to be optimal for the a-mean by just removing the priors in Stage 2 and by using the EDR in Stage 3. Thus, can the proposed theoretical work in the multi-class framework also be easily adapted to study the intricate class-imbalance problem in the semi-supervised framework?

Similarly to the semi-supervised learning framework, there was a lack of supporting works in **the class-imbalance literature**. This absence also prevented me from directly assuming a multi-dimensional framework. Thus, concerning the uni-dimensional framework, there are a few paths to directly extend the introduced work:

- Theoretically, the utilised novel framework to study the implication of the class-imbalance to the performance of the trained classifier can be easily complemented with several other classifiers rather than the BDR, other literature methodologies to deal with class skewness can be analysed and more performance scores can be tested.
- Methodologically, a more exhaustive analysis of the summaries for the class distributions of multi-class problems can also be proposed. A large number of distance/dissimilarity functions over a larger set of problems, or, even introducing other measures for the rest of threats which harm the performance of the classifiers can be used.

It is important to remark that, in this case, I believe that the generalisation of both works to the multi-dimensional scenario is approachable and beneficial to the literature.

- It can be easily seen that the theoretical framework proposed to study the implication of the class distribution on the performance of the trained classifier is directly generalisable using the global notation of this introductory part to the PhD dissertation. However, as happened in multi-class with the multi-minority and multi-majority cases, new and different class-imbalance scenarios will appear; an underrepresented nominal vector of class variables is a totally different class-imbalance case than having underrepresented class values for a given class variable. Moreover, the dependences between the class-variables will have an impact on this intricate threat. Having solved these adversities, the actual problem in this extension lies in the above mentioned limited amount of diverse performance scores in the multi-dimensional framework. Thus, here, I also highlight the necessity of an exhaustive study on the performance scores.

- Secondly, measuring the class-imbalance extent of multi-dimensional real-world problems also seems an upcoming step to take. Sadly, the introduced measure does not generalise to multiple class variables due to the fact that, first, it is necessary to establish which class distribution represents the balance scenario in multi-dimensional problems. Fortunately, the feasibility of this future work has been ensured in the literature. Charte et al [137] propose a generalisation of the imbalance-ratio to measure the class-imbalance extent of multi-class classification problems.

3.3 List of Publications

As a result of this research project, several articles have been published in JCR indexed journals. In this section, the articles selected to be included in Part II of this dissertation and their associated references are introduced.

Chapter 4: Approaching Sentiment Analysis by Using Semi-supervised Learning of Multi-dimensional Classifiers

- This article, cited more than 70 times, includes the contributions described in Section 2.1 of Part I. There, in order to solve a real-world problem related to the Sentiment Analysis domain, I not only propose the use of multi-dimensional classification for Sentiment Analysis domains, but also I contribute with a methodological extension of the multi-dimensional classification framework to the semi-supervised domain. Experimental results in this real-world problem show the potential benefits of the proposed semi-supervised multi-dimensional approach in Sentiment Analysis approaches. This work is published in *Neurocomputing* (Q2, Impact Factor: 1.632) as:

  J. Ortigosa-Hernández, J.D. Rodríguez, L. Alzate, M. Lucania, I. Inza and J.A. Lozano. Approaching Sentiment Analysis by Using Semi-supervised Learning of Multi-dimensional Classifiers. *Neurocomputing*, vol. 92, pp. 98-115, Sep. 2012.

Note: this work was published as a main paper and a separate supplementary file containing an extra set of experiments. Inside Chapter 4, both documents are included. Moreover, preliminary versions can be found published in a national conference [138] and as an internal report [139].

Chapter 5: Semi-supervised Multi-class Classification Problems with Scarcity of Labelled Data: A Theoretical Study

- The contribution described in Section 2.2 of Part I is encompassed in a manuscript published in a Q1 journal (Impact Factor: 6.018) named *IEEE Transactions on Neural Networks and Learning Systems*. It theoretically investigates, under a set of assumptions, the optimal procedure to solve multi-class classification problems in the semi-supervised setting, its associated probability of error and the minimum number of labelled data required to perform such a task. Moreover, the potential practical impact of the introduced proposals and the open challenges in semi-supervised learning are also discussed. This theoretical research can be found published as:

  J. Ortigosa-Hernández, I. Inza and J.A. Lozano. Semisupervised Multiclass Classification Problems With Scarcity of Labeled Data: A Theoretical Study. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27 num. 12, pp. 2602-2614, Dec. 2016.

  Note: Chapter 5 includes both the main paper and supplementary file containing mathematical proofs, artificial experiments and the source code. Both files were integrated and published as a single document in an internal report [140].

Chapter 6: Towards Competitive Classifiers for Unbalanced Multi-class Classification Problems: A Study on the Performance Scores

- This work aims at shedding some light on the theoretical foundation of the class-imbalance problem by addressing the three key questions exposed in Section 2.3 of Part I: (i) which performance scores are adequate to determine the competitiveness of a classifier in unbalanced domains? (ii) Which performance scores are maximised in the most common learning solutions designed to deal with skewed class distributions? (iii) Can bounds guaranteeing the competitiveness of a classifier be provided for certain adequate performance scores? This work is, at the present, under revision in *Machine Learning*, a prestigious JCR indexed journal in the second quartile with an impact factor of 1.855.

  J. Ortigosa-Hernández, I. Inza and J.A. Lozano. Towards Competitive

Classifiers for Unbalanced Multi-class Classification Problems: A Study on the Performance Scores. *Submitted to Machine Learning (second review process).*

Note: A preliminary version of this research can also be consulted in the public repository of electronic preprint arXiv [30], where it was published with reference *arXiv:1608.08984*.

Chapter 7: Measuring the Class-imbalance Extent of Multi-class Problems

- This manuscript is published in *Pattern Recognition Letters*, a Q2 journal with an impact factor of 1.952. It criticises the broad use of the imbalance-ratio to measure the imbalance extent of all kinds of unbalanced problems. Although it is an informative measure for binary problems, I prove that it is not adequate for the whole classification spectrum since it is incapable of completely and honestly describing disparity among the frequencies of more than two classes in the multi-class scenario. In order to overcome this drawback, a novel, normalised and sensitive measure for the class-imbalance extent, able to deal with both binary and multi-class classification problems, was proposed. This work presents the contribution described in Section 2.4 of Part I and it can be found in:

  J. Ortigosa-Hernández, I. Inza and J.A. Lozano. Measuring the Class-imbalance Extent of Multi-class Problems. *Pattern Recognition Letters*, vol. 98, pp. 32-38, Oct. 2017.

  Note: BIRD, the Institutional Repository Data of the Basque Centre for Applied Mathematics, holds a technical report [141] version of this research.

Part II

Selected Refereed Publications

4

Approaching Sentiment Analysis by Using Semi-supervised Learning of Multi-dimensional Classifiers

Upward, not Northward!

- Edwin A. Abbott, *Flatland: A Romance of Many Dimension*

Neurocomputing 92 (2012) 98–115

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Approaching Sentiment Analysis by using semi-supervised learning of multi-dimensional classifiers

Jonathan Ortigosa-Hernández [a,*], Juan Diego Rodríguez [a], Leandro Alzate [b], Manuel Lucania [b], Iñaki Inza [a], Jose A. Lozano [a]

[a] Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, Computer Science Faculty, The University of the Basque Country UPV/EHU, San Sebastián, Spain
[b] Socialware©, Bilbao, Spain

ARTICLE INFO

Available online 5 March 2012

Keywords:
Sentiment Analysis
Multi-dimensional classification
Multi-dimensional class Bayesian network classifiers
Semi-supervised learning
EM algorithm

ABSTRACT

Sentiment Analysis is defined as the computational study of opinions, sentiments and emotions expressed in text. Within this broad field, most of the work has been focused on either Sentiment Polarity classification, where a text is classified as having positive or negative sentiment, or Subjectivity classification, in which a text is classified as being subjective or objective. However, in this paper, we consider instead a real-world problem in which the attitude of the author is characterised by three different (but related) target variables: Subjectivity, Sentiment Polarity, Will to Influence, unlike the two previously stated problems, where there is only a single variable to be predicted. For that reason, the (uni-dimensional) common approaches used in this area yield to suboptimal solutions to this problem. Somewhat similar happens with multi-label learning techniques which cannot directly tackle this problem. In order to bridge this gap, we propose, for the first time, the use of the novel multi-dimensional classification paradigm in the Sentiment Analysis domain. This methodology is able to join the different target variables in the same classification task so as to take advantage of the potential statistical relations between them. In addition, and in order to take advantage of the huge amount of unlabelled information available nowadays in this context, we propose the extension of the multi-dimensional classification framework to the semi-supervised domain. Experimental results for this problem show that our semi-supervised multi-dimensional approach outperforms the most common Sentiment Analysis approaches, concluding that our approach is beneficial to improve the recognition rates for this problem, and in extension, could be considered to solve future Sentiment Analysis problems.

1. Introduction

Sentiment Analysis (SA), which is also known as Opinion Mining, is a broad area defined as the computational study of opinions, sentiments and emotions expressed in text [33]. It mainly originated to meet the need for organisations to automatically find the opinions or sentiments of the general public about their products and services, as expressed on the Internet. A fundamental methodology in many current SA applications and problems is the well-known pattern recognition field called *classification* [6].

Most of the work within this field has focused on the *Sentiment Polarity classification*, i.e. determining if an opinionated text has positive or negative sentiment [38]. However, motivated by different real-world problems and applications, researchers have considered a wide range of closely related problems over a variety of different types of corpora [37]. As an example of these problems, we can find the following: *Subjectivity classification*, which consists of determining if a text is subjective or objective [41], *Authorship identification*, which deals with the problem of identifying the author of a given text [2] or *Affect Analysis*, which recognises emotions in a text [1].

In an analogous fashion, a real-world application within this field has recently been tackled in Socialware©,[1] one of the most relevant companies in mobilised opinion analysis in Europe. The main goal of this application is to determine the attitude of the customers that write a post about a particular topic in a specific forum. The characterisation of these costumers is performed in this problem

[*] Corresponding author.
E-mail addresses: jonathan.ortigosa@ehu.es (J. Ortigosa-Hernández), juandiego.rodriguez@ehu.es (J.D. Rodríguez), leandro.alzate@asomo.net (L. Alzate), manuel.lucania@socialware.eu (M. Lucania), inaki.inza@ehu.es (I. Inza), ja.lozano@ehu.es (J.A. Lozano).

[1] http://www.asomo.net/indexE.jsp

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2012.01.030

by measuring three different dimensions: the sentiment, the subjectivity and the potential influence of each post in the forum.

By just relying on the previous work done in the SA domain, we can approach this problem by dividing it into three different subproblems (one per each dimension to be classified) and tackle them separately, i.e. study them in isolation by learning different classifiers to predict the value of each target variable as if they were independent.

However, some of the individual approaches explored in the literature for each subproblem could be adapted to the others. Moreover, a large number of papers [35,56] proposed approaches for the problems of Sentiment Polarity classification and Subjectivity classification, and sometimes by even using the same corpus. This could indicate a certain degree of correlation between these subproblems, and consequently between their target variables, as noticed in [37]. Nevertheless, this relation has never been directly demonstrated due to the fact that, to the best of our knowledge, no technique capable of dealing with several target variables has ever been used to embrace SA problems. In spite of that, it is relatively easy to notice the relation that exists between the sentiment and subjectivity (a neutral review probably indicates objectivity). So, why not learn a single classifier to classify the three dimensions simultaneously so as to make use of the statistical similarities between them? Finding a more predictive classifier by means of this thought would demonstrate that there is an actual relationship between these target variables. Also, in extension to the SA domain, why not join some previously cited problems in the same classification task in order to find more accurate multi-dimensional classifiers that take advantage of their closeness?

In order to embody this perception, we propose the use of the recently proposed multi-dimensional Bayesian network classification framework [53] to deal with multiple class classification problems in the context of SA by solving a real-world application. This methodology performs a simultaneous classification by exploiting the relationships between the class variables to be predicted. Note that, under this framework, several target variables could be taken into account to enrich the SA problem and create market intelligence.

Most papers have already addressed the SA task by building classifiers that exclusively rely on labelled examples [33]. However, in practice, obtaining enough labelled examples for a classifier may be costly and time consuming, an annotator has to read loads of text to create a reliable corpus. And this problem is accentuated when using multiple target variables. So, why not make use of the huge amount of unlabelled data available on the Internet to improve our solutions? Thus, the scarcity of labelled data also motivates us to deal with unlabelled examples in a semi-supervised framework when working with the exposed multi-dimensional view.

Motivated by the aforementioned comments, the following contributions are presented in this paper:

1. A novel and competitive methodology to solve the exposed real-world SA application.
2. An innovative perspective to manage the SA domain by dealing with several related problems in the same classification task.
3. The use of multi-dimensional class Bayesian network classifiers as supervised methodology to solve these multi-dimensional problems. This demonstrates that there is an actual correlation between sentiment and subjectivity as previously observed in several SA researches [35,37,56].
4. A supervised filter learning algorithm for multi-dimensional *J/K* dependences Bayesian network classifiers.
5. The extension of multi-dimensional classification to the semi-supervised framework by proposing a set of semi-supervised learning algorithms. This demonstrates that more predictive models can be found by making use of the unlabelled examples.

The rest of the paper is organised as follows. Section 2 describes the real multi-dimensional problem extracted from the SA domain which is solved in this paper, reviews the work related to SA and its problems, and motivates the use of multi-dimensional approaches in this context. The multi-dimensional supervised classification paradigm is defined in Section 3. Section 4 describes the multi-dimensional class Bayesian network classifiers. A group of algorithms to learn different types of multi-dimensional Bayesian classifiers in a supervised framework is introduced in Section 5. Section 6 not only introduces the idea of semi-supervised learning into the multi-dimensional classification, but also extends the supervised algorithms presented in Section 5 to the semi-supervised framework. Section 7 shows the experimental results of applying the proposed multi-dimensional classification algorithms using different feature sets to the real problem stated in Section 3. Finally, Section 8 sums up the paper with some conclusions and future work recommendations.

2. Problem statement and state-of-the-art review

2.1. The Sentiment Analysis domain

The concept of SA, motivated by different real-world applications and business-intelligence requirements, has recently been interpreted more broadly to include many different types of analysis of text, such as the treatment of opinion, sentiment or subjectivity [37].

Within this broad field, the most known problem is referred to as *Sentiment Polarity classification*, in which the problem of classifying documents by their overall sentiment is considered, i.e. determining whether a review is positive or negative [38]. Several papers have expanded this original goal by, for instance, adding a neutral sentiment [56] or considering a multi-point scale [48] (e.g. one to five stars for a review) or using sentences or phrases instead of reviews as input of the sentiment classifier [56].

Work in Sentiment Polarity classification often assumes the incoming documents to be opinionated [37]. For many applications, though, we may need to decide whether a given document contains subjective information or not. This is referred to as *Subjectivity classification* and has gained considerable attention in the research community [41,42,54]. Due to its ability to distinguish subjective texts from the factual ones, it is also of great importance in SA. There are works in which a subjectivity classifier is used to filter the objective documents from a dataset before applying a sentiment classifier [56,35]. There are even works in which the need to predict both the Sentiment Polarity and the Subjectivity has been noticed [21]. As stated in [37], "the problem of distinguishing subjective versus objective instances has often proved to be more difficult than subsequent polarity classification, so improvements in subjectivity classification promise to positively impact sentiment classification". From this quotation we can infer that Sentiment Polarity classification depends on Subjectivity classification and that there is a need to improve the methodologies used in Subjectivity classification.

Other closely related problems can be found in the SA domain: the set of problems called *Viewpoints and Perspectives* [37] which includes problems such as classifying political texts as liberal or conservative or placing texts along an ideological scale. *Authorship identification* deals with the problem of identifying the author of a given text [2]. *Affection Analysis* consists of extracting different types of emotions or affects from texts [1]. *Sarcasm Recognition* deals with the SA hard-nut problem of recognising sarcastic sentences [52]. More problems and applications are discussed in [37].

2.2. The ASOMO problem: a multi-dimensional perspective of SA

In this paper, we deal with a recent real-world problem studied in Socialware©. This problem is extracted from its *ASOMO service* of mobilised opinion analysis and it has an underlying multi-dimensional nature that can be viewed as an augmentation of the classical problems of Sentiment Polarity classification and Subjectivity classification.

The main goal of this application is to determine the attitude of a customer when he writes a post about a particular topic in a specific forum through three different dimensions: Sentiment Polarity and Subjectivity (as widely used in the SA domain), and a third one called Will to Influence, which is frequently used in the framework of ASOMO. The latter is defined as the dimension that rates the desire of the opinion holder to cause a certain reaction in the potential readers of the text. While some people leave a post on a forum to tell of their experience, others seek to provoke a certain kind of reaction in the readers. In our application, this class variable has four possible values: none (declarative text), question (soft Will to Influence), complaint/recommendation (medium Will to Influence) and appellation (strong Will to Influence). We use two example cases in order to introduce the ASOMO problem:

- A customer who has bought an iPhone does not know how to set up 3G on the phone, so he writes on a forum: “How can I configure 3G on my iPhone?”.
- Another customer is upset with the iPhone battery lifetime and writes in the same forum: “If you want long battery life, then don’t buy an iPhone”.

The attitude of both customers is very different. The first one has a doubt and writes to obtain a solution to his problem {neutral sentiment, objective and soft Will to Influence} while the second writes so as not to recommend the iPhone {negative sentiment, subjective and strong Will to Influence}. Fig. 1 shows one possible view of this problem and how we can translate the attitude of the author in three different class variables that are strongly correlated.

As previously mentioned, Will to Influence has four possible values: declarative text, soft, medium and strong will to influence. Sentiment Polarity, in this dataset, has five different labels as occurs in the 1–5 star ratings. In addition to the three classic values (positive, neutral and negative), it has the values “very negative” and “very positive”. Note that in using this approach, the label “neutral” in Sentiment Polarity is ambiguous, as happens in the SA literature [37]. So, it can be used as a label for the objective text (no opinion) or as a label for the sentiment that lies between positive and negative. As usual, Subjectivity has two values: objective and subjective. Fig. 2 shows not only the label distribution in the dataset for each different class variable (the three bar diagrams on the left), but also the joint label distribution of these three class variables over the labelled subset of the ASOMO dataset (the table on the right). Note that there are configurations of the joint label distribution that are equal to zero, this is because there are configurations of the class variables which are not possible, e.g. {strong Will to Influence, negative sentiment and objective}.

2.2.1. The ASOMO SA dataset

In order to deal with the previous problem, the ASOMO dataset was collected by Socialware©. This corpus was extracted from a single online discussion forum in which different customers of a specific company had left their comments, doubts and views about a single product. The forum consists of 2542 posts written in Spanish, with an average of 94 ± 74 words per post. One hundred and fifty of these documents have been manually labelled by an expert in Socialware© according to the exposed three different dimensions and 2392 are left as unlabelled instances.

As a result of the extensive work carried out by Socialware© on manually dealing with the ASOMO problem in the recent past, high levels of experience and understanding of determining the major factors that characterise the attitude of the customers have been gained. These factors are the following: (1) the implication of the author with the other customers in the forum, (2) the position of authority of the customer, and (3) the subjective language used in the text.

These broad factors have been helpful in detecting a list of 14 morphological features which characterise each analysed document. In order to engineer this list of features, each document is preprocessed using an open source morphological analyser [3,8]. Firstly, spelling in the entire corpus is checked. Then, the analyser provides information related to the part-of-the-speech (PoS) tagging [38]. Once the preprocessing task is performed, determining the values of the features is carried out by just looking for specific patterns in the corpus. In the following paragraphs, a detailed introduction of each factor is given, as well as a description of the features used in each factor.

The implication of the author: This factor covers the features that are related with the interaction between the author and the other customers in the forum. It consists of six different features that are described in Table 1. For each feature, we show its description and an example (with its translation into English) of the type of pattern that matches with the feature.

The position of authority of the opinion holder is mainly characterised by the purpose of the written post and it is related to the potential influence on the readers of the forum. The author

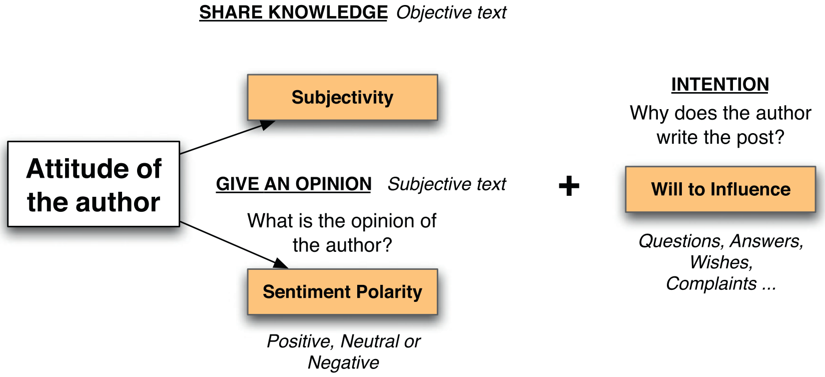
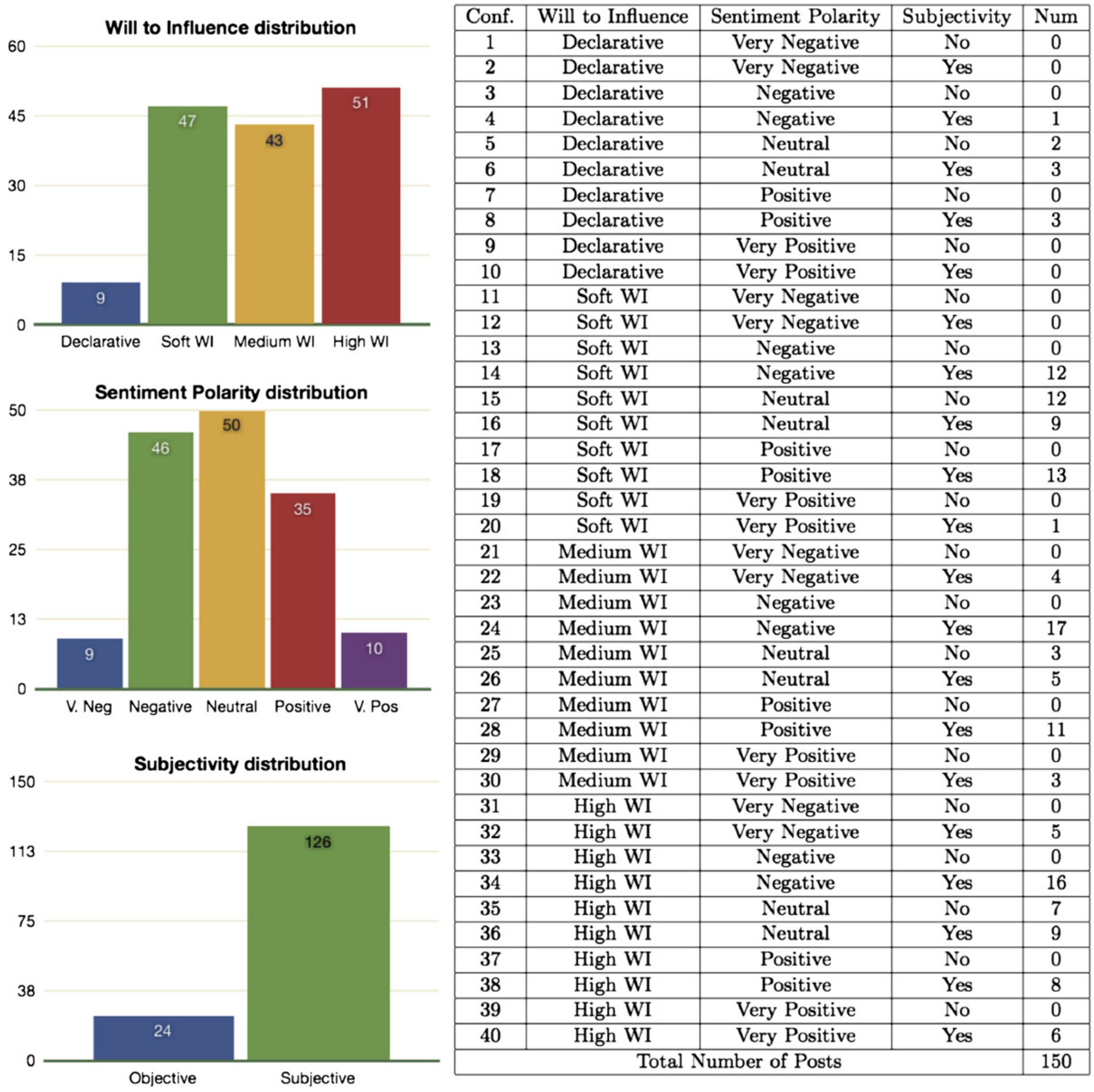

SHARE KNOWLEDGE Objective text
Subjectivity
Attitude of the author
GIVE AN OPINION Subjective text
What is the opinion of the author?
Sentiment Polarity
Positive, Neutral or Negative
+
INTENTION
Why does the author write the post?
Will to Influence
Questions, Answers, Wishes, Complaints ...

Fig. 1. The vision of Socialware© of the problem of determining the attitude of the writers in a forum.



Conf.	Will to Influence	Sentiment Polarity	Subjectivity	Num
1	Declarative	Very Negative	No	0
2	Declarative	Very Negative	Yes	0
3	Declarative	Negative	No	0
4	Declarative	Negative	Yes	1
5	Declarative	Neutral	No	2
6	Declarative	Neutral	Yes	3
7	Declarative	Positive	No	0
8	Declarative	Positive	Yes	3
9	Declarative	Very Positive	No	0
10	Declarative	Very Positive	Yes	0
11	Soft WI	Very Negative	No	0
12	Soft WI	Very Negative	Yes	0
13	Soft WI	Negative	No	0
14	Soft WI	Negative	Yes	12
15	Soft WI	Neutral	No	12
16	Soft WI	Neutral	Yes	9
17	Soft WI	Positive	No	0
18	Soft WI	Positive	Yes	13
19	Soft WI	Very Positive	No	0
20	Soft WI	Very Positive	Yes	1
21	Medium WI	Very Negative	No	0
22	Medium WI	Very Negative	Yes	4
23	Medium WI	Negative	No	0
24	Medium WI	Negative	Yes	17
25	Medium WI	Neutral	No	3
26	Medium WI	Neutral	Yes	5
27	Medium WI	Positive	No	0
28	Medium WI	Positive	Yes	11
29	Medium WI	Very Positive	No	0
30	Medium WI	Very Positive	Yes	3
31	High WI	Very Negative	No	0
32	High WI	Very Negative	Yes	5
33	High WI	Negative	No	0
34	High WI	Negative	Yes	16
35	High WI	Neutral	No	7
36	High WI	Neutral	Yes	9
37	High WI	Positive	No	0
38	High WI	Positive	Yes	8
39	High WI	Very Positive	No	0
40	High WI	Very Positive	Yes	6
Total Number of Posts				150

Fig. 2. Distribution of the labels of the three class variables over the labelled subset. The marginal distributions of Will to Influence, Sentiment Polarity and Subjectivity are represented as bar diagrams (left) and the joint distribution is represented in a table (right).

Table 1
Subset of features related to the implication of the author with other customers.

Feature	Description	Example	Translation
First persons	Number of verbs in the fist person	Contraté …	I hired …
Second persons	Number of verbs in the second person	Tienes …	You have …
Third persons	Number of verbs in the third person	Sabe …	He knows …
Relational forms	Number of phatic expressions, i.e. expressions whose only function is to perform a social task	(1) Hola (2) Gracias de antemano	(1) Hello (2) Thanks in advance
Agreement expressions	Number of expressions that show agreement or disagreement	(1) Estoy de acuerdo contigo (2) No tienes razón	(1) I agree with you (2) You're wrong
Request	Number of sentences that express a certain degree of request	(1) Me gustaría saber … (2) Alguien podría …	(1) I'd like to know … (2) I would appreciate it if someone could …

could express advice, disapproval with a specific product, predictions, etc. Table 2 shows the six features that are part of this major factor.

Subjective language deals with the opinion of the author. In order to determine this factor, we consider only the adjective detected with the PoS recogniser, as commonly carried out in the state-of-the-art literature [27]. Then, the adjectives are classified in polarity terms by means of a hand-annotated sentiment-lexicon. As a result of this task, we obtain two features: *Positive Adjectives* and *Negative Adjectives*, which are the number of positive and negative adjectives, respectively, in the text.

Table 2
Subset of features related to the position of authority of the customer.

Feature	Description	Example	Translation
Imperatives	Number of imperative verbs in the second person	No compres	Do not buy
Exhorts and Advice	Number of exhort verbs, e.g. recommend, advise, prevent, etc.	Te recomiendo …	I recommend that you …
Sufficiency Expressions	Number of expressions used to corroborate other sentences of the text	(1) Por supuesto (2) Naturalmente, …	(1) Of course (2) Naturally, …
Prediction Verbs	Number of verbs in the future	(1) Voy a probar (2) Llamaré.	(1) I'm going to try (2) I'll call
Authority	Number of expressions that denote high degree of authority, usually written in the subjunctive mode	Si fuera tú, …	If I were you, …
Questions	Number of question in the post, both direct and indirect	(1) ¿ Qué tal es? (2) Dime qué te parece	(1) How is it? (2) Tell me what you think of it

The 14 features (the ASOMO features) are normalised to be in the range [0,1] by dividing them by the maximal observed value.

2.3. The need for multi-dimensional classification techniques

In order to solve the typical problems of the SA domain (those exposed in Section 2.1), there are two main types of techniques that can be distinguished: *Symbolic* and *Machine Learning* [20]. The symbolic approach uses manually crafted rules and lexicons, whereas the machine learning approach uses supervised or semi-supervised learning to construct a model from a training corpus. Due to the fact that the main proposal of this paper is to solve a real SA problem by means of a novel machine learning technique, this paper focuses on the latter. The machine learning approaches have gained interest because of (1) their capability to model many features and, in doing so, capturing context, (2) their easier adaptability to changing input, and (3) the possibility to measure the degree of uncertainty by which a classification is made [20]. Supervised methods that train from examples which have been manually classified by humans are the most popular.

Most of the work that has been carried out in tuning up these machine learning techniques (as also happens in text processing tasks) has been dedicated to addressing the problem of converting a piece of text into a feature vector (i.e. model features able to capture the context of the text) in order to improve the recognition rates. The most common approaches use the single lower cased words (unigrams) as features, which in several cases reports pretty good results as in [38]. However, other common approaches can be found, such as *n*-grams [35] or PoS information [38]. A deeper study of such work is beyond the scope of this paper. The reader who is interested in feature engineering can consult [22], where there is an extensive body of work that addresses feature selection for machine learning approaches in general.

On the other hand, little research has been done on the induction of the classifiers. Most of the existing works learn either a naive Bayes [38,56] or a support vector machine (SVM) [2,38], i.e. uni-dimensional classifiers able to predict a single target variable. For that reason, the classification models used in the SA literature seem inappropriate to model the three-dimensional problem exposed in this paper. However, there are several possibilities to adapt these uni-dimensional classifiers to multi-dimensional classification problems, and the ASOMO problem is no exception. Unfortunately, none of these approaches captures exactly the underlying characteristics of the problem [44]:

- One approach is to develop multiple classifiers, one for each class variable. However, this approach does not capture the real characteristics of the problem, because it does not model the correlations between the different class variables and so, it does not take advantage of the information that they may provide. It treats the class variables as if they were independent. In the case of the previously exposed problem, it would be splitting it into three different uni-dimensional problems, one per each class variable.
- Another approach consists of constructing a single artificial class variable that models all possible combinations of classes. This class variable models the Cartesian product of all the class variables. The problem of this approach arises because this compound class variable can easily end up with an excessively high cardinality. This leads to computational problems because of the high number of parameters the model has to estimate. Furthermore, the model does not reflect the real structure of the classification problem either. By means of this approach, the ASOMO problem would be redefined as a uni-dimensional problem with a 40-label class variable.

The previous approaches are clearly insufficient for the resolution of problems where class variables have high cardinalities or large degrees of correlation among them. The first approach does not reflect the multi-dimensional nature of the problem because it does not take into account any correlation among the class variables. The second approach, however, does not consider the possible conditional independences between the classes and assumes models that are too complex. As can be seen in the experiments section, these deficiencies in capturing the real relationship between the class variables may cause a low performance, so new techniques are required to bridge the gap between the solutions offered by the learning algorithms used in the SA literature and the multi-dimensional underlying nature of the ASOMO problem.

Multi-label learning [51], which deals with problems with several labels per each instance, could also be viewed as a potential solution to this problem. However, as we show in the following section, the ASOMO problem cannot be directly tackled by the multi-label techniques. This problem is characterised for having several class variables, instead of several labels.

Within this framework, in order to yield more adequate models for problems with several target variables, multi-dimensional classification appears. It is able to use the correlations and conditional independencies between class variables in order to help in the classification task in both supervised and semi-supervised learning frameworks.

3. Multi-dimensional classification

In this section we present, in detail, the nature of the multi-dimensional supervised classification paradigm and how to define and evaluate a multi-dimensional classifier.

3.1. Multi-dimensional supervised classification problems

A typical (uni-dimensional) supervised classification problem consists of building a classifier from a labelled training dataset (see Table 3) in order to predict the value of a class variable C given a set of features $\mathbf{X}=(X_1,..,X_n)$ of an unseen unlabelled instance $\mathbf{x}=(x_1,..,x_n)$.

If we suppose that $(\mathbf{X},C)$ is a random vector with a joint feature-class probability distribution $p(\mathbf{x},c)$ then, a classifier ψ is a function that maps a vector of features $\mathbf{X}$ into a single class variable C

$$\psi : \{0,\dots,r_1-1\}\times\cdots\times\{0,\dots,r_n-1\}\mapsto\{0,\dots,t-1\}$$
$$\mathbf{x}\mapsto c$$

where r_i and t are the number of possible values of each feature $X_i,(i=1,\dots,n)$ and the class variable respectively.

A generalisation of this problem to the simultaneous prediction of several class variables has recently been proposed in the research community [5,17,40,43,44,53]. This generalisation is known as multi-dimensional supervised classification. Its purpose is to simultaneously predict the value of each class variable in the class variable vector $\mathbf{c}=(c_1,..,c_m)$ given the feature vector $\mathbf{x}=(x_1,..,x_n)$ of an unseen unlabelled instance. The training dataset, in this multi-dimensional framework, is expressed as shown in Table 4.

Thus, the classifier ψ becomes a function that maps a vector of features $\mathbf{X}$ into a vector of class variables $\mathbf{C}$

$$\psi : \{0,\dots,r_1-1\}\times\cdots\times\{0,\dots,r_n-1\}\mapsto\{0,\dots,t_1-1\}\times\cdots\times\{0,\dots,t_m-1\}$$
$$\mathbf{x}\mapsto\mathbf{c}$$

where r_i and t_j are the cardinalities of each feature X_i (for $i=1,\dots,n$) and each class variable C_j (for $j=1,\dots,m$) respectively. Note that we consider all variables, both predictive features and class variables, as discrete random variables.

A classifier is learnt from a training set (see Table 4) with a classifier induction algorithm $A(\cdot)$. Given the induction algorithm $A(\cdot)$, which is assumed to be a deterministic function of the training set, the multi-dimensional classifier obtained from a training set D is denoted as $\psi=A(D)$.

3.2. Related areas

In this paper we deal with multi-dimensional classification problems, and they must not be confused with other classification tasks which have similar designations, e.g. *multi-class* [50]: problems with a single class variable that can take more than two values, *multi-task* [9]: an inductive transfer approach, where a main task is predicted with the help of the prediction of some extra tasks, or *multi-label classification* [51]: where an instance can be classified with several different labels.

Table 3
A possible representation of a (uni-dimensional) labelled training dataset.

X_1	X_2	…	X_n	C
$x_1^{(1)}$	$x_2^{(1)}$	…	$x_n^{(1)}$	$c^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	…	$x_n^{(2)}$	$c^{(2)}$
⋮	⋮	⋮	⋮	⋮
$x_1^{(N)}$	$x_2^{(N)}$	…	$x_n^{(N)}$	$c^{(N)}$

Table 4
Representation of a multi-dimensional labelled training dataset.

X_1	X_2	…	X_n	C_1	C_2	…	C_m
$x_1^{(1)}$	$x_2^{(1)}$	…	$x_n^{(1)}$	$c_1^{(1)}$	$c_2^{(1)}$	…	$c_m^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	…	$x_n^{(2)}$	$c_1^{(2)}$	$c_2^{(2)}$	…	$c_m^{(2)}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$x_1^{(N)}$	$x_2^{(N)}$	…	$x_n^{(N)}$	$c_1^{(N)}$	$c_2^{(N)}$	…	$c_m^{(N)}$

Note, however, that a multi-label problem can be easily modelled as a multi-dimensional classification problem where each label or category is a binary class variable whose value is one when the instance is included in that category or zero otherwise. The opposite, which is redefining multi-dimensional problems as multi-label problems, seems very unnatural and has an important drawback: current multi-label methods cannot always handle the multi-dimensional nature of this kind of problems (illustrated in Example 1). This limitation gave rise to the development of multi-dimensional techniques, which nowadays, is differentiated to multi-label learning in the machine learning research community, see for instance [5,44]. These concerns can be demonstrated by setting up the following simple example:

Example 1. Suppose we consider a multi-dimensional problem where we want to determine the sex, the colour of the eyes and hair colour given several characteristics of a person (as represented in Table 5). The problem has five discrete predictive variables and three class variables: Sex, Eyes and Hair. Sex have two possible values: male and female, Eyes has three: blue, dark and green, and Hair has four possible labels: black, blonde, brown and ginger. Note that this kind of problem is similar to our application due to the fact that both have several class variables with more than two values.

If we wanted to tackle this problem by means of a multi-label algorithm, we would have to force it to fit in the multi-label framework. The most straightforward way to transfer it is to treat each value of each class variable as one independent label, i.e. treat each value Male, Female, Blue, Dark, etc. as a different label. In order to accomplish that the following conversion has to be done:

1. First, define each instance as a list of three labels (view the last column of Table 5), one per each class variable.
2. Second, deal with the main drawback that this approach has: there are several configuration of labels that are forbidden and/or senseless in the original multi-dimensional problem. For that reason, several restrictions to the multi-label technique have to be added in order to reflect the true nature of the problem. These constraints are the following:
   (a) Fix the number of labels per each instance, e.g. forbid the instances classified as {Male} or {Male, Blue, Black, Ginger}. Each instance must have just three labels.
   (b) Ensure that each instance has just one label per each class variable of the original multi-dimensional problem. We cannot classify an instance as {Male, Female}.

To the best of our knowledge, adapting multi-label techniques by adding several restrictions to deal with this type of problems, as stated in the previous paragraphs, has not been proposed by

Table 5
An example of a multi-dimensional problem and the most straightforward way to transfer it to the multi-label domain.

inst.	X_1	X_2	X_3	X_4	X_5	Sex	Eyes	Hair	List of labels
(1)	C	C	C	A	A	Male	Blue	Black	{Male, Blue, Black}
(2)	B	A	B	A	D	Male	Dark	Brown	{Male, Dark, Brown}
(3)	A	A	D	A	B	Female	Dark	Blonde	{Female, Dark, Blonde}
(4)	C	B	C	D	A	Male	Green	Blonde	{Male, Green, Blonde}
(5)	B	B	D	A	C	Female	Blue	Ginger	{Female, Blue, Ginger}

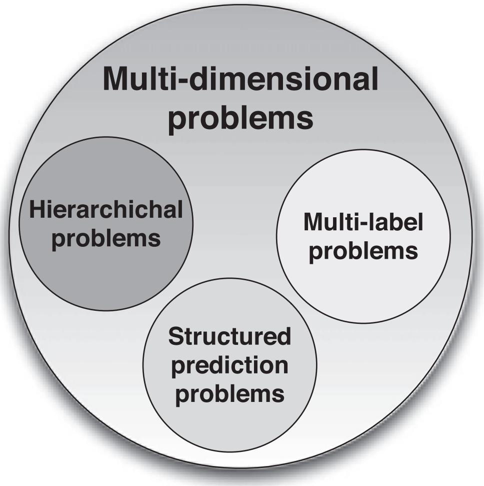


Fig. 3. Well known multi-dimensional classification subproblems, such as multi-label and multi-task learning, and structure prediction, contained in the set of multi-dimensional problems.

the research community. Therefore, it is not possible to fit most of the multi-dimensional problems (those that have at least one class variable with more than two values) into the current multi-label framework.

In analogous fashion, the exposed ASOMO problem cannot be solved by means of a multi-label technique. It has two class variables, Will to Influence and Sentiment polarity, with four and five values, respectively. For that reason, multi-label techniques cannot be applied to the application presented in this paper.

In addition to multi-label classification, other classification tasks in pattern recognition can also naturally be modelled as a multi-dimensional classification problem [5]. For instance, *structured prediction* [4,16], where there are several class variables with a conditional structure among them, or *hierarchical classification* [19], where there is a hierarchical structure (two or more levels) among the class variables. Therefore, the multi-dimensional techniques can be applied to these subproblems as the set of multi-dimensional problems contains these well known subproblems, as shown in Fig. 3.

3.3. Multi-dimensional classification rule

In probabilistic classification, the induction multi-dimensional algorithm learns a probability distribution $p(\mathbf{x},\mathbf{c})$ or $p(\mathbf{c}|\mathbf{x})$ from the training data and classifies a new unlabelled instance based on it. For that purpose, a classification rule must be defined.

In uni-dimensional supervised classification, the most common classification rule returns the most likely class value given the features

$$\hat{c} = \arg\max_{c}\{p(c|x_1,\ldots,x_n)\}$$

The multi-dimensional nature of the problem allows us to develop several classification rules that would make no sense in single-class classification because they take into account multiple class variables. Nevertheless, the previous one-dimensional classification rule can be easily generalised to the prediction of more than one class variable. In this case, the multi-dimensional classifier returns the most probable combination of class variables given the features. This rule is known as *joint classification rule* [44]

$$(\hat{c}_1,\ldots,\hat{c}_m) = \arg\max_{c_1,\ldots,c_m}\{p(c_1,\ldots,c_m|x_1,\ldots,x_n)\}$$

Although several other classification rules are proposed in [44], it is shown that the joint classification rule obtains better results.

3.4. Multi-dimensional classification evaluation

Once a classifier is constructed, its associated error needs to be measured. The prediction error of a single-class classifier ψ is the probability of the incorrect classification of an unlabelled instance $\mathbf{x}$ and is denoted as $\epsilon(\psi)$

$$\epsilon(\psi) = p(\psi(\mathbf{X}) \neq C) = E_{\mathbf{X}}[\delta(c,\psi(\mathbf{x}))]$$

where $\delta(x,y)$ is a loss function whose results are 1 if $x \neq y$ and 0 if $x=y$.

However, in multi-dimensional classification, the correctness of a classifier can be measured in two different ways:

- *Joint evaluation*: This consists of evaluating the estimated values of all class variables simultaneously, that is, it only counts a success if all the classes are correctly predicted, and otherwise it counts an error

  $$\epsilon(\psi) = p(\psi(\mathbf{X}) \neq \mathbf{C}) = E_{\mathbf{X}}[\delta(\mathbf{c},\psi(\mathbf{x}))]$$

  This rule is the generalisation of the previous single-class evaluation measure to multi-dimensional classification.
- *Single evaluation*: After a multi-dimensional learning process, this consists of separately checking if each class is correctly classified. For example, if we classify an instance $\mathbf{x}$ as $(\hat{c}_1=0,\hat{c}_2=1)$ and the real value is $(c_1=0,c_2=0)$, we count $\hat{c}_1$ as a success and $\hat{c}_2$ as an error. This approach provides one performance measure for each class C_j (for $j=1,\ldots,m$). The output of this evaluation is a vector $\boldsymbol{\epsilon}$ of size m with the performance function of the multi-dimensional classifier for each of the class variables

  $$\epsilon_j(\psi) = p(\psi_j(\mathbf{X}) \neq C_j) = E_{\mathbf{X}}[\delta(c_j,\psi_j(\mathbf{x}))]$$

  where $\psi_j(\mathbf{x})$ is the estimation of the multi-dimensional classifier for the j-th class variable.

Ideally, we would like to exactly calculate the error of a classifier, but in most real world problems the feature-label probability distribution $p(\mathbf{x},\mathbf{c})$ is unknown. So, the prediction error of a classifier ψ is also unknown; it cannot be computed exactly, and thus, must be estimated from data.

Several approaches to estimate the prediction error can be used. In this work, we use one of the most popular error estimation techniques: k-fold cross-validation (k-cv) [49] in its repeated version. In k-cv the dataset is divided into k folds, a classifier is learnt using $k-1$ folds and an error value is calculated by testing the learnt classifier in the remaining fold. Finally, the k-cv estimation of the error is the average value of the errors made in each fold. The repeated r times k-cv consists of estimating the error as the average of r k-cv estimations with different random partitions into folds. This method considerably reduces the variance of the error estimation [45].

In multi-dimensional classification we could be interested in either learning the most accurate classifier for all class variables simultaneously (measured with a joint evaluation) or in finding the most accurate classifier for each single class variable (measured with single evaluations). In this paper, we are mainly interested in using the joint evaluation for evaluation. However, in our application to SA we also measure the performance of the algorithms with a single evaluation per each class variable in order to compare both types of evaluation and perform a deeper analysis of the results.

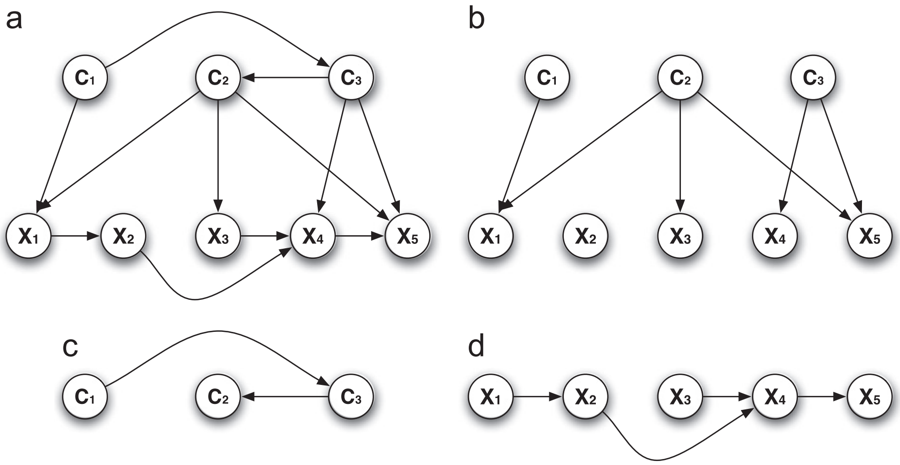


Fig. 4. A multi-dimensional Bayesian classifier and its division. (a) Complete graph, (b) feature selection subgraph, (c) class subgraph and (d) feature subgraph.

4. Multi-dimensional Bayesian network classifiers

In this section, multi-dimensional class Bayesian network classifiers [5,53], which are able to deal with multiple class variables to be predicted, are presented as a recent generalisation of the classical Bayesian network classifiers [32].

4.1. Bayesian network classifiers

Bayesian networks are powerful tools for knowledge representation and inference under uncertainty conditions [39]. These formalisms have been extensively used as classifiers [31] and have become a classical and well-known classification paradigm.

A Bayesian network is a pair $B=(S,\mathbf{\Theta})$ where S is a directed acyclic graph (DAG) whose vertices correspond to random variables and whose arcs represent conditional (in)dependence relations among variables, and where $\mathbf{\Theta}$ is a set of parameters.

A Bayesian classifier is usually represented as a Bayesian network with a particular structure. The class variable is on the top of the graph and it is the parent of all predictive variables.

In spite of the popularity of Bayesian network classifiers, few works have taken into account their generalisation to multiple class variables [5,17,43,44,53]. In multi-dimensional classification, we consider Bayesian networks over a finite set $\mathbf{V}=\{C_1,\ldots,C_m,X_1,\ldots,X_n\}$ where each class variable C_j and each feature X_i takes a finite set of values. $\mathbf{\Theta}$ is formed by parameters π_{ijk} and θ_{ijk}, where $\pi_{ijk}=p(C_i=c_k|\mathbf{Pa}(C_i)=\mathbf{Pa}(c_i)_j)$ for each value c_k that can take each class variable C_i and for each value assignment $\mathbf{Pa}(c_i)_j$ to the set of the parents of C_i. Similarly, $\theta_{ijk}=p(X_i=x_k|\mathbf{Pa}(X_i)=\mathbf{Pa}(x_i)_j)$ for each value x_k that can take each feature X_i and for each value assignment $\mathbf{Pa}(x_i)_j$ to the set of the parents of X_i.

Thus, the network B defines a joint probability distribution $p(c_1,\ldots,c_m,x_1,\ldots,x_n)$ which is given by

$$p(c_1,\ldots,c_m,x_1,\ldots,x_n)=\prod_{i=1}^{m}\pi_{ijk}\prod_{i=1}^{n}\theta_{ijk}$$

4.2. Structure of multi-dimensional class Bayesian network classifiers

A multi-dimensional class Bayesian network classifier is a generalisation of the classical one-class variable Bayesian classifiers for domains with multiple class variables [53]. It models the relationships between the variables by means of directed acyclic graphs (DAG) over the class variables and over the feature variables separately, and then connects the two sets of variables by means of a bi-partite directed graph. So, the DAG structure $S=(\mathbf{V},\mathbf{A})$ has the set $\mathbf{V}$ of random variables partitioned into the sets $\mathbf{V}_C=\{C_1,\ldots,C_m\}$, $m>1$, of class variables and the set $\mathbf{V}_F=\{X_1,\ldots,X_n\}$ $(n\geq 1)$ of features. Moreover, the set of arcs $\mathbf{A}$ can be partitioned into three sets: $\mathbf{A}_{CF}$, $\mathbf{A}_C$ and $\mathbf{A}_F$ with the following properties:

- $\mathbf{A}_{CF}\subseteq\mathbf{V}_C\times\mathbf{V}_F$ is composed of the arcs between the class variables and the feature variables, so we can define the feature selection subgraph of S as $S_{CF}=(\mathbf{V},\mathbf{A}_{CF})$. This subgraph represents the selection of features that seems relevant for classification given the class variables.
- $\mathbf{A}_C\subseteq\mathbf{V}_C\times\mathbf{V}_C$ is composed of the arcs between the class variables, so we can define the class subgraph of S induced by $\mathbf{V}_C$ as $S_C=(\mathbf{V}_C,\mathbf{A}_C)$.
- $\mathbf{A}_F\subseteq\mathbf{V}_F\times\mathbf{V}_F$ is composed of the arcs between the feature variables, so we can define the feature subgraph of S induced by $\mathbf{V}_F$ as $S_F=(\mathbf{V}_F,\mathbf{A}_F)$.

Fig. 4 shows a multi-dimensional class Bayesian network classifier with 3 class variables and 5 features, and its partition into the three subgraphs.

Depending on the structure of the three subgraphs, the following sub-families[2] of multi-dimensional class network classifiers are proposed in the state-of-the-art literature:

- *Multi-dimensional naive Bayes classifier (MDnB)*: The class subgraph and the feature subgraph are empty and the feature selection subgraph is complete.
- *Multi-dimensional tree-augmented Bayesian network classifier (MDTAN)*: Both the class subgraph and the feature subgraph are directed trees. It could be viewed as the multi-dimensional version of the (uni-dimensional) tree-augmented Bayesian network classifier (TAN) proposed in [24].
- *Multi-dimensional J/K dependences Bayesian classifier (MD J/K)*: This structure is the multi-dimensional generalisation of the well-known K-DB [47] classifier. It allows each class variable C_i to have a maximum of J dependences with other class variables C_j, and each predictive variable X_i to have, apart from the class variables, a maximum of K dependences with other predictive variables.

[2] In [53,17,43], instead of *multi-dimensional*, the term *fully* is used in order to name the classifiers.

In the following section, several algorithms are provided in order to learn from a given dataset the previous sub-families of multi-dimensional Bayesian network classifiers.

5. Learning multi-dimensional Bayesian network classifiers

As in the classic Bayesian network learning task, learning a multi-dimensional class Bayesian network classifier from a training dataset consists of estimating its structure and its parameters. These two subtasks are called structure learning and parameter learning respectively. Due to the fact that each previously introduced sub-family has different restrictions in its structure, a different learning algorithm is needed for each one.

In this section, we provide algorithms for learning MDnB [53], MDTAN [53] and MD *J*/*K* classifiers from a given dataset. The MDnB and MD *J*/*K* learning algorithms use a filter approach, i.e. the learning task precedes the classification evaluation. However, as it is proposed in its original work [53], the MDTAN learning algorithm is formulated as a wrapper approach [25], i.e. it tries to find more accurate classifiers by taking advantage of the classification evaluation.

As Fig. 5 shows, in the MDnB classifier, each class variable is parent of all the features, and each feature has only all the class variables as parents. Conventionally, the class subgraph and the feature subgraph are empty and the feature selection subgraph is complete. This classifier assumes conditional independence between each pair of features given the entire set of class variables. Due to the fact that it has no structure learning (the structure is fixed for a determined number of class variables and features), learning a MDnB classifier consists of just estimating the parameters Θ of the actual model by using a training dataset D. This is achieved by calculating the maximum likelihood estimator (MLE) [15].

Instead, learning a MDTAN classifier consists of learning both structure and parameters. A wrapper structure learning algorithm is proposed in [53]. Its aim is to produce the MDTAN structure (see Fig. 6) that maximises the accuracy from a given dataset. This algorithm has a main part called *Feature subset selection algorithm*, which follows a wrapper approach [25] by performing a local search over the A_{CF} structure. In order to obtain a MDTAN structure in each iteration, it generates a set of different A_{CF} structures from a current A_{CF} and learns its class subgraph and feature subgraph by using the following sub-algorithms:

1. A_C *structure learning algorithm* is the algorithm that learns the structure between the class variables by building a maximum weighted spanning [29] tree using mutual information.
2. A_F *structure learning algorithm* learns the A_F subgraph by using conditional mutual information, by means of the Chow and Liu algorithm [11].

After that, the accuracies of all the learnt models are computed. The iterative process continues by setting the A_{CF} of the best classifier, in terms of estimated accuracy, as current A_{CF}. This algorithm belongs to the hill-climbing family of optimisation algorithms, i.e. when no improvement is achieved by generating a new set of structures, the algorithm stops.

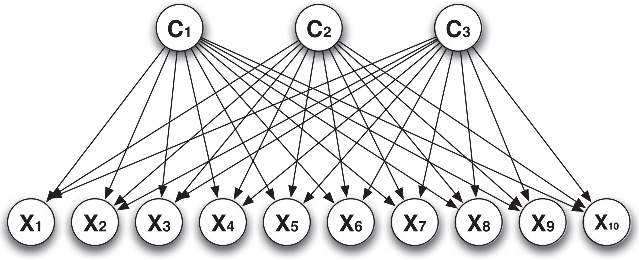


Fig. 5. An example of a multi-dimensional naive Bayes structure.

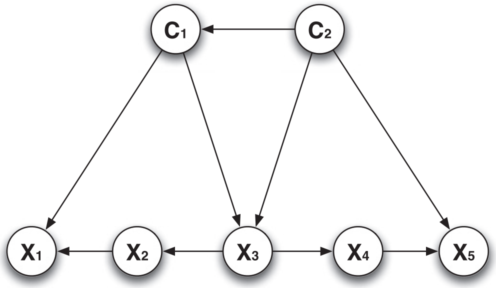


Fig. 6. An example of a multi-dimensional tree-augmented network structure.

5.1. Multi-dimensional J/K dependences Bayesian classifier

The MD *J*/*K* (see Fig. 7), which is introduced in [44], is the generalisation of the *K* dependence structure [47] to the multi-dimensional framework. It is able to move through the spectrum of allowable dependence in the multi-dimensional framework, from the MDnB to the full multi-dimensional Bayesian classifier. Note that from the setting $J=K=0$, we can learn a MDnB, setting $J=K=1$ a MDTAN structure is learned and so on. The full multi-dimensional Bayesian classifier, which is the classifier that has the three complete subgraphs, can be learnt by setting $J=(m-1)$ and $K=(n-1)$, where m and n are the number of class variables and predictive features respectively.

Although the MD *J*/*K* structure has been proposed in the state-of-the-art literature, to the best of our knowledge, a specific MD *J*/*K* learning algorithm for the multi-dimensional framework has not been defined by the research community. To bridge this gap, in this paper, we propose a filter algorithm in a supervised learning framework capable of learning this type of structure (see Algorithm 1).

In this algorithm, we do not directly use the mutual information as measured in the previous MDTAN learning algorithm [53]. This is due to the fact that the mutual information is not normalised when the cardinalities of the variables are different, so we use an independence test to determine if a dependence between two variables is strong enough to be part of the model: It

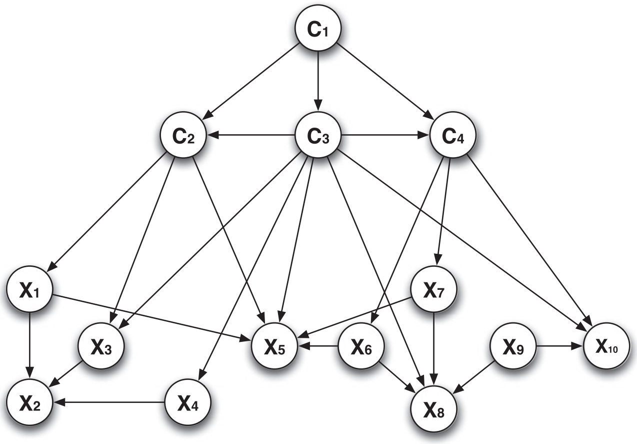


Fig. 7. An example of a multi-dimensional 2/3-dependence Bayesian network structure.

is known [30] that $2N\hat{I}(X_i,X_j)$ asymptotically follows a χ^2 distribution with $(r_i-1)(r_j-1)$ degrees of freedom, where N is the number of cases, if X_i and X_j are independent, i.e. $Lim_{N\to\infty}2N\hat{I}(X_i,X_j) \rightsquigarrow \chi^2_{(r_i-1)(r_j-1)}$.

Algorithm 1. A MD *J/K* structure learning algorithm using a filter approach.

```
1. Learn the $A_C$ structure
   1. Calculate the $p$-value using the independence test for each
      pair of class variables, and rank them.
   2. Remove the $p$-value higher than the threshold
      $(1-s_\alpha)=0.10$.
   3. Use the ranking to add arcs between the class variables
      fulfilling the conditions of no cycles between the class
      variables and no more than $J$-parents per class.
2. Learn the $A_{CF}$ structure
   1. Calculate the $p$-value using the independence test for each
      pair $C_i$ and $X_j$ and rank them.
   2. Remove the $p$-value higher than the threshold
      $(1-s_\alpha)=0.10$.
   3. Use the ranking to add arcs from the class variables to the
      features.
3. Learn the $A_F$ structure
   1. Calculate the $p$-value using the conditional independence
      test for each pair $X_i$ and $X_j$ given $\mathbf{Pa_c}(X_j)$ and rank them.
   2. Remove the $p$-value higher than the threshold
      $(1-s_\alpha)=0.10$.
   3. Use the ranking to add arcs between the class variables
      fulfilling the conditions of no cycles between the features
      and no more than $K$-parents per feature.
```

Based on this result, a statistical hypothesis test can be carried out in a multi-dimensional Bayesian network classifier to check the robust dependences in A_C. The null hypothesis H_0 is that the random variables C_i and C_j are independent. If the quantity $2N\hat{I}(C_i,C_j)$ surpasses a threshold s_α for a given test size

$$\alpha=\int_{s_\alpha}^{\infty}\chi^2_{(t_i-1)(t_j-1)}\,ds$$

where t_i is the cardinality of C_i and t_j the cardinality of C_j, the null hypothesis is rejected and a dependence between C_i and C_j is considered. Therefore the arc between these class variables is included in the model. The dependences in A_{CF} are calculated using the same procedure, the null hypothesis H_0 is that "The random variables C_i and X_j are independent". So, if $2N\hat{I}(C_i,X_j)$ surpasses the threshold s_α, then the null hypothesis is rejected and an arc is included in the model. This test was also used on single-class Bayesian network classifiers to check the dependences among the class variables and the features [7].

Using this approach, the structures A_C and A_{CF} are learnt in steps 1 and 2 of Algorithm 1, respectively.

In order to calculate the structure A_F, we need to use the conditional mutual information between a feature X_i and a feature X_j given its class parents $\mathbf{Pa_c}(X_j)$ to determine if the relation between both predictive features should be included in the model. For that purpose, we use the generalisation of the previous result to the case of conditional mutual information as defined in [30]

$$Lim_{N\to\infty}2N\hat{I}(X_i,X_j\,|\,\mathbf{Pa_c}(X_j)) \rightsquigarrow \chi^2_{(r_i-1)(r_j-1)(|\mathbf{Pa_c}(X_j)|)}$$

where r_i is the cardinality of X_i, r_j the cardinality of X_j and $|\mathbf{Pa_c}(X_j)|$ the cardinality of the class parents of X_j.

Analogously to the hypothesis test previously described, based on these results we can perform the following conditional independence test: the null hypothesis assumes that the random variables X_i and X_j are conditionally independent given $\mathbf{Pa_c}(X_j)$. So, if the quantity $2N\hat{I}(C_i,C_j\,|\,\mathbf{Pa_c}(X_j))$ surpasses a threshold s_α for a given test size

$$\alpha=\int_{s_\alpha}^{\infty}\chi^2_{(t_i-1)(t_j-1)(|\mathbf{Pa_c}(X_j)|)}\,ds$$

the null hypothesis is rejected and the random variables X_i and X_j are considered dependent given $\mathbf{Pa_c}(X_j)$. Therefore, the arc is included in the model. The structure A_{CF} is learnt using this hypothesis test in step 3 of Algorithm 1.

6. Semi-supervised multi-dimensional classification

In this section, we proceed with the extension of the previous multi-dimensional learning algorithms to the semi-supervised learning framework. When large amounts of labelled data are available, one can apply familiar and powerful machine learning techniques such as the previous multi-dimensional Bayesian network algorithms in order to learn accurate classifiers. However, when there is a scarcity of such labelled data and a huge amount of unlabelled data, as happens in the SA domain, one can wonder if it is possible to learn competitive classifiers from unlabelled data.

In this context, where the training dataset consists of labelled and unlabelled data, the semi-supervised learning approach [10,57,58] appears as a promising alternative. It is motivated from the fact that in many real world problems, obtaining unlabelled data is relatively easy, e.g. collecting posts from different blogs, while labelling is expensive and/or labor intensive, due to the fact that the tasks of labelling the training dataset is usually carried out by human beings. Thus, it is highly desirable to have learning algorithms that are able to incorporate a large number of unlabelled data with a small number of labelled data when learning classifiers.

In the semi-supervised learning framework, the training dataset D, as shown in Table 5, is divided into two parts: the subset of instances D_L for which labels are provided, and the subset D_U, where the labels are not known. Therefore, we have a dataset of N instances, where there are L labelled examples and $(N-L)$ unlabelled examples. Normally, $(N-L)\gg L$, i.e. the unlabelled subset tends to have a very large amount of instances whilst the labelled subset tends to have a small size.

Therefore, the aim of a semi-supervised learning algorithm is to build more accurate classifiers using both labelled and unlabelled data, rather than using exclusively labelled examples as happens in supervised learning.

6.1. Learning multi-dimensional Bayesian network classifiers in the semi-supervised framework

In this section, we propose the extension of multi-dimensional Bayesian network classifiers to the semi-supervised learning framework by using the EM algorithm [18]. Although this method was proposed in [18] and was deeply analysed in [34], it had been used much earlier, e.g. [26], and it is still widely used in many recent semi-supervised learning algorithms, e.g. [13,14,36].

The aim of the EM algorithm as typically used in semi-supervised learning is to find the parameters of the model that maximise the likelihood of the data, using both labelled and unlabelled instances. The iterative process, which ensures that the likelihood is maximised in each step, works as follows: in the Kth iteration the algorithm alternates between completing the unlabelled instances by using the parameters $\mathbf{\Theta}^{(K)}$ (E-step) and updating the parameters of the model $\mathbf{\Theta}^{(K+1)}$ using MLE with the whole dataset (M-step), i.e. the labelled data and the unlabelled instances that have been previously classified in the E-Step. Note that the structure remains fixed in the whole iterative process.

Although good results have been achieved with the EM algorithm in uni-dimensional classification [13,36], we are concerned about the restriction of only maximising the parameters of a fixed structure in our extension of the EM algorithm to the multi-dimensional domain, where there are several class variables to be predicted. As stated in [12], if the correct structure of the real distribution of the data is obtained, unlabelled data improve the classifier, otherwise, unlabelled data can actually degrade performance. For this reason, it seems more appropriate to perform a structural search in order to find the real model. Thus, we perform several changes to the EM algorithm in order to avoid fixing the structure of the model during the iterative process. The proposal is shown in Algorithm 2.

Algorithm 2. Our version of the EM Algorithm.

```
Input A training dataset with both labelled and unlabelled data (Table 6) and an initial model $\psi^{(K=0)}$ with a fixed structure and with an initial set of parameters $\boldsymbol{\Theta}^{(K=0)}$.
while the model $\psi^{(K)}$ does not converge do
    E-STEP Use the current model $\psi^{(K)}$ to estimate the probability of each configuration of class variables for each unlabelled instance.
    M-STEP Learn a new model $\psi^{(K+1)}$ with structure and parameters, given the estimated probabilities in the E-STEP.
end while
Output classifier $\psi$, that takes an unlabelled instance and predicts the class variables.
```

In this version of the EM algorithm, we want to find the model, both structure and parameters, that maximises the likelihood of the whole dataset. So, in this version, the iterative process is performed as follows: in the Kth iteration, the algorithm alternates between completing the unlabelled instance by the previously learnt model $\psi^{(K)}$ (E-step) and learning a new model $\psi^{(K+1)}$ by using a learning algorithm with the whole dataset, both labelled and completed instances (M-step). In the semi-supervised learning research community, the input initial parameter $\psi^{(K=0)}$ of the EM Algorithm is usually learnt from the labelled subset D_L. Hence, we will continue to use this modus operandi in this version of the algorithm. Note that our version of the EM algorithm is closer to the Bayesian structural EM algorithm proposed in [23] rather than the original formulation of the algorithm [18]. However, in the case of the MDnB classifier, it is just a parametric search since it has a fixed structure.

Table 6
A formal representation of a multi-dimensional semi-supervised training dataset.

	X_1	X_2	…	X_n	C_1	C_2	…	C_m
D_L	$x_1^{(1)}$	$x_2^{(1)}$	…	$x_n^{(1)}$	$c_1^{(1)}$	$c_2^{(1)}$	…	$c_m^{(1)}$
	$x_1^{(2)}$	$x_2^{(2)}$	…	$x_n^{(2)}$	$c_1^{(2)}$	$c_2^{(2)}$	…	$c_m^{(2)}$
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	$x_1^{(L)}$	$x_2^{(L)}$	…	$x_n^{(L)}$	$c_1^{(L)}$	$c_2^{(L)}$	…	$c_m^{(L)}$
D_U	$x_1^{(L+1)}$	$x_2^{(L+1)}$	…	$x_n^{(L+1)}$	?	?	…	?
	$x_1^{(L+2)}$	$x_2^{(L+2)}$	…	$x_n^{(L+2)}$	?	?	…	?
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	$x_1^{(N)}$	$x_2^{(N)}$	…	$x_n^{(N)}$	?	?	…	?

Using Algorithm 2, all the supervised learning approaches proposed in the previous section can be straightforwardly used in this semi-supervised scenario. The learning algorithm is used in the M-step, where it learns a model using labelled and unlabelled data that have been previously labelled in the E-step. So, applying our adaptation of the EM Algorithm, we have extended the multi-dimensional Bayesian network classifiers to the semi-supervised learning framework.

7. Experimentation

7.1. Artificial experimentation

Before solving the ASOMO problem, we have tested our proposed semi-supervised algorithms over a set of designed artificial datasets as commonly carried out in the machine learning research community. This has been done due to the fact that, unfortunately, we cannot apply our proposals in the baseline datasets of the SA domain. The multi-dimensional classification paradigm has recently been proposed in the research community [5,17,40,43,44,53], and, to the best of our knowledge, there are no benchmark multi-dimensional datasets in this domain to test our proposals as they consider just one target variable at a time (usually sentiment polarity). So, in order to bridge this gap, we provide a detailed report of these artificial experiments on several synthetic datasets in the following website.[3]

The major conclusions extracted from this experimentation can be summarised as follows:

1. As happens in the uni-dimensional framework [12], when using the real structure to semi-supervisely learnt multi-dimensional classifiers, the unlabelled data always help.
2. There is a tendency to achieve better classifiers in terms of joint accuracy in the semi-supervised framework when the used multi-dimensional algorithm can reach the generative structure.
3. In the uni-dimensional approaches, performance degradation occurs in the semi-supervised framework. This is probably due to the fact that the uni-dimensional approaches are not able to match the actual multi-dimensional structure of the problems.
4. Although there are small differences between the uni-dimensional and the multi-dimensional approaches in the supervised framework (only the MD J/K reports statistical differences), in the semi-supervised framework these differences grow larger (except for the case of the MDTAN learning algorithm, the qrest of the multi-dimensional approaches report statistical differences).
5. In the semi-supervised framework, clearly the multi-dimensional classifiers outperform the uni-dimensional techniques, with the exception of the MDTAN classifier.
6. The MDnB learning algorithm [53] is very specific, it obtains very good results when dealing with problems with an underlying MDnB structure, but when the generative models are more complex, its rigid structure makes the algorithm lead to very suboptimal solutions.
7. The MDTAN algorithm [53] also shows very poor performances in the semi-supervised framework.
8. The MD J/K learning algorithms have great flexibility to capture different types of complex structures, which results in an improvement in terms of joint accuracy in the semi-supervised framework.

[3] http://www.sc.ehu.es/ccwbayes/members/jonathan/home/News_and_Notables/Entries/2010/11/30_Artificial_Experiments_2010.html

In brief, these artificial experiments show that not only the multi-dimensional approaches statistically outperform the uni-dimensional approaches in the supervised framework when dealing with multi-dimensional problems, but also more accurate classifiers can be found using the semi-supervised learning framework.

7.2. *Solving the ASOMO SA problem*

Two different series of experiments with the ASOMO corpus were performed: the first series (Section 7.2.1) shows the comparison between the ASOMO features and three different state-of-the-art feature sets broadly used in the SA domain. The second series (Section 7.2.2) is devoted to demonstrate that the addition of unlabelled instances could achieve better results in this very application. There, a multi-dimensional classification solution for the ASOMO problem is proposed and analysed. By means of these experiments in a real SA problem, we would like to shed some light on the truthfulness of the following hypotheses:

1. The choice of the feature set is a key matter when dealing with the exposed problem, and in extension, with the SA problems.
2. The uni-dimensional models obtained with the common approaches of the SA domain yield suboptimal solutions to the ASOMO problem.
3. The explicit use of the relationships between the class variables in this real-world problem can be beneficial to improve their recognition rates, i.e. multi-dimensional techniques are able to outperform the most common uni-dimensional techniques.
4. When there is a scarcity of labelled data, multi-dimensional techniques can work with unlabelled data in order to improve the classification rates in this context.

7.2.1. *Comparison between different feature sets in both uni-dimensional and multi-dimensional learning scenarios*

By means of this experiment, we evaluate the new feature set proposed in Section 2 (the ASOMO features) with the most commonly used in the related literature. If we only use them to test the multi-dimensional Bayesian network classifiers in this real dataset, it is difficult to assess their efficacy without using other viable baseline methods. For this reason, we also use three different commonly used feature sets in order to perform a benchmark comparison: *unigrams*, *unigrams+bigrams* and *PoS*. In order to avoid computation burden, we limited consideration to (1) the 357 unigrams that appear at least five times in our 150-post corpus, (2) the 563 unigrams and bigrams occurring at least five times, and (3) the PoS to 766 features. No stemming or stoplists were used. These benchmark feature sets have been constructed using the TagHelper tool [46]. Finally, since the ASOMO features are continuous, they were discretised into three values using equal frequency discretisation in order to apply the algorithms proposed in this paper.

To assess the efficacy and efficiency of the exposed multi-dimensional Bayesian network classifiers in this problem, we conducted a comparison in the supervised framework between the presented multi-dimensional classifiers and the two different uni-dimensional attempts to adapt single-class classifiers to multi-dimensional problems: (1) develop multiple uni-dimensional classifiers and (2) construct a Cartesian product class variable. In order to perform such comparison, we use naive Bayes classifiers in the uni-dimensional attempts (one per each class variable in the first uni-dimensional approach, and another one for the compound class of the second) and a MDnB classifier in the multi-dimensional solution. The parameters of both types of models are calculated by MLE corrected with Laplace smoothing and the forbidden configurations of the vector of class variables have been taken into account in the learning process. Note that this comparison is performed in the supervised framework, so the 2392 unlabelled posts were not used in this experiment.

The classification accuracies, which have been estimated via 20 runs of 5-fold non-stratified cross validation (20 × 5cv) [45], are shown in Table 7. The results of using the four different feature sets (unigrams, unigrams + bigrams, PoS, and the ASOMO features) in conjunction with the three different learning approaches (multiple uni-dimensional, Cartesian class variable, and multi-dimensional classifiers) in a supervised framework are shown. The first eight rows of Table 7 correspond to the two uni-dimensional approaches while the last four correspond to the multi-dimensional approach. In addition to the estimation of the joint accuracy, the accuracies per each class variable are also shown.

For each column, the best estimated accuracy rate is highlighted in **bold** for each single class variable and the joint accuracy. Moreover, statistical hypothesis tests (Student's t-test with $\alpha = 0.05$) have been performed in order to determine if there are significant statistical differences between the tested accuracies. For each column, the symbol '†' is used to mark the accuracies that are statistically outperformed by the highlighted best estimated accuracy (in **bold**). The symbol '‡' corresponds to the accuracies that, despite not being significantly worse, show a p-value on the significant borderline ($0.05 < p\text{−value} \leq 0.1$). Besides, the CPU time spent on learning a classification model with the 150 posts (for each approach and feature set) is shown in the results.

Table 7
Estimated accuracy values on the ASOMO dataset using three different types of feature sets in both uni and multi-dimensional scenarios (20 × 5cv).

Approach	Classifier	Feature set	#feat.	Will to influence Acc	Sentiment P. Acc	Subjectivity Acc	Joint Acc	Time (ms)
Uni-dimen.	Multiple classif.	Unigrams	357	†51.83 ± 2.79	33.00 ± 1.82	†78.90 ± 2.16	†11.70 ± 1.68	1011
		Uni.+bigrams	563	†47.17 ± 2.47	32.90 ± 1.96	†79.47 ± 2.75	†9.80 ± 1.47	1470
		PoS	766	†47.57 ± 2.52	†30.80 ± 2.75	†66.33 ± 2.96	†9.43 ± 1.65	1755
		ASOMO feat.	14	**55.07 ± 1.32**	†26.07 ± 2.09	†82.17 ± 1.35	†10.37 ± 2.30	110
	Cartesian class	Unigrams	357	N/A	N/A	N/A	†9.90 ± 2.35	32
		Uni.+bigrams	563	N/A	N/A	N/A	†8.87 ± 2.05	74
		PoS	766	N/A	N/A	N/A	†10.33 ± 2.11	106
		ASOMO feat.	14	N/A	N/A	N/A	‡13.70 ± 2.60	17
Multi-dimen.	MDnB	Unigrams	357	†41.63 ± 3.74	†31.00 ± 2.29	†75.30 ± 2.20	†10.23 ± 1.92	47
		Uni.+bigrams	563	†38.03 ± 3.33	**33.40 ± 2.37**	†74.27 ± 2.71	†9.63 ± 2.05	107
		PoS	766	†39.50 ± 3.32	†30.53 ± 2.22	†76.40 ± 2.47	†9.86 ± 1.93	122
		ASOMO feat.	14	†53.23 ± 1.62	†30.87 ± 2.52	**83.53 ± 0.69**	**14.97 ± 1.94**	17

Based on the results of Table 7, different conclusions can be extracted:

1. With respect to the feature set comparison, the ASOMO feature set significantly outperforms the n-grams, not only in terms of joint accuracy, but also in terms of accuracy of two (out of the three) dimensions, i.e. in Will to Influence and in Subjectivity. Also, as stated in [38], n-grams information is the most effective feature set in the task of Sentiment Polarity classification. However, the unigrams + bigrams feature set outperforms unigrams instead of the opposite as reported in [38]. Finally, the PoS approach obtains very low performance.
2. According to the comparison between the uni-dimensional and multi-dimensional approaches, the best joint accuracy is given by the MDnB model when using the ASOMO features. It significantly outperforms both the multi-dimensional approaches with the state-of-the-art feature sets, and the uni-dimensional approaches. Looking at the class variables in isolation, the uni-dimensional approaches only outperform the multi-dimensional approach in the case of the Will to Influence target dimension.
3. An analysis of the CPU computational times show that, as expected, learning one classifier per each class variable is the most time-consuming. Next, multi-dimensional Bayesian network classifiers are found. The least-time consuming approach is the uni-dimensional learning process that uses a compound class. Moreover, the number of features is also an important issue with respect to the computational time. The ASOMO feature set, which has only 14 attributes, is, by far, the least time-consuming of the four different types of feature set.

However, in this problem, errors are not simply present or absent, their magnitude can be computed, e.g. it is not the same to misclassify a negative post as having a very negative sentiment or misclassify it with a very positive sentiment. For that reason, in addition to the accuracy term, we also estimate the numeric error of each classifier. Note that the values of the three class variables can be trivially translated into ordinal values without changing their meaning. Therefore, using this approach, the previous example could be exposed as: it is not the same misclassify a post, which has its sentiment equal to 2, with a 1 or misclassify it with a 5.

In order to estimate the numeric error, we use the mean absolute error (MAE) term [55], which is a measure broadly used in evaluating numeric prediction. It is defined as the measure that averages the magnitude of the individual errors without taking their sign into account. It is given by the following formula:

$$MAE\epsilon_j(\psi) = \frac{\sum_{i=1}^{N} |\psi_j(\mathbf{x}^{(i)}) - c_j^{(i)}|}{N}$$

where $\psi_j(\mathbf{x}^{(i)})$ is the value of the class variable C_j resulting from the classification of the instance $\mathbf{x}^{(i)}$ using the classifier ψ_j and $c_j^{(i)}$ is the actual class value in that instance. N is the number of instances in the test set. Note that the resulting error varies between 0 and $(|C_j|-1)$, where $|C_j|$ is the cardinality of the class variable C_j.

In a similar way to the accuracy, we also compute a joint measure for simultaneously characterising this error in all the class variables. Due to this, we estimate the joint MAE (JMAE) for each learning algorithm. It is the sum of the normalised value of the MAE term in each class variable

$$JMAE\epsilon(\psi) = \sum_{j=1}^{m} \frac{1}{|C_j|-1} MAE\epsilon_j(\psi)$$

Note that the JMAE term varies between 0 and m, being m the number of class variables.

The MAE values of the exposed experimentation setup, which have also been estimated via 20 runs of 5-fold non-stratified cross validation [45], are shown in Table 8. It has the same shape as in Table 7, i.e. each row represents each learning algorithm with a specific feature set, and each column represents each class variable and the JMAE value. The best estimated error per classifier is also highlighted in **bold** and Student's t-tests ($\alpha = 0.05$) have been performed in order to study the significance of estimated differences. Table 8 reports conclusions similar to the ones extracted with the accuracy term:

1. The feature set comparison reports the same conclusions to those obtained with the accuracy: the ASOMO feature set significantly outperforms the n-grams and PoS, not only in terms of joint accuracy, but also in terms of two (out of the three) dimensions.
2. The best joint accuracy is given by the multi-dimensional approach which uses the ASOMO feature set and it significantly outperforms both the multi-dimensional approaches with the state-of-the-art feature sets, and the uni-dimensional approaches. With respect to the class variables in isolation, the only case in which the uni-dimensional approaches outperform the multi-dimensional approach is in the Sentiment Polarity target dimension.

In brief and regarding the feature set, for this specific problem, we strongly recommend the use of the ASOMO features, not only because of their performance, but also for their lower learning times. The results also show that the multi-dimensional classification approach to SA is a novel attractive point of view that needs to be taken into account due to the fact that it could lead to better results in terms of accuracy as well as in MAE. In addition, learning a multi-dimensional classifier is faster than learning

Table 8
Estimated mean absolute error rates on the ASOMO dataset using three different types of feature sets in both uni and multi-dimensional scenarios (20 × 5cv).

Approach	Classifier	Feature set	#feat.	Will to influence MAE	Sentiment P. MAE	Subjectivity MAE	JMAE
Uni-dimen.	Multiple classif.	Unigrams	357	†0.632 ± 0.040	†1.048 ± 0.042	†0.211 ± 0.017	†0.684 ± 0.025
		Uni. + bigrams	563	†0.700 ± 0.043	**0.956 ± 0.043**	†0.196 ± 0.014	†0.669 ± 0.023
		PoS	766	†0.878 ± 0.063	†1.147 ± 0.052	†0.339 ± 0.043	†0.918 ± 0.054
		ASOMO feat.	14	0.563 ± 0.014	†1.036 ± 0.036	†0.173 ± 0.011	†0.620 ± 0.014
	Cartesian class	Unigrams	357	N/A	N/A	N/A	†0.650 ± 0.026
		Uni. + bigrams	563	N/A	N/A	N/A	†0.680 ± 0.030
		PoS	766	N/A	N/A	N/A	†0.757 ± 0.040
		ASOMO feat.	14	N/A	N/A	N/A	†0.640 ± 0.060
Multi-dimen.	MDnB	Unigrams	357	†0.788 ± 0.049	†1.104 ± 0.039	†0.247 ± 0.026	†0.789 ± 0.031
		Uni. + bigrams	563	†0.852 ± 0.052	†1.107 ± 0.042	†0.263 ± 0.040	†0.824 ± 0.049
		PoS	766	†0.728 ± 0.044	†1.037 ± 0.040	†0.240 ± 0.020	†0.742 ± 0.029
		ASOMO feat.	14	**0.559 ± 0.029**	†1.019 ± 0.027	**0.167 ± 0.009**	**0.608 ± 0.016**

different classifiers for each dimension. Although the reported times are not a problem in the current supervised learning settings, in the semi-supervised framework, where the computation time increases dramatically with the number of instances, the learning process could be intractable.

7.2.2. *Experiments with the ASOMO corpus in the supervised and semi-supervised learning frameworks*

With the knowledge that the ASOMO features can lead us to better classification rates in this problem, we evaluated their performance in both supervised and semi-supervised frameworks. With this experiment we want to determine if the use of unlabelled examples when learning a classifier can lead to better solutions to the ASOMO problem. In order to do so, the following experiment was performed: the ASOMO dataset has been used to learn three different (uni-dimensional) Bayesian network classifiers and three different sub-families of multi-dimensional classifiers in both frameworks.

For uni-dimensional classification, naive Bayes classifier (*nB*), tree-augmented Bayesian network classifier (*TAN*) [24] and a 2-dependence Bayesian classifier (2 DB) [47] have been chosen. The uni-dimensional approach selected for these experiments is that which consists of splitting the problem into three different uni-dimensional problems (this is because it is more common in the state-of-the-art solutions to solve different problems rather than create an artificial class variable by means of the Cartesian product). From the multi-dimensional aspect, *MDnB*, *MDTAN*, *MD*1/1 and *MD*2/*K* (with *K*=2,3,4) structures have been selected. *MD*1/1 is included as an algorithm able to learn MDTAN structures due to the poor performance shown by the MDTAN learning algorithm [53] in the artificial experiments. Although both multi-dimensional learning approaches learn MDTAN structures, each learning algorithm follows a different path to come to that end. While the MD 1/1 uses a filter approach, the MDTAN learning algorithm follows a wrapper scheme.

The supervised learning procedure only uses the labelled dataset (consisting of 150 documents), whilst the semi-supervised approach uses the 2532 posts (2392 unlabelled). Our multi-dimensional extension of the EM algorithm is used in the latter approach and it terminates after finding a local likelihood maxima or after 250 unsuccessful trials.

Finally, the performance of each model has been estimated via 20 runs of 5-fold non-stratified cross validation. Due to fact that, in semi-supervised learning, the labels of the unlabelled subset of instances are unknown, only the labelled subset is divided into five folds to estimate the performance of the proposed approaches. So, in each iteration of the cross validation, a classifier is learnt with four labelled folds and the whole unlabelled subset, and then it is tested in the remaining labelled fold. This modified cross validation is illustrated in Fig. 8 for the case of three folds. As done in the previous experiments, we use the accuracy and the MAE terms as evaluation measures.

Table 9 shows the results of applying the different uni-dimensional and multi-dimensional algorithms over the ASOMO dataset in terms of accuracy and Table 10 in terms of MAE. Both tables can be described as follows: for each classifier (row), the joint performance and the single evaluation measure (for each class variable) are shown. In order to simultaneously compare uni-dimensional with respect to multi-dimensional approaches, and supervised with respect to semi-supervised learning, the results are highlighted as follows:

1. In order to compare the supervised and the semi-supervised frameworks, for each type of classifier and accuracy measure (class variable and joint performance), we have highlighted the best single value and joint performance in **bold** (analysed per row). Note that, in the case of the accuracy term (Table 9), the highlighted values are the greatest values, while in the

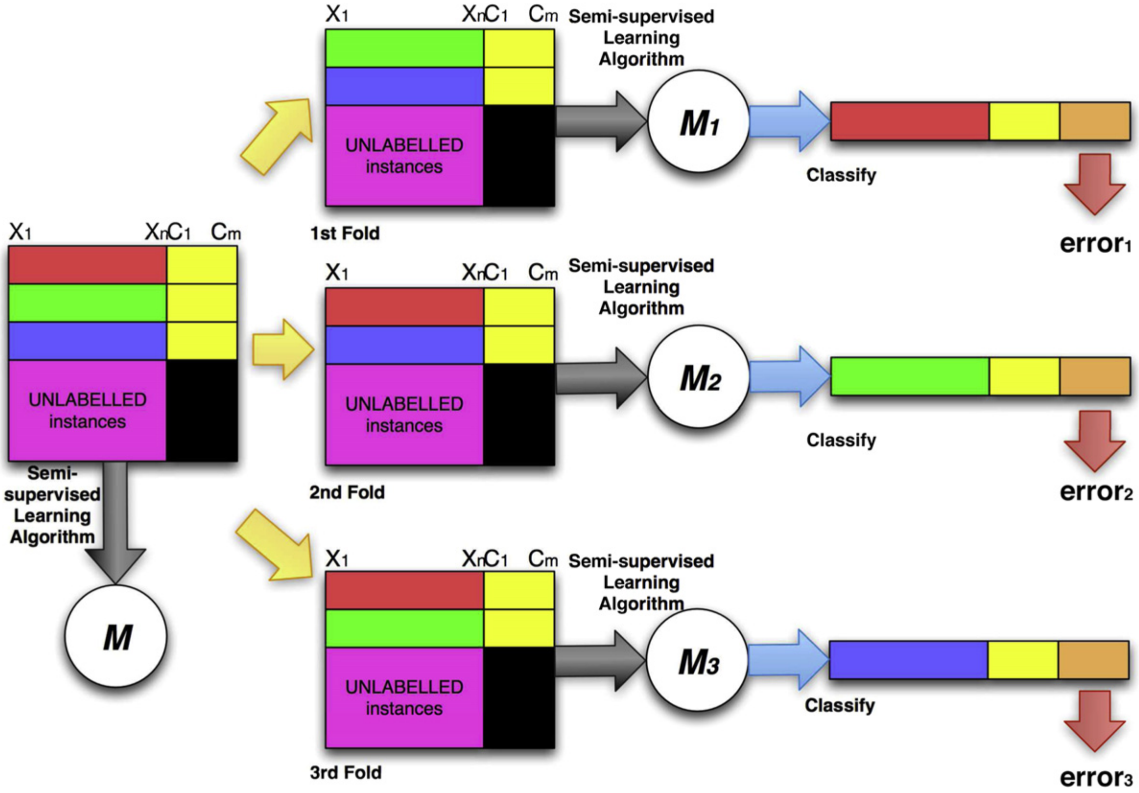


Fig. 8. Applying 3-fold cross validation to a dataset with labelled and unlabelled instances.

Table 9
Accuracies on the ASOMO dataset with the ASOMO features in the supervised and the semi-supervised learning frameworks (20 × 5cv).

Classif.	Labelled data (supervised learning)				Labelled + unlabelled data (semi-supervised learning)			
	W. influence Acc	Sentiment P. Acc	Subjectivity Acc	Joint Acc	W. influence Acc	Sentiment P. Acc	Subjectivity Acc	Joint Acc
nB	†‡55.07 ± 1.32	‡**26.07 ± 2.09**	‡**82.17 ± 1.35**	‡**10.37 ± 2.30**	***56.70 ± 2.85***	†‡21.97 ± 3.11	†‡58.07 ± 3.78	†‡7.43 ± 2.27
TAN	‡**52.93 ± 3.23**	†‡26.30 ± 2.16	‡**80.10 ± 1.46**	‡9.47 ± 1.29	†‡48.77 ± 3.44	‡**29.00 ± 3.69**	†‡75.10 ± 3.42	‡**9.87 ± 1.79**
2DB	***57.13 ± 1.85***	‡**27.43 ± 2.82**	‡**82.30 ± 1.57**	‡**13.17 ± 1.59**	†‡54.13 ± 3.37	†‡24.80 ± 2.95	†‡61.30 ± 4.83	†‡8.60 ± 2.08
MDnB	‡53.23 ± 1.62	†*30.87 ± 2.52*	*83.53 ± 0.69*	†14.97 ± 1.94	‡**54.00 ± 2.08**	***32.27 ± 1.41***	***83.63 ± 0.55***	***16.83 ± 1.14***
MDTAN	‡**52.60 ± 3.48**	†‡27.13 ± 2.43	‡82.57 ± 1.72	‡**12.80 ± 2.38**	†‡31.10 ± 4.37	‡**29.43 ± 2.48**	‡**82.60 ± 1.37**	†‡8.97 ± 1.70
MD 1/1	**56.67 ± 2.93**	**29.97 ± 3.75**	**78.17 ± 2.59**	**15.63 ± 2.00**	†‡53.27 ± 2.78	‡29.17 ± 2.54	‡77.90 ± 2.20	‡15.33 ± 1.37
MD 2/2	**56.60 ± 3.63**	‡**28.93 ± 2.36**	‡**77.47 ± 2.24**	15.17 ± 1.99	†‡53.30 ± 2.08	‡28.77 ± 2.37	‡76.93 ± 2.97	‡**15.50 ± 0.94**
MD 2/3	**56.70 ± 2.87**	29.87 ± 3.20	‡**77.03 ± 1.41**	*15.90 ± 2.67*	†‡52.77 ± 1.53	‡**30.77 ± 2.25**	‡77.90 ± 2.20	**16.63 ± 1.32**
MD 2/4	**56.97 ± 2.11**	‡28.53 ± 3.13	‡**76.87 ± 2.39**	**15.57 ± 2.41**	†‡52.47 ± 2.05	**29.30 ± 3.02**	‡75.27 ± 3.24	‡15.43 ± 1.28

Table 10
Mean absolute error rates on the ASOMO dataset with the ASOMO features in the supervised and the semi-supervised learning frameworks (20 × 5cv).

Classif.	Labelled data (supervised learning)				Labelled + unlabelled data (semi-supervised learning)			
	W. influence MAE	Senti. P. MAE	Subject. MAE	JMAE	W. influence MAE	Senti. P. MAE	Subject. MAE	JMAE
nB	‡**0.563 ± 0.014**	**1.036 ± 0.036**	**0.173 ± 0.011**	‡**0.620 ± 0.014**	†‡0.613 ± 0.040	†‡1.209 ± 0.069	†‡0.421 ± 0.036	†‡0.986 ± 0.041
TAN	‡**0.614 ± 0.041**	†‡1.044 ± 0.043	‡**0.204 ± 0.015**	‡**0.670 ± 0.029**	†‡0.664 ± 0.055	***0.991 ± 0.041***	†‡0.225 ± 0.038	†‡0.718 ± 0.039
2DB	‡**0.553 ± 0.028**	**1.035 ± 0.053**	**0.171 ± 0.009**	**0.614 ± 0.018**	†‡0.636 ± 0.041	†‡1.114 ± 0.082	†‡0.387 ± 0.048	†‡0.885 ± 0.056
MDnB	‡0.559 ± 0.029	*1.019 ± 0.027*	†*0.167 ± 0.009*	†*0.608 ± 0.016*	***0.549 ± 0.019***	**1.002 ± 0.028**	***0.612 ± 0.005***	***0.596 ± 0.007***
MDTAN	‡**0.567 ± 0.045**	‡**1.101 ± 0.052**	‡0.180 ± 0.015	‡**0.644 ± 0.026**	†‡0.878 ± 0.075	‡1.102 ± 0.028	‡**0.172 ± 0.011**	†‡0.750 ± 0.027
MD 1/1	**0.529 ± 0.034**	‡**1.056 ± 0.032**	‡0.219 ± 0.022	‡**0.659 ± 0.029**	†0.556 ± 0.031	‡1.061 ± 0.054	‡**0.216 ± 0.018**	‡0.666 ± 0.015
MD 2/2	***0.525 ± 0.033***	‡**1.057 ± 0.054**	‡**0.222 ± 0.021**	‡**0.660 ± 0.034**	†0.560 ± 0.040	‡1.075 ± 0.050	‡0.227 ± 0.023	†‡0.682 ± 0.022
MD 2/3	**0.531 ± 0.032**	‡**1.061 ± 0.032**	‡0.237 ± 0.034	‡0.679 ± 0.037	†*0.549 ± 0.016*	‡1.048 ± 0.049	‡**0.223 ± 0.033**	‡**0.678 ± 0.033**
MD 2/4	**0.529 ± 0.041**	‡**1.069 ± 0.058**	‡0.227 ± 0.020	‡**0.670 ± 0.031**	†‡0.579 ± 0.038	‡1.047 ± 0.038	‡**0.225 ± 0.023**	‡0.680 ± 0.021

case of the MAE (Table 10) they are the lowest. Pairwise statistical hypothesis tests (Student's t-test with $\alpha = 0.05$) have been performed in order to determine if there are significant statistical differences between the values of the tested techniques. We use the symbol '†' to mark the values that are statistically outperformed by the highlighted best estimated measure (in **bold**).

2. To compare the performance between the uni-dimensional and the multi-dimensional approaches, the best accuracy per class variable, as well as the joint performance, has been highlighted in *italics*. For each column, statistical hypothesis tests (Student's t-test with $\alpha = 0.05$) have been performed in order to determine if there are significant statistical differences. The symbol '‡' is used to mark the values that are statistically worse than the best estimated value (in *italics*).

Several conclusions can be extracted from the supervised and semi-supervised comparison in Tables 9 and 10 (analysed per row):

1. The uni-dimensional models tend to perform worse when they are learnt using the semi-supervised learning framework. This could be due to the fact that incorrect models tend to lead to performance degradation in the semi-supervised framework due to the fact that they are not able to match the underlying generative structure [12]. This phenomenon occurs in both evaluation measures.
2. As occurs in the artificial experiments, the MDTAN approach [53] tends to behave more similarly to the uni-dimensional approaches rather than to the multi-dimensional approaches.
3. With respect to the accuracy measure, in the multi-dimensional scenario, the Will to Influence class variable tends to degrade its performance in the semi-supervised scenario, whilst Sentiment Polarity and Subjectivity tend to achieve better single accuracies. In the uni-dimensional approach, the opposite happens.
4. The MAE results show that, unlike what happens in the uni-dimensional framework where the semi-supervised degradation is significant, in the multi-dimensional scenario similar results are reported for supervised and semi-supervised learning. However, there are cases in which the semi-supervised learning algorithms obtain better results and in one case there is a significant statistical gain.
5. The MDnB method in its semi-supervised framework is the best solution for the ASOMO problem. In terms of accuracy, it obtains statistically significant better results in joint accuracy and in the Sentiment Polarity target variable, as well as better results in the other two variables. With respect to the MAE, MDnB obtains statistically significant better results in joint accuracy and in the Subjectivity dimension, as well as better results in the other dimensions.

Regarding the comparison of the uni-dimensional and multi-dimensional approaches (for each column), the following comments can be extracted:

1. With the exception of Will to Influence, the single class variables tend to achieve better accuracies in the multi-dimensional approaches.
2. The MAE terms show similar results to those found with the accuracy evaluation measure. However, in this case, the exception is provided by the estimated MAE obtained in the Sentiment Polarity dimension in the semi-supervised version of the uni-dimensional TAN algorithm.

3. The multi-dimensional classification approach statistically outperforms the uni-dimensional framework in terms of joint accuracy.
4. MDnB is also the best technique in terms of global performance, i.e. in accuracy and in MAE metrics.

One explanation for the surprising success of the MDnB could be the use of the knowledge of the experts in engineering the features, as stated in [28]: when applying rational criteria in determining the predictive features of a problem, the resulting features are usually probabilistically independent given the class variables. This characteristic favours the learning scheme provided by the MDnB algorithm. Moreover, this is crucial in the success obtained in the semi-supervised framework due to the fact that it matches the actual underlying domain structure [12].

The MDnB assumes conditional independence between the three class variables. In spite of this assumption, we cannot talk about independence between the class variables as happens when the problem is approached by several uni-dimensional classifiers. Each class variable in this model uses the information of the remaining class variables when it carries out the classification task. It also simultaneously uses all the class variables to learn the parameters of the structure. The success of this algorithm can shed some light on the relation between the three variables used in this problem: the multi-dimensional framework achieves better results than the uni-dimensional counterparts. Therefore, it seems that there is a certain relation between them. Furthermore, it can be seen that simultaneously using the information of these class variables in the same classification task by means of a multi-dimensional technique conceives better predictive classifiers.

In conclusion, we show that the proposed semi-supervised multi-dimensional formulation designs a novel perspective for this kind of SA problems, opening new ways to deal with this domain. In addition, it can also be seen that the explicit use of the different class variables in the same classification task has successfully solved the ASOMO problem, where the MDnB in a semi-supervised framework is the best solution.

In conclusion, we show that the proposed semi-supervised multi-dimensional formulation designs a novel perspective for this kind of SA problems, opening new ways to deal with this domain. In addition, it can also be seen that the explicit use of the different class variables in the same classification task has successfully solved the ASOMO problem, where the MDnB in a semi-supervised framework is the best solution.

8. Conclusions and future work

In this paper, we solve a real-world multi-dimensional SA problem. This real problem consists of characterising the attitude of a customer when he writes a post about a particular topic in a specific forum through three differently related dimensions: Will to Influence, Polarity and Subjectivity.

Due to the fact that it has three different target variables, the SA (uni-dimensional) state-of-the-art classification techniques seem inappropriate. They do not match the underlying multi-dimensional nature of this problem. The problem also cannot be directly tackled by multi-label techniques. For that reason, we propose the use of multi-dimensional Bayesian network classifiers as a novel methodological tool which joins the different target variables in the same classification task in order to exploit the potential relationships between them. Within this methodology, in this paper, we have proposed a new filter algorithm to learn multi-dimensional *J/K* dependences Bayesian network structures in order to explore a wider range of structures while dealing with this application.

Moreover, in order to avoid the arduous and time-consuming task of labelling examples in this field, we extend, by means of the EM algorithm, these multi-dimensional techniques to semi-supervised learning framework so as to make use of the huge amount of unlabelled data available on the Internet.

Experimental results of applying the proposed battery of multi-dimensional learning algorithms to a corpus consisting of 2542 posts (150 manually labelled and 2392 unlabelled) show that: (1) the uni-dimensional approaches cannot capture the multi-dimensional underlying nature of this problem, (2) engineering a suitable feature set is a key factor for obtaining better solutions, (3) more accurate classifiers can be found using the multi-dimensional approaches which perform a simultaneous classification task, (4) the use of large amounts of unlabelled data in a semi-supervised framework can be beneficial to improve the recognition rates, and (5) the MDnB classifier in a semi-supervised framework is the best solution for this problem because it matches the actual underlying domain structure [28,12].

The proposed multi-dimensional methodology can be improved or extended in several ways. For instance, in the ASOMO multi-dimensional problem, the values of all class variables are missing in each sample of the unlabelled subset. However, by means of the EM algorithm, the learning algorithms can be easily generalised to the situation where not all the class variables are missing in all the samples of the unlabelled data subset.

Besides, we are concerned about the scalability of the multi-dimensional Bayesian network classifiers in the semi-supervised framework. The computational burden is not a problem when dealing with these datasets, but it could happen when the number of variables increases. This could open a line in researching feature subset selection techniques for multi-dimensional classification.

Regarding the application of multi-dimensional classification to the SA domain, this work can be extended in a number of different ways:

- The proposed multi-dimensional Bayesian network classifiers can be directly applied to Affect Analysis. This area is concerned with the analysis of text containing emotions and it is associated with SA [1]. However, Affect Analysis tries to extract a large number of potential emotions, e.g. happiness, sadness, anger, hate, violence, excitement, fear, etc, instead of just looking at the polarity of the text. Additionally, in the case of Affect Analysis, the emotions are not mutually exclusive and certain emotions may be correlated. So, this can easily be viewed as a multi-label classification problem, a type of problem in which multi-dimensional Bayesian network classifiers have reported good results in the recent past [5].
- Within SA, the same corpus can be used to deal with different target dimensions. This could open different research lines in adding more target variables in the same classification task so as to take advantage of these existing relationships, engineering a suitable feature set for working with several dimensions, etc. For instance, in the works where the need to predict both the Sentiment Polarity and the Subjectivity has been noticed [21].

Acknowledgements

This work has been partially supported by the Saiotek, Etortek and Research Groups 2007–2012 (IT-242-07) programs (Basque Government), TIN2008-06815-C02-01, TIN2010-14931, Consolider Ingenio 2010-CSD2007-00018 projects and MEC-FPU Grant AP2008-00766 (Spanish Ministry of Science and Innovation), and COMBIOMED network in computational bio-medicine (Carlos III Health Institute).

References

[1] A. Abbasi, H. Chen, S. Thoms, T. Fu, Affect analysis of web forums and blogs using correlation ensembles, IEEE Trans. Knowledge Data Eng. 20 (9) (2008) 1168–1180.

[2] S. Argamon, C. Whitelaw, P. Chase, S. Raj Hota, N. Garg, S. Levitan, Stylistic text classification using functional lexical features, J. Am. Soc. Inf. Sci. Technol. 58 (6) (2007) 802–822.

[3] J. Atserias, B. Casas, E. Comelles, M. Gonzalez, L. Padro, M. Padro, Freeling 1.3: Syntactic and semantic services in an open-source NLP library, in: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06), 2006, pp. 48–55.

[4] G. Baklr, B. Taskar, T. Hofmann, B. Scholkopf, A. Smola, S.V.N. Vishwanathan, Predicting Structured Data, MIT Press, 2007.

[5] C. Bielza, G. Li, P. Larrañaga, Multi-dimensional classification with Bayesian networks, Int. J. Approx. Reason. 52 (6) (2011) 705–727.

[6] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.

[7] R. Blanco, Learning Bayesian Networks from Data with Factorisation and Classification Purposes. Applications in Biomedicine. PhD Thesis, University of the Basque Country, 2005.

[8] X. Carreras, I. Chao, L. Padro, M. Padro, An open-source suite of language analyzers, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation, vol. 10, 2006, pp. 239–242.

[9] R. Caruana, Multitask learning, Machine Learning 28 (1997) 41–75.

[10] O. Chapelle, B. Scholkopf, A. Zien, Semi-Supervised Learning, The MIT Press, 2006.

[11] C.I. Chow, S. Member, C.N. Liu, Approximating discrete probability distributions with dependence trees, IEEE Trans. Inf. Theory 14 (1968) 462–467.

[12] I. Cohen, Semisupervised Learning of Classifiers with Application to Human–Computer Interaction, PhD thesis, University of Illinois at Urbana-Champaign, 2003.

[13] I. Cohen, F.G. Cozman, A. Bronstein, The effect of unlabeled data on generative classifiers, with application to model selection, in: Proceedings of the SPIE 93 Conference on Geometric Methods in Computer Vision, 2002.

[14] I. Cohen, F.G. Cozman, N. Sebe, M.C. Cirelo, T.S. Huang, Semisupervised learning of classifiers: theory, algorithms and their application to human–computer interaction, IEEE Trans. Pattern Anal. Mach. Intell. 26 (12) (2004) 1553–1567.

[15] G. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (1992) 309–347.

[16] H. Daumé, D. Marcu, Learning as search optimization: approximate large margin methods for structured prediction, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 169–176.

[17] P.R. de Waal, L.C. van der Gaag, Inference and learning multi-dimensional Bayesian network classifiers, Lect. Notes Artif. Intell. 4724 (2007) 501–511.

[18] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Series B (Methodological) 39 (1) (1977) 1–38.

[19] S. Dumais, Hierarchical classification of web content, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, pp. 256–263.

[20] E. Boiy, M.F. Moens, A machine learning approach to sentiment analysis in multilingual web text, Inf. Retrieval 12 (5) (2009) 526–558.

[21] A. Esuli, F. Sebastiani, Sentiwordnet: a publicly available lexical resource for opinion mining, in: Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC06), 2006, pp. 417–422.

[22] G. Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. 3 (2003) 1289–1305.

[23] N. Friedman, The Bayesian structural EM algorithm, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1998, pp. 129–138.

[24] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (2–3) (1997) 131–163.

[25] J.H. George, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Machine Learning, Proceedings of the Eleventh International Conference, 1994, pp. 121–129.

[26] H. Hartley, Maximum likelihood estimation from incomplete data, Biometrics 14 (1958) 174–194.

[27] V. Hatzivassiloglou, J.M. Wiebe, Effects of adjective orientation and gradability on sentence subjectivity, in: Proceedings of the 18th Conference on Computational linguistics, 2000, pp. 299–305.

[28] R. Kohavi, Wrappers for Performance Enhancement and Oblivious Decision Graphs. PhD Thesis, Stanford University, 1995.

[29] J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Am. Math. Soc. 7 (1) (1956) 48–50.

[30] S. Kullback, Information Theory and Statistics, Wiley, 1959.

[31] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: Proceedings of the 10th National Conference on Artificial Intelligence, 1992, pp. 223–228.

[32] P. Larrañaga, J.A. Lozano, J.M. Peña, I. Inza, Special issue on probabilistic graphical models for classification, Mach. Learn. 59 (3) (2005).

[33] B. Liu, Sentiment analysis and subjectivity, in: N. Indurkhya, F.J. Damerau (Eds.), Handbook of Natural Language Processing, second ed., Chapman & Hall, 2010.

[34] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley-Interscience, 1997.

[35] V. Ng, S. Dasgupta, S.M.N. Arifin, Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews, in: Proceedings of the COLING/ACL Conference, 2006, pp. 611–618.

[36] K. Nigam, A. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2) (2000) 103–134.

[37] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (1–2) (2008) 1–135.

[38] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'02), 2002, pp. 79–86.

[39] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers Inc., 1988.

[40] M. Qazi, G. Fung, S. Krishnan, R. Rosales, H. Steck, R.B. Rao, D. Poldermans, D. Chandrasekaran, Automated heart wall motion abnormality detection from ultrasound images using Bayesian networks, in: IJCAI'07: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2007, pp. 519–525.

[41] E. Riloff, J. Wiebe, Learning extraction patterns for subjective expressions, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'03), 2003, pp. 105–112.

[42] E. Riloff, J. Wiebe, T. Wilson, Learning subjective nouns using extraction pattern bootstrapping, in: Proceedings of the Seventh Conference on Natural Language Learning, 2003, pp. 25–32.

[43] J.D. Rodríguez, J.A. Lozano, Multi-objective learning of multi-dimensional Bayesian classifiers, in: Eighth Conference on Hybrid Intelligent Systems (HIS'08), September 2008, pp. 501–506.

[44] J.D. Rodríguez, J.A. Lozano, Learning Bayesian Network Classifiers for Multi-dimensional Supervised Classification Problems by Means of a Multi-objective Approach. Technical Report EHU-KZAA-TR-3-2010, Department of Computer Science and Artificial Intelligence, University of the Basque Country, San Sebastián, Spain, 2010.

[45] J.D. Rodriguez, A. Perez, J.A. Lozano, Sensitivity analysis of k-fold cross-validation in prediction error estimation, IEEE Trans. Pattern Anal. Mach. Intell. 32 (3) (2010) 569–574.

[46] C. Rose, Y.C. Wang, Y. Cui, J. Arguello, K. Stegmann, A. Weinberger, F. Fischer, Analyzing collaborative learning processes automatically: exploiting the advances of computational linguistics in computer-supported collaborative learning, Int. J. Computer-Supported Collab. Learn. 3 (3) (2007).

[47] M. Sahami, Learning limited dependence Bayesian classifiers, in: AAAI Press (Ed.), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996, pp. 335–338.

[48] B. Snyder, R. Barzilay, Multiple aspect ranking using the good grief algorithm, in: Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL), 2007, pp. 300–307.

[49] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. R. Stat. Soc. B 36 (1) (1974) 111–147.

[50] D.M.J. Tax, R.P.W Duin, Using two-class classifiers for multiclass classification, in: Proceedings of 16th International Conference on Pattern Recognition, 2002, pp. 124–127.

[51] G. Tsoumakas, I. Katakis, Multi label classification: an overview, Int. J. Data Warehousing Mining 3 (3) (2007) 1–13.

[52] O. Tsur, D. Davidiv, A. Rappoport, ICWSM a great catchy name: semi-supervised recognition of sarcastic sentences in product reviews, in: Proceedings of the International AAAI Conference on Weblogs and Social Media—ICWSM10, 2010.

[53] L.C. van der Gaag, P.R. de Waal, Multi-dimensional Bayesian network classifiers, in: Proceedings of the Third European Workshop in Probabilistic Graphical Models, 2006, pp. 107–114.

[54] J. Wiebe, T. Wilson, R. Bruce, N. Bell, M. Martin, Learning subjective language, Comput. Ling. 30 (3) (2004) 277–308.

[55] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, in: Morgan Kaufmann Series in Data Management Systems, second ed., Morgan Kaufmann, 2005.

[56] H. Yu, V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2003.

[57] X. Zhu, Semi-supervised Learning Literature Survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison, 2005.

[58] X. Zhu, A.B. Goldberg, Introduction to Semi-Supervised Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2009.

Jonathan Ortigosa-Hernández received an MSc degree in Computer Science from *The University of the Basque Country*, Donostia-San Sebastián, Spain in 2008, and a BSc in Informatics from *Coventry University*, United Kingdom, in the same year. In 2009, he received a FPU fellowship from the Spanish Ministry of Education. Since then, he has been a PhD student in *The University of the Basque Country* and a member of the *Intelligent Systems Group* research team. His research interests are multi-dimensional classification, semi-supervised learning, and sentiment and affect analysis. http://www.sc.ehu.es/ccwbayes/members/jonathan

Juan Diego Rodríguez received his MSc in Computer Science from *The University of the Basque Country* in 2002. At present, he is a PhD student and a member of the *Intelligent Systems Group*. His research interests include evaluation techniques on supervised classification, Bayesian networks and multi-dimensional Bayesian classification. He has been published in top refereed machine learning journals such as IEEE TPAMI. http://www.sc.ehu.es/ccwbayes/members/juandiego/

Leandro Alzate received a BA in Arts from *The University of the Basque Country*, Bilbao, Spain in 1997. In 2003, he received an MBA in Business Intelligence from *Universitat Oberta de Catalunya*, Barcelona, Spain. He is one of the founders of *Socialware*©, where he worked in the areas of Product Innovation and R+D+I from 2000 to 2010. His main interests are NLP and sentiment analysis.

Manuel Lucania received his MSc degree in Computer Science Engineering from *ICAI University (Universidad Pontificia Comillas)*, Madrid, Spain in 2007. From 2006 to 2008, he worked as an IT consultant in network security at *Hewlett–Packard*, Madrid. After that, between 2008 and 2009, he worked in *Comunicaciones Galileo*, Madrid, developing a syntactic and semantic analyser for the Spanish language. Since 2009, he has led the ASOMOs R&D Natural Language Processing and Text Mining projects for sentiment analysis and reputation monitoring at *Socialware*©. His main interests are NLP, advanced Sentiment Analysis based on the psychology theories of Plutchik, linguistic problem-solving, web crawling, data analysis and algorithm design.

Iñaki Inza received his PhD in computer science from *The University of the Basque Country* in 2002 and he is a member of the *Intelligent Systems Group* research team. He is an associate professor in *The University of the Basque Country*. Some of his main interests are in feature selection methods, Bayesian networks and applications to biological domains. He has eight book chapters in six books and 31 publications in refereed journals and conferences, some of them in top machine learning and biology journals such as Bioinformatics, Machine Learning, IEEE TPAMI or Artificial Intelligence in Medicine. www.sc.ehu.es/ccwbayes/members/inaki.htm

Jose A. Lozano received an MSc degree in mathematics and an MSc degree in computer science from the *The University of the Basque Country*, Spain, in 1991 and 1992 respectively, and a PhD degree in computer science from the *University of the Basque Country*, Spain, in 1998. Since 2008 he is s full professor in the Department of Computer Science and Artificial Intelligence in *The University of the Basque Country*, where he leads the *Intelligent Systems Group*. He is the co-author of more than 50 ISI journal publications and co-editor of the first book published about Estimation of Distribution Algorithms. His major research interests include machine learning, pattern analysis, evolutionary computation, data mining, metaheuristic algorithms, and real-world applications. Prof. Lozano is associate editor of IEEE Transactions on Evolutionary Computation and a member of the editorial board of Evolutionary Computation journal, Soft Computing and another three journals. www.sc.ehu.es/ccwbayes/members/jalozano

Appendices for the article entitled "*Approaching Sentiment Analysis by Using Semi-supervised Learning of Multi-dimensional Classifiers*"

Jonathan Ortigosa-Hernández, Iñaki Inza, and Jose A. Lozano

Contents

1 Appendix A. Artificial Experiments **1**
1.1 Simulation of datasets 2
1.2 Experimental Setup 3
1.3 Results 3
1.3.1 Uni-dimensional and multi-dimensional learning algorithms comparison 7
1.3.2 Learning the generative structure 8
1.3.3 Reaching the generative structure 9
1.3.4 Supervised and semi-supervised learning frameworks comparison . . 10
1.3.5 Behaviour of the learning algorithms in the semi-supervised framework 12
1.4 General conclusions 12

1 Appendix A. Artificial Experiments

In this report, we execute the proposed battery of semi-supervised multi-dimensional learning algorithms over a set of designed artificial datasets as commonly carried out in the machine learning research community. These experiments are performed in order to demonstrate that the proposed algorithms are able to take advantage of the underlying nature of the multi-dimensional problems even in the presence of a small set of labelled data and a huge set of unlabelled data. By means of these experiments, we would like to shed some light on the following questions:

1. Are there significant differences between the uni-dimensional and the multi-dimensional supervised learning algorithms when there is a scarcity of labelled examples? Are there significant differences between the uni-dimensional and the multi-dimensional semi-supervised learning algorithms?

2. If the correct structure of the generative model is obtained, do unlabelled data improve the classifier?

3. If the learning algorithm can lead to the correct structure of the generative model, do unlabelled data improve the classifier?

4. Can adding unlabelled data contribute to an increase in the classification performance (in terms of joint accuracy) when there is a small amount of labelled data in a multi-dimensional framework?

5. Do multi-dimensional classifiers performs better than uni-dimensional classifiers in a multi-dimensional semi-supervised framework?

1.1 Simulation of datasets

In order to carry out the experimentation process required to evaluate our proposals, we use a set of artificial multi-dimensional datasets. These datasets are sampled from multi-dimensional feature-class variable probability distributions $p(\mathbf{x}, \mathbf{c})$ represented as multi-dimensional Bayesian network classifiers. These classifiers have been created in four steps: First, the structure of the multi-dimensional Bayesian network classifier was created. Second, the parameters of the classifiers were obtained by sampling a Dirichlet distribution. Third, a dataset from each classifier was sampled. Finally, the semi-supervised nature of the sampled dataset, i.e. both subsets of labelled and unlabelled data, was generated by choosing instances at random for both types of data subsets. The entire process of creating the artificial datasets has been performed by means of the free software ICLAB library [1].

The following sub-families of multi-dimensional Bayesian network structures have been chosen for this experimentation: *MDnB*, *MDTAN*, *MD* 2/2 and *MD* 2/3. A number of 5 different structures per each sub-family with a different number of features (from 5 to 20) and a different number of class variables (from 2 to 4) have randomly been created. The cardinality of the features ranges from 2 to 4 and the cardinality of the class variables from 2 to 3. By means of these sub-families and structures we are trying to cover a broad range of structures of different complexity in order to check the statement "performance degradation may occur whenever the modelling assumptions adopted for a particular classifier do not match the characteristics of the distribution generating the data [5]". In order to generate the parameters of the classifiers which are defined by the previous structures, a different Dirichlet distribution with all its parameters equal to one is sampled per each classifier. The associated optimal Bayes accuracies (the opposite to Bayes errors) of the resulting classifiers can be seen in Table 1, measured as percentages.

Once the multi-dimensional Bayesian network classifiers have been constructed, a dataset from each of them is sampled. Specifically, we sample 20 artificial datasets (5 for each different sub-family) of $15,100$ instances. Then, we divide each dataset into two different parts: a training set of $10,100$ instances used to learn the classifiers, and a test set of $5,000$ samples used to estimate the prediction error of the classifiers. In order to simulate the semi-supervised nature of the training dataset, 100 instances were chosen at random to form the subset D_L, in which labelled instances are i.i.d. as in $p(\mathbf{x}, \mathbf{c})$. The labels of the class variables of the remaining $10,000$ instances are removed.

All the datasets, as well as the structures and the parameters of the designed classifiers, can be found in the following website[1].

1.2 Experimental Setup

Once the semi-supervised multi-dimensional datasets are created, we use them to perform the following set of experiments. In the evaluation phase, we proposed nine different algorithms to learn the classifiers from the sampled datasets. The first four algorithms are the approaches explicitly designed for multi-dimensional classification proposed in the paper: *MDnB*, *MDTAN*, *MD* 1/1. *MD* 2/2 and *MD* 2/3. Due to the fact that MDTAN learning algorithm [13] follows a wrapper approach, the MD 1/1 is included in these experiments as a filter approach in order to establish a comparison between both techniques. The others are the well-known uni-dimensional approaches: naive Bayes classifier (*nB*), tree-augmented network classifier (*TAN*) [8] and two K dependence Bayesian classifiers [12] (one setting $K = 2$, 2-*DB*, and the other setting $K = 3$, 3-*DB*). As stated in the article, the uni-dimensional approach cannot be straightforwardly applied to deal with the multi-dimensional problems, so, we divided the multi-dimensional problem into several one-class variable tasks and tackled them as independent.

In order to compare the supervised and the semi-supervised frameworks, all the algorithms are learnt in both scenarios. In the case of supervised learning the algorithms are straightforwardly applied to the 100 instances of the labelled subset. When learning in the semi-supervised framework, on the contrary, the algorithms are used, as proposed in this work, in conjunction with our extension of the EM algorithm (Algorithm 2 of the article) and applied to the whole training dataset (100 labelled + $10,000$ unlabelled). The EM algorithm terminates after finding a local likelihood maxima or after 250 unsuccessful trials. The parameters of the model are calculated by maximum likelihood estimation (MLE) [4], corrected with Laplace smoothing. The estimation method for performance evaluation metrics is hold-out [10]. In hold-out, a subset of instances is chosen randomly from the initial dataset to form a training set used to learn a classifier, and the remaining instances are retained as the testing data, used to estimate the error of the classifier. This method has been chosen in order to evaluate all the algorithms in the same testing set, avoiding the variance of the error estimation given by the cross-validation methods [11]. Finally, the performance evaluation is performed by the joint evaluation criteria.

1.3 Results

Table 1 shows the results of the nine algorithms over the 20 datasets when they are applied in the supervised scenario. Table 2 shows, instead, their results in the semi-supervised

[1] http://www.sc.ehu.es/ccwbayes/members/jonathan/home/ISG/News_and_Notables/Entries/2010/11/30_IMACS_2011.html

framework. Each value in both tables corresponds to the joint accuracy obtained for each algorithm when the learnt classifier is evaluated in the testing set of $5,000$ samples. The accuracies in **bold** (per row) correspond to the technique with the best accuracy in just the labelled dataset (Table 1), and the best in the whole dataset (Table 2). Moreover, the best joint accuracy per dataset (per row in Tables 1 and 2) is highlighted in ***bold italics***.

In order to answer the questions proposed in the introduction of this report, we have performed an exhaustive analysis of the results of Tables 1 and 2. The following sections summarise the studies made to shed some light on the questions. In each section, the conclusions that answer these questions have been underlined.

LABELLED DATA											
Family	Dataset	$(1 - e_B)$	nB	TAN	2DB	3DB	MDnB	MDTAN	MD 1/1	MD2/2	MD2/3
MDnB	NB01	*93.93*	77.06	80.20	75.68	74.38	**88.48**	66.22	77.38	76.40	75.96
	NB02	*93.25*	76.96	80.80	79.10	78.40	**89.90**	71.58	82.14	76.62	75.24
	NB03	*91.65*	76.98	79.94	76.86	75.54	**88.29**	68.43	79.72	75.82	73.86
	NB04	*74.36*	43.71	48.81	40.35	42.83	**65.57**	39.59	40.85	43.43	47.38
	NB05	*66.45*	40.20	43.92	37.19	38.55	**59.44**	44.64	40.71	37.99	34.82
MDTAN	TA01	*66.86*	57.94	***61.01***	56.14	47.86	53.53	57.86	58.18	57.98	54.36
	TA02	*47.61*	31.13	33.61	28.97	29.29	29.75	34.77	**37.46**	36.36	36.04
	TA03	*56.80*	40.59	46.36	39.35	37.74	37.56	39.12	**46.60**	44.26	44.38
	TA04	*64.21*	43.33	46.43	43.87	36.40	43.31	41.85	**49.26**	48.14	44.01
	TA05	*58.75*	31.78	41.70	28.81	30.85	27.85	42.03	45.87	45.31	**45.89**
MD2/2	2201	*66.26*	56.04	**58.64**	53.48	46.46	54.48	56.50	57.46	55.32	53.84
	2202	*65.44*	45.01	42.29	41.55	36.39	39.19	**46.11**	38.41	44.89	43.01
	2203	*61.72*	44.92	40.52	39.80	39.96	44.82	41.80	**48.06**	47.36	47.90
	2204	*84.85*	80.86	79.98	80.02	81.22	81.47	**83.96**	81.22	82.91	83.45
	2205	*50.28*	37.98	34.37	33.41	29.65	33.83	***43.80***	40.72	39.62	39.34
MD2/3	2301	*57.56*	**43.38**	40.93	39.40	37.80	39.16	38.82	41.84	41.36	42.90
	2302	*69.11*	61.38	57.10	60.52	55.14	56.24	55.70	60.54	**61.44**	59.88
	2303	*50.31*	29.10	34.35	28.98	27.04	28.58	31.06	***34.41***	31.22	33.41
	2304	*78.45*	67.13	68.41	67.27	61.63	64.09	***73.85***	71.91	72.15	71.33
	2305	*61.34*	48.82	47.72	46.40	49.34	47.48	**56.12**	54.24	51.92	54.48

Table 1: Estimated accuracies of the proposed algorithms in the supervised scenario.

LABELLED AND UNLABELLED DATA											
Family	Dataset	$(1 - e_B)$	nB	TAN	2DB	3DB	MDnB	MDTAN	MD 1/1	MD2/2	MD2/3
MDnB	NB01	*93.93*	49.24	50.12	49.88	49.98	***92.16***	50.86	83.52	86.16	81.22
	NB02	*93.25*	41.30	42.26	41.34	41.58	***91.06***	41.94	87.32	84.06	83.78
	NB03	*91.65*	41.69	64.05	42.65	43.09	***89.09***	55.68	81.31	83.75	84.41
	NB04	*74.36*	3.62	30.66	14.17	24.08	***67.63***	26.72	48.94	50.24	48.64
	NB05	*66.45*	3.79	22.03	13.19	5.95	***60.43***	15.81	51.57	47.29	46.36
MDTAN	TA01	*66.86*	43.28	50.45	39.86	36.81	48.61	49.33	59.60	**60.02**	48.44
	TA02	*47.61*	5.43	19.08	6.27	21.46	25.15	28.71	***39.82***	37.68	38.30
	TA03	*56.80*	20.42	47.08	20.57	19.60	35.08	41.45	***47.82***	46.74	47.36
	TA04	*64.21*	28.39	39.45	19.52	22.49	43.35	38.55	52.56	53.06	***53.28***
	TA05	*58.75*	28.83	***48.01***	28.44	32.05	26.27	42.59	45.99	45.57	46.33
MD2/2	2201	*66.26*	14.10	36.20	27.12	28.30	53.24	35.46	57.80	***59.65***	59.06
	2202	*65.44*	31.31	33.59	31.71	24.72	38.93	15.62	40.89	***47.11***	46.27
	2203	*61.72*	27.66	35.12	28.76	35.96	42.72	31.40	45.94	48.56	***49.52***
	2204	*84.85*	73.94	72.26	73.86	73.74	78.18	***83.97***	79.14	80.28	81.33
	2205	*50.28*	20.96	11.01	12.77	12.41	28.95	30.61	42.18	**42.62**	41.62
MD2/3	2301	*57.56*	23.48	24.16	24.08	26.18	37.08	25.96	43.53	45.08	***46.22***
	2302	*69.11*	51.10	50.98	50.98	48.94	50.00	42.34	62.74	62.42	***64.12***
	2303	*50.31*	22.78	27.86	22.34	20.10	27.22	26.06	**33.20**	29.50	29.80
	2304	*78.45*	36.35	29.75	24.50	23.76	61.47	39.21	70.95	72.33	**72.81**
	2305	*61.34*	21.08	31.24	24.74	39.66	43.78	***57.72***	50.94	52.70	51.68

Table 2: Estimated accuracies of the proposed algorithms in the semi-supervised scenario.

1.3.1 Uni-dimensional and multi-dimensional learning algorithms comparison

Firstly, we want to determine if there are significant differences between the uni-dimensional and the multi-dimensional learning algorithms in both learning frameworks, i.e. in the supervised and semi-supervised learning frameworks. In order to do so, for each table, we compare the 20 results obtained by each uni-dimensional algorithm with those obtained by its multi-dimensional generalisation, i.e. nB with MDnB, TAN with MDTAN and so on. This comparison is made per columns (in Tables 1 and 2) by means of the Wilcoxon signed-rank test with $\alpha = 0.05$, a non-parametric statistical hypothesis test. The null hypothesis is that ”Both classifiers have the same distribution, i.e. there is no statistical difference in the behaviour of both learning algorithms”. The use of this non-parametric test is justified: the Kolmogorov-Smirnov test ($\alpha = 0.05$) rejects the Gaussian assumption of the results.

The results of the Wilcoxon test can be found in Table 3. The first 5 rows correspond to the supervised framework (Table 1) and the last 5 to the semi-supervised (Table 2). The rows in Table 3 are as follows; the learning algorithms involved in the comparison, the means (per column) of both algorithms, the p-value of the null hypothesis and the result of the Wilcoxon test (if the null hypothesis is accepted or rejected) are shown. Moreover, the greatest mean per row is highlighted in **bold**.

Framework	Comparison	Uni-D Mean	Multi-D Mean	p-value	H_0
Supervised	nB vs MDnB	51.72	**53.65**	0.34	Accepted
	TAN vs MDTAN	**53.35**	51.69	0.15	Accepted
	TAN vs MD1/1	53.35	**54.35**	0.10	Accepted
	2DB vs MD2/2	49.86	**53.53**	> 0.01	**Rejected**
	3DB vs MD2/3	47.82	**53.07**	> 0.01	**Rejected**
Semi-supervised	nB vs MDnB	29.44	**52.02**	> 0.01	**Rejected**
	TAN vs MDTAN	38.27	**38.99**	0.37	Accepted
	TAN vs MD1/1	38.27	**56.29**	> 0.01	**Rejected**
	2DB vs MD2/2	29.53	**56.74**	> 0.01	**Rejected**
	3DB vs MD2/3	31.54	**56.48**	> 0.01	**Rejected**

Table 3: Results of the Wilcoxon signed-rank test ($\alpha = 0.05$) of comparing the results in the 20 datasets of each uni-dimensional algorithm with its generalisation to the multi-dimensional framework.

From Table 3, the following comments can be extracted: In most of the cases, the multi-dimensional approach obtains the best results in terms of mean joint accuracy. Although there are small differences between the uni-dimensional and the multi-dimensional approaches in the supervised framework (only the MD 2/3 and MD 2/3 report statistical differences), in the semi-supervised framework these differences grow larger (except for the case of the MDTAN learning algorithm, the rest of the multi-dimensional approaches report statistical differences). In this framework, the multi-dimensional approaches lead to better results, while performance degradation occurs in the uni-dimensional ones, where the mean accuracies drop dramatically. A reason to explain this could be that the uni-dimensional approaches cannot match the underlying multi-dimensional nature of the generative structures. Moreover, the good results obtained by the MD J/K highlight the flexibility of this kind of learning algorithms to capture different types of structures.

1.3.2 Learning the generative structure

We are concerned about the fact that there are situations in which the addition of unlabelled data causes degradation of the performance of the classifier [3], in contrast to the improvement of performance when adding unlabelled data, as happens with the uni-dimensional approaches in the previous section.

Many researchers, in order to prevent these situations, have proposed in the literature certain assumptions [2][14] that must be held when learning in a semi-supervised framework. One of the most important assumptions is the hypothesis presented in [3] which states that "If the correct structure of the generative model is obtained, unlabelled data improve the classifier, otherwise, unlabelled data can actually degrade performance". So therefore, we need to test if this hypothesis is verified in the proposed multi-dimensional domains. Hence, we fix, for each dataset, the structure that generates the data and learn the parameters of the model in the supervised (MLE) and semi-supervised (using the EM algorithm as defined in [6]) frameworks.

The results of this learning process in the 20 databases are shown in Table 4. The "LABELLED" column shows the accuracies obtained in all the datasets by using just supervised learning whilst the "ALL DATA" column shows the accuracies of the semi-supervised learning process. The column "$(1 - e_B)$" shows the optimal Bayes accuracy.

Based on these results, we can conclude that, as happens in the uni-dimensional framework [3], when using the real structure in the multi-dimensional framework, the unlabelled data always helps.

Family	Dataset	$(1-e_B)$	LABELLED	ALL DATA	Helps?
MDnB	NB01	*93.93*	88.48	92.16	Yes
	NB02	*93.25*	89.90	91.06	Yes
	NB03	*91.65*	88.29	89.09	Yes
	NB04	*74.36*	65.57	67.33	Yes
	NB05	*66.45*	59.44	60.43	Yes
MDTAN	TA01	*66.86*	66.69	66.79	Yes
	TA02	*47.61*	43.56	43.80	Yes
	TA03	*56.80*	50.53	50.89	Yes
	TA04	*64.21*	58.57	60.77	Yes
	TA05	*58.75*	49.99	50.71	Yes
MD2/2	2201	*66.26*	62.04	63.86	Yes
	2202	*65.44*	60.10	60.77	Yes
	2203	*61.72*	50.22	51.24	Yes
	2204	*84.85*	83.46	83.65	Yes
	2205	*50.28*	47.82	47.82	Equal
MD2/3	2301	*57.56*	53.32	54.30	Yes
	2302	*69.11*	67.26	67.62	Yes
	2303	*50.31*	43.45	45.22	No
	2304	*78.45*	75.36	75.78	Yes
	2305	*61.34*	57.34	57.40	Yes

Table 4: Accuracies obtained by supplying the EM Algorithm (as it is usually used in semi-supervised learning) with the structure that generates the data.

1.3.3 Reaching the generative structure

In almost all problems that we face in the machine learning field, there is no clue for the generative structure of the dataset. For that reason, we want to check if we can reach the generative structure and, therefore, obtain better results in terms of accuracy using the specific semi-supervised learning algorithms in each multi-dimensional Bayesian network family.

In order to do so, we check the 5 results obtained by each multi-dimensional algorithm in the family where it can lead to the generative structured, i.e. the MDnB learning algorithm in the MDnB structure family, etc. To answer the main question of this section, we compare the 5 results obtained in the supervised framework (Table 1) with the results obtained in the semi-supervised framework (Table 2). This comparison is made with a Wilcoxon signed-rank test with $\alpha = 0.05$ (the Kolmogorov-Smirnov test ($\alpha = 0.05$) reject the Gaussian assumption).

Table 5 sums up the results of the statistical test. The rows are as follows; the family of multi-dimensional Bayesian network in which the learning algorithm is applied, the learning algorithm used, the means (over just 5 results) of the learning algorithm in both supervised (Table 1) and semi-supervised frameworks (Table 2), the p-value of the null hypothesis and the result of the Wilcoxon test are shown. Moreover, the greatest mean per row is highlighted in **bold**.

Famiiy	Algorithm	Supervised Mean	Semi-supervised Mean	p-value	H_0
MDnB	MDnB	78.34	**80.07**	0.02	**Rejected**
MDTAN	MDTAN	**43.13**	40.13	0.11	Accepted
	MD 1/1	47.47	**49.16**	0.02	**Rejected**
MD 2/2	MD 2/2	54.02	**55.64**	0.11	Accepted
MD 2/3	MD 2/3	52.40	**52.93**	0.34	Accepted

Table 5: Results of the Wilcoxon signed-rank test ($\alpha = 0.05$) of comparing the results of both supervised and semi-supervised frameworks using the multi-dimensional algorithms that can lead to the generative structure in the five datasets of each family.

From these results, the following comments can be made:

- With the exception of the MDTAN learning algorithm, in the semi-supervised framework the learning algorithms lead to better results.
- Although it is difficult to obtain significant differences with only 5 results per each framework, in two cases (MDnB and MD 1/1) the differences are significant. So, we can claim that there is a tendency to better results in the semi-supervised framework when the used algorithm can lead to the generative structure.
- From the results obtained in this section and Section 4.1, it seems that the MDTAN algorithm proposed in [13] leads to very sub-optimal solutions.

1.3.4 Supervised and semi-supervised learning frameworks comparison

Once tested the algorithms that can reach the generative structure, a wider comparison has to be made. In this section, we compare the behaviour of each learning algorithm (both uni-dimensional and multi-dimensional ones) in both supervised and semi-supervised scenarios.

To achieve this task, we compare by means of a Wilcoxon signed-rank test ($\alpha = 0.05$) the results obtained in all the datasets in both supervised (Table 1) and semi-supervised learning (Table 2) frameworks, i.e. comparing the accuracies per columns, e.g. the nB column in Table 1 with the nB column in Table 2, etc.

Scenario	Algorithm	Supervised Mean	Semi-supervised Mean	p-value	H_0
Uni-dimensional	nB	**51.72**	29.43	> 0.01	**Rejected**
	TAN	**53.35**	38.27	> 0.01	**Rejected**
	2DB	**49.86**	29.54	> 0.01	**Rejected**
	3DB	**47.83**	31.54	> 0.01	**Rejected**
Multi-dimensional	MDnB	**53.65**	52.02	> 0.01	**Rejected**
	MDTAN	**51.69**	39.00	> 0.01	**Rejected**
	MD1/1	54.35	**56.29**	0.01	**Rejected**
	MD2/2	53.53	**56.74**	> 0.01	**Rejected**
	MD2/3	53.07	**56.48**	> 0.01	**Rejected**

Table 6: Results of the Wilcoxon signed-rank test ($\alpha = 0.05$) of comparing the 20 results of each learning algorithm in both supervised and semi-supervised frameworks

Table 6 shows the results of the statistical test. The rows are as follows; the learning algorithm, the means of the learning algorithm in both supervised (Table 1) and semi-supervised frameworks (Table 2), the p-value of the null hypothesis and the result of the Wilcoxon test are shown. Moreover, the greatest mean per row is highlighted in **bold**.

From the results, the following conclusions can be extracted:

- All the learning algorithms behave differently in both learning frameworks.
- In the uni-dimensional approaches, performance degradation occurs in the semi-supervised framework. This is probably because “If the correct structure of the generative model is obtained, unlabelled data improve the classifier, otherwise, unlabelled data can actually degrade performance” [3].
- With respect to the MD J/K learning algorithms, an improvement in terms of accuracy in the semi-supervised framework is observed. As stated before, this highlights the flexibility of this kind of learning algorithms to capture different types of complex structures.
- The MDnB learning algorithm reports significant performance degradation in the semi-supervised, while in the previous section, it reports significant improvement when learning MDnB structures. It denotes that the MDnB learning algorithm is very specific, it obtains very good results while dealing with problems with an underlying MDnB structure, but when the generative models are a bit complex, its rigid structure makes the algorithm lead to very sub-optimal solutions.
- In this comparison, the MDTAN algorithm also shows very poor performances in the semi-supervised framework. In addition, from the numerical results (means), it seems to be closer to the uni-dimensional approaches rather than to the multi-dimensional ones.

1.3.5 Behaviour of the learning algorithms in the semi-supervised framework

After showing the potential of semi-supervised learning, we are going to check whether statistical differences exist among the semi-supervised classifiers: not only between the multi-dimensional approaches, but also among the uni-dimensional ones. Specifically, we use Friedman test [7] with a Shaffer's static post-hoc test with $\alpha = 0.1$ as recommended in [9]. The test results are represented by means of critical difference diagrams [7], which show the mean ranks of each algorithm across all the domains in a numbered line. If there is no statistically significant difference between two algorithms, they are connected in the diagram by a straight line.

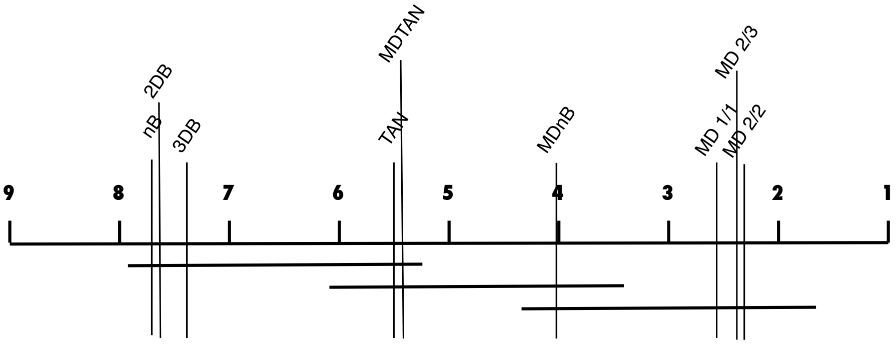


Figure 1: Accuracy ranking using both labelled and unlabelled data for the different algorithms on the 20 artificial datasets, $\alpha = 0.05$.

From the critical difference diagram (see Figure 1), we confirm the sensations extracted from the previous sections and deduce that, in the semi-supervised framework, clearly the multi-dimensional classifiers outperform the uni-dimensional techniques, with the exception of the MDTAN classifier.

1.4 General conclusions

From the experimental results shown in this report, the following major conclusions, that answer the experimental questions, can be extracted:

1. As happens in the uni-dimensional framework [3], when using the real structure to semi-supervisely learnt multi-dimensional classifiers, the unlabelled data always helps.

2. There is a tendency to achieve better classifiers in terms of joint accuracy in the semi-supervised framework when the used multi-dimensional algorithm can reach the generative structure.

3. In the uni-dimensional approaches, performance degradation occurs in the semi-supervised framework. This is probably due to the fact that the uni-dimensional approaches are not able to match the actual multi-dimensional structure of the problems.

4. Although there are small differences between the uni-dimensional and the multi-dimensional approaches in the supervised framework (only the MD 2/2 and MD 2/3 report statistical differences), in the semi-supervised framework, these differences grow larger (except for the case of the MDTAN learning algorithm, the rest of the multi-dimensional approaches report statistical differences).

5. In the semi-supervised framework, clearly the multi-dimensional classifiers outperform the uni-dimensional techniques, with the exception of the MDTAN classifier.

6. The MDnB learning algorithm [13] is very specific, it obtains very good results when dealing with problems with an underlying MDnB structure, but when the generative models are more complex, its rigid structure makes the algorithm lead to very sub-optimal solutions.

7. The MDTAN algorithm [13] also shows very poor performances in the semi-supervised framework.

8. The MD J/K learning algorithms have great flexibility to capture different types of complex structures that results in an improvement in terms of joint accuracy in the semi-supervised framework.

References

[1] B. Calvo and J. L. Flores. Iclab intelligent computing laboratory: A library of data mining and optimization algorithms, 2009. `http://iclab.sourceforge.net/`.

[2] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning.* The MIT Press, 2006.

[3] I. Cohen. *Semisupervised Learning of Classifiers with Application to Human-Computer Interaction.* PhD thesis, University of Illinois at Urbana-Champaign, 2003.

[4] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[5] F. Cozman and I. Cohen. Risk of semi-supervised learning. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-supervised learning.* The MIT Press, 2006.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[7] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[8] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[9] S. García and F. Herrera. An extension on "statistical comparisons of classifiers over multiple data sets". *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[10] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley, 1992.

[11] J. D. Rodriguez, A. Perez, and Lozano J. A. Sensitivity analysis of k-fold cross-validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):569–574, 2010.

[12] M. Sahami. Learning limited dependence bayesian classifiers. In AAAI Press, editor, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 335–338, 1996.

[13] L. C. van der Gaag and P. R. de Waal. Multi-dimensional Bayesian network classifiers. In *Proceedings of the Third european workshop in probabilistic graphical models*, pages 107–114, 2006.

[14] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

5

Semi-supervised Multi-class Classification Problems with Scarcity of Labelled Data: A Theoretical Study

How is an error possible in mathematics?

- Henri Poincare, *The Foundations of Science: Science and Hypothesis, the Value of Science, Science and Method*

Semi-supervised multi-class classification problems with scarcity of labelled data: A theoretical study

Jonathan Ortigosa-Hernández, Iñaki Inza, and Jose A. Lozano, *Member, IEEE.*

***Abstract*—In recent years, the performance of semi-supervised learning has been theoretically investigated. However, most of this theoretical development has focussed on binary classification problems. In this paper, we take it a step further by extending the work of Castelli and Cover [1] [2] to the multi-class paradigm. Particularly, we consider the key problem in semi-supervised learning of classifying an unseen instance $\mathbf{x}$ into one of K different classes, using a training dataset sampled from a mixture density distribution and composed of l labelled records and u unlabelled examples. Even under the assumption of identifiability of the mixture and having infinite unlabelled examples, labelled records are needed to determine the K decision regions. Therefore, in this paper, we first investigate the minimum number of labelled examples needed to accomplish that task. Then, we propose an optimal multi-class learning algorithm which is a generalisation of the optimal procedure proposed in the literature for binary problems. Finally, we make use of this generalisation to study the probability of error when the binary class constraint is relaxed.**

***Index Terms*—Semi-supervised learning, probability of error, labelled and unlabelled samples, multi-class classification.**

I. Introduction

THROUGHOUT recent years, the problem of learning from both labelled and unlabelled observations has been of practical relevance. In many applications, an enormous amount of unlabelled examples is available with little cost, whilst obtaining enough labelled examples to learn a classifier may be costly and time consuming. In such cases, semi-supervised learning (SSL) [3] appears to be a tool that is able to obtain accurate classifiers in such circumstances.

Within the state-of-the-art literature, SSL has been empirically and theoretically studied. Regarding the practical applications, it has been used to tackle (i) binary problems [4], (ii) problems with multiple class values [5], or even (iii) multi-dimensional problems [6], where several multi-class variables have to be predicted simultaneously. The probability of error of SSL has also been theoretically investigated. However, the scope of the studied problems does not cover the entire range of the practical applications. The majority of the theoretical works proposed in this area have mainly focussed on standard binary problems [1] [2] [7] [8]. To the best of our knowledge, only in [9], is multi-class framework explicitly tackled; yet, it has been studied with a slightly different perspective as how it is in this paper. Moreover, most of the works assume the datasets have a large enough number of labelled observations [8] [9], which is an unnatural situation in this scenario.

For those reasons, we think that it is interesting and demanding to generalise several of the theoretical findings of the state-of-the-art literature in SSL binary problems to the scenarios where there are more than two classes, concentrating on the cases where the number of labelled observations is minimal. However, as we show throughout the whole paper, the previous state-of-the-art studies do not straightforwardly work for multi-class problems, so there are several theoretical gaps that must be covered.

Therefore, in order to allow a potential enlargement of the scope of the theoretically studied SSL problems, in this paper, we first perform an exhaustive review of the previous theoretical findings. It is focussed on the frameworks utilised in each study, the feasibility of their conclusions to the multi-class frameworks and the remaining open questions found in them. So, guided by this, we contribute with a natural extension to the multi-class paradigm of the SSL binary framework already proposed by Castelli and Cover [1] [2]. This extension is performed by addressing the following issues:

- First, the proposal of an optimal theoretical SSL algorithm able to work in the multi-class framework: PC_{SSL} (**P**ermutation of **C**omponents in **S**emi-**S**upervised **L**earning). It is a natural extension of the optimal procedure proposed in [1] and [2] for binary problems.
- Even under the assumption of having ∞ unlabelled records and identifiability of the decision regions, labelled samples are still needed to determine the labels of the K decision regions. However, what is the minimum number of labelled records needed to uniquely determine the decision regions? In the case of binary problems, just one labelled datum is needed [1]. In the multi-class scenario, however, the calculation of this value becomes more complex. For that reason, in this paper, we define and calculate l_K as the expected minimum number of labelled records to uniquely determine those K decision regions.
- A formula to calculate the probability of error, $P_e(l,\infty)$ (given l labelled instances and an infinite number of unlabelled records), for SSL problems where the binary constraint is relaxed and the pairwise intersections among the decision regions are empty. When the regions are non-mutually disjoint, upper and lower bounds are given for $P_e(l,\infty)$, generalising the statements of [1] [2] to the multi-class scenario. In both scenarios, $P_e(l,\infty)$ decreases to the Bayes error exponentially fast in l.

The rest of the paper is structured as follows: In Section II, the notation, the properties and the proposed multi-class framework are introduced. Then, the state-of-the-art literature

The authors are with the Intelligent Systems Group in the Department of Computer Science and Artificial Intelligence, Computer Science Faculty, The University of the Basque Country, Donostia-San Sebastián, 20018, Spain
E-mail: jonathan.ortigosa@ehu.es

is reviewed in Section III. Section IV reviews the framework proposed in [1] and [2] for binary problems, highlighting the issues that must be solved before extending it. Our algorithm PC$_{\text{SSL}}$ is proposed in Section V. In that section, the Voting learning procedure [9], the recently proposed multi-class approximated method, is also introduced. In Section VI, the problem of determining the minimum number of labelled records needed to determine the decision regions in the multi-class framework is tackled. Whilst Section VII is devoted to the calculation of the probability of error in the SSL multi-class scenario, in Section VIII, we carry out an empirical experimentation on the contributions of this paper. Then, the issue of extending the contributions of this paper to practical SSL is approached in Section IX. Finally, Section X provides a summary of the paper.

Furthermore, due to the limited space, some appendices are placed in a separated document available to download from our website[1]. There, Appendix A contains mathematical proofs for the less crucial theorems. Appendix B presents supplementary experimental results. Finally, Appendix C shows the source code used to ensure the replicability of the exposed studies.

II. General notation and Framework

Firstly, we introduce the multi-class framework which will be used throughout the rest of the paper and which has been borrowed and extended from that proposed by Castelli and Cover in the key works [1] and [2] for binary problems.

A. Framework

As we want to study the optimal probability of error $P_e(l,u)$ of classifying the instance $(\mathbf{x}^{(0)}, c^{(0)})$ in the SSL multi-class scenario having l labelled instances and u unlabelled records, the following framework is proposed: Let $D = \mathcal{L} \cup \mathcal{U}$ be a training dataset of a common SSL problem with K classes which can be divided into two different subsets: $\mathcal{L}$, the set of l labelled examples $\{(\mathbf{x}^{(1)}, c^{(1)}), \ldots, (\mathbf{x}^{(l)}, c^{(l)})\} = \{(\mathbf{x}^{(n)}, c^{(n)})\}_{n=1}^{l}$, and $\mathcal{U}$, the set of u unlabelled examples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(u)}\} = \{\mathbf{x}^{(m)}\}_{m=1}^{u}$. Due to the fact that the applications of SSL deal with very few labelled examples and a huge amount of unlabelled data ($l/u \sim 0$) [3], the theoretical studies usually make the reasonable assumption of having $l > 0$ labelled and ∞ unlabelled records. Moreover, with this assumption and by proposing an optimal learning algorithm, we can establish a fundamental limit in the performance of any existing SSL multi-class algorithm. Therefore, unless otherwise specified, we assume that $u = \infty$. Then, let the class labels $\{c^{(n)}\}_{n=1}^{l}$ be l i.i.d. random values where the prior probability of observing a sample of class c_i is $\eta_i = P(C = c_i) > 0, i = 1, \ldots, K$ and $\sum_i \eta_i = 1$. We also assume that each observation $\mathbf{x} \in \mathcal{L}$ is i.i.d. according to a mixture component $f(\mathbf{x}|C = c_i; \boldsymbol{\theta}_i) \in \mathcal{F}$, where $\mathcal{F}$ is a function set containing the mixture components of a mixture density. There, $\boldsymbol{\theta}_i$ stands for the set of the parameters of the mixture component i, being $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ the vector of the parameters of the whole mixture. For simplicity of notation, henceforth, we denote $f(\mathbf{x}|C = c_i; \boldsymbol{\theta}_i)$ by $f_i(\mathbf{x})$. Then, we define the mixture joint density, which generates the labelled samples, as

$$f(\mathbf{x}, c) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x}) \mathbb{1}(c = c_i), \tag{1}$$

where the function $\mathbb{1}(c = c_i)$ is 1 if $c = c_i$, and 0 otherwise, i.e. each mixture component models just one class value.

The infinite unlabelled samples appear to be i.i.d. random variables distributed according to the mixture density given by

$$f(\mathbf{x}) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x}) \tag{2}$$

and which corresponds to the marginal of $f(\mathbf{x}, c)$ on $\mathbf{x}$.

Let $(\mathbf{x}^{(0)}, c^{(0)})$ be a instance to be classified and distributed according to the joint density (1). As we want to infer $c^{(0)}$ from the observation $\mathbf{x}^{(0)}$, when $\forall i, f_i(\mathbf{x})$ and η_i are known, the optimal classifier is given by the Bayes decision rule (BDR)

$$\hat{c}^{(0)} = \arg\max_i \eta_i f_i(\mathbf{x}^{(0)}) \tag{3}$$

with a corresponding probability of error

$$e_B = 1 - \sum_{i=1}^{K} \eta_i \int_{R_i} f_i(\mathbf{x}) d\mathbf{x}, \tag{4}$$

which is called the Bayes error and is the highest lower bound of the probability of error of any classification rule. There,

$$R_i = \{\mathbf{x} : \eta_i f_i(\mathbf{x}) - \max_{i' \neq i} \eta_{i'} f_{i'}(\mathbf{x}) > 0\} \tag{5}$$

is the region where $\eta_i f_i(\mathbf{x})$ is maximum and so the instances are assigned to the class c_i. However, this classifier cannot be used in practise as, in general, $f_i(\mathbf{x})$ and η_i are unknown.

B. Identifiability

Having an infinite amount of unlabelled examples ($u = \infty$) available is equivalent to knowing $f(\mathbf{x})$, i.e. the mixture density can almost surely be recovered from the unlabelled data [1]. So, in order to be able to take advantage of the information provided by the unlabelled data, in this framework, we assume that the mixture $f(\mathbf{x})$ is identifiable, i.e. the components of the mixture $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}) \in \mathcal{F}$ and the class priors $\eta_1, \ldots, \eta_K$ can be uniquely decomposed from the density function. This assumption is well-grounded since it holds for most of the well-known distributions. In the continuous case and having a finite K, $f(\mathbf{x})$ is identifiable iff $\mathcal{F}$ is said to be linearly independent [10]. i.e. for real constants $\alpha_i, i = 1, 2, \ldots, K$,

$$\sum_{i=1}^{K} \alpha_i f_i(\mathbf{x}) = 0 \Longrightarrow \forall i, \alpha_i = 0.$$

Particularly, it has also been shown that the mixtures of univariate Gaussian, Gamma, exponential, Cauchy and Poisson functions are identifiable iff there are no empty components ($\forall i, \exists \mathbf{x}$ s. t. $f_i(\mathbf{x}) \neq 0$), and there are not two components with the same parameters ($\forall i \neq j, \exists \mathbf{x}$ s. t. $f_i(\mathbf{x}) \neq f_j(\mathbf{x})$). In general, discrete distributions are not identifiable, except for

[1] www.sc.ehu.es/ccwbayes/members/jonathan/home/error.html

the case of binomial and multinomial distributions. They are identifiable if $K < \infty$ [11] [12] [13].

C. Probability of error in the absence of labelled examples

Next, if the generative model is identifiable, we can recover all the single-component distributions and all the class priors of the generative model from just the unlabelled data. However, it is only identifiable up to a permutation π of its single-components. That is, in a scenario of absence of labelled data, each recovered component can be labelled with a conventional name j which does not necessarily coincide with the real label i of the component. This permutation can be defined as $\pi = (\pi(1), \ldots, \pi(K)) \in S_K$, where each element $\pi(j) = i$ denotes that the j-th decomposed component distribution, i.e. $f_{\pi(j)}(\mathbf{x})$, is associated to the i-th class value, and S_K represents the set of all possible component-label correspondences. Then, the mixture distribution (eq. (2)) can be expressed as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x}) = \sum_{j=1}^{K} \eta_{\pi_c(j)} f_{\pi_c(j)}(\mathbf{x}),$$

where π_c is the unknown correct correspondence between real labels i and decomposed components j, which cannot be determined without labelled records. Generalising the conclusions of [3] for binary problems, it can be seen that, by means of the infinite unlabelled records, the set containing all the possible models which can generate the data is reduced to just only a set containing $K!$ possibilities (where the real generative model is included). Unfortunately, without labelled data, this reduction is pointless, as shown in the following theorem:

Theorem 1. ***(Probability of error with no labelled data)***[2] *The probability of error of classifying a new sample $(\mathbf{x}^{(0)}, c^{(0)})$ of a K-class problem with any classifier learnt with no labelled examples and any $u \geq 0$ number of unlabelled samples coincides with the probability of error of the random classifier, which it is equal to*

$$P_e(0, u) = e_0 = \frac{(K-1)}{K}, \ \forall u \geq 0. \tag{6}$$

Then, in SSL, the use of several labelled examples is crucial. Only for $l > 0$, the unlabelled records influence the reduction of the probability of error.

III. Literature review

The probability of error of SSL has been investigated in the literature. Throughout recent years, several key results have been presented on this topic. Although these papers theoretically approach SSL by means of different frameworks and under different assumptions, their findings are equivalent in most of the cases. In the following paragraphs, we taxonomise these theoretical proposals into three different subsets assumed in the SLL community (a summary can be found in Table I):

1) Papers which deal with correct models[3], i.e. the semi-supervisely learned models match the generative models,
2) works in which incorrect models are assumed, i.e. the models do not match the generative distribution, and
3) papers dealing with imperfect models, i.e. those models which, despite not matching the generative models perfectly, have a presumedly small error.

Although incorrect models and imperfect models have been clearly defined in the literature, they are almost equivalent. Neither of them match the generative distribution of the data which causes performance degradation of the learned classifiers. The subtle difference relies on the perspective of the authors towards them. While the authors who deal with incorrect models only perceive the degradation of the performance, the authors dealing with imperfect models study the impact of the difference between the generative and learnt models on the resulting error, or they even try to improve the safeness of SSL techniques. In the following paragraphs, we review several contributions to these three different approaches.

A. Correct models

1) Ratsaby and Venkatesh [7]: The authors try to shed some light on the question "How many unlabelled examples is each labelled example worth?" under the Probably Approximately Correct (PAC) learning framework. Their goal is to determine how the error rate depends on the sample sizes l and u, and on the dimensionality n. In order to achieve this goal, several assumptions are made: (1) Learning the correct model. (2) Two-class multivariate Gaussian mixture problem with equal unit variance matrices ($\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{I}\}$). (3) Equiprobable class priors, i.e. $\eta_i = 1/2$. (4) $\mathbf{x}$ is n-dimensional.

With the aim of reaching a rough measure on the value of one labelled example in terms of unlabelled samples, the authors calculate how many unlabelled records must be added to a supervised algorithm to remove just one labelled record while keeping a similar error. The result is the following:

$$\frac{u_{SSL}}{(l_{SUP} - l_{SSL})} = \frac{zn^n}{\epsilon^3 \delta \log n},$$

where u_{SSL} is the number of unlabelled examples in a semi-supervised problem, l_{SUP} is the number of labelled instances in a purely labelled problem, l_{SSL} is the number of labelled examples in the semi-supervised problem, z is a constant, ϵ is the upper bound of the error with at least $(1-\delta)$ confidence, and n is the dimensionality. They conclude that unlabelled data are extremely helpful due to the fact that they reduce the demands on the number of labelled examples. Then, each labelled datum is more valuable and the probability of error decreases exponentially fast in l_{SSL}, not polynomially fast in l_{SUP}, as happens in supervised learning.

2) Castelli and Cover [1] [2]: The same conclusion as in the previous work is reached but from the perspective of the decision theory framework and by weakening several assumptions. The detailed explanation of their findings can be found in Section IV.

[2] This holds true independently of the value of the class priors $\eta_1, \ldots, \eta_K$.

[3] Note that, by the assumptions of an infinite number of unlabelled examples and identifiability, our paper relies on this category.

Model	Ref.	Framework	Addressed Question	Assumed Model	MC	Conclusion
CORRECT	[7]	PAC learning	How many unlabelled examples is each labelled example worth?	Multi-variate Gaussian mixture model with unit covariance matrices and equiprobable classes.	No	If the parametric model assumptions are satisfied, labelled examples are exponentially more valuable than unlabelled examples in reducing the error.
	[1], [2]	Decision theory	How the optimal probability of error varies in l and u?	Identifiable mixture densities.	No	
	[14]	Parameter estimation	What is the contribution of unlabelled data in the parameter estimation?	Generic parametric models $p(x, c\|\theta) = p(x\|\theta)p(c\|x, \theta)$.	Yes	Unlabelled data always helps due to the fact that the Fisher information is increased.
INCORRECT	[4]	Bayesian networks	Does unlabelled data always help?	Naive Bayes, and Tree-augmented naive Bayes.	No	Unlabelled data only helps when the learned matches the generative model.
	[15]	Parameter estimation	What is the relationship between the model misspecification and performance degradation?	Identifiable mixture densities (Gaussian mixtures for the examples).	No	As the number of unlabelled record increases, the probability of degradation is positive with incorrect model.
IMPERFECT	[8]	Density estimation	What is the value of labelled and unlabelled data when the assumed densities do not follow the parametric model?	Two equiprobable n-dimensional spherical Gaussian mixtures. Let e be the difference between them.	No	The error is reduced exponentially with the number of labelled records until e. After that, the error is only reduced polynomially fast.
	[9]	Density estimation	What is the convergence rate of the error?	Identifiable mixture densities.	Yes	Similar results to [1] and [2] for the multi-class scenario.

TABLE I: Summary of the theoretical SSL state-of-the-art literature (MC = multi-class).

3) Zhang and Oles [14]: The authors address the problem of the value of unlabelled data, i.e. how unlabelled records help in reducing the probability of error, by analysing their efficacy in the estimation of the parameters of the model. They argue that unlabelled examples have a positive impact on the efficacy of the estimations of the real model parameters $\boldsymbol{\theta}$, in the cases of combining both (1) parametric generative models defined as $p(x, c|\boldsymbol{\theta}) = p(x|\boldsymbol{\theta})p(c|x, \boldsymbol{\theta})$, and (2) SSL techniques where a classifier is trained with labelled and unlabelled data in an iterative manner. Then, the authors claim that under the correct model assumption, adding unlabelled data always helps because Fisher information is increased:

$$I_{(l+u)}(\boldsymbol{\theta}) = I_l(\boldsymbol{\theta}) + I_u(\boldsymbol{\theta}),$$

where $I_{(l+u)}(\boldsymbol{\theta})$ is the Fisher information of $\boldsymbol{\theta}$ using both labelled and unlabelled subsets, $I_l(\boldsymbol{\theta})$ using the labelled subset, and $I_u(\boldsymbol{\theta})$ using the unlabelled data.

4) Cohen et al. [4]: The authors address the question of whether unlabelled data always helps. By means of Bayesian network classifiers (naive Bayes and tree augmented naive Bayes models), they focus on the convergence of the semi-supervised maximum likelihood estimator of the model, $\boldsymbol{\theta}^*$,. They argue that the limiting value of the MLE, as the number of labelled and unlabelled records increases, is a linear combination of the supervised and unsupervised expected log-likelihood functions:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \Big[\beta E\big[\log p(c, \mathbf{x}|\boldsymbol{\theta})\big] + (1-\beta)E\big[\log p(\mathbf{x}|\boldsymbol{\theta})\big]\Big],$$

where β is the probability of sampling labelled data, i.e. the ratio of the amount of labelled and unlabelled observations. They conclude that unlabelled examples always improve the performance when the correct model assumption is met, and may degrade it when the opposite happens.

B. Incorrect models

1) Yang and Priebe [15]: Under the assumption of learning the correct model, SSL techniques seem to work appropriately. However, when this requirement is not met, performance degradation may occur in the classifiers as unlabelled examples are introduced. Therefore, in order to study the degradation, the authors define $\boldsymbol{\theta}_l^*$ as the limiting value of the supervised MLE (as the number of labelled data increases) of the real model parameters $\boldsymbol{\theta}$, and $\boldsymbol{\theta}_u^*$ as the limit of the unsupervised MLE (as the number of unlabelled records increases) of $\boldsymbol{\theta}$, assuming that the generative model is a finite Gaussian mixture model ($\boldsymbol{\theta}_i = \{\mu_i, \sigma\}$) and the estimators exist under mild regularity conditions. First, the authors corroborate the achievements of [1] when the correct model assumption is met by proving that both limits tend to the same parameter value. However, when the learnt model is misspecified, the supervised and the semi-supervised MLE parameters may converge to different values, i.e. $\boldsymbol{\theta}_l^* \neq \boldsymbol{\theta}_u^*$. They also state that for any fixed finite l or $l \to \infty$, as $l/u \to 0$, the limit of the maxima of the semi-supervised likelihood parameters is the unsupervised MLE limit $\boldsymbol{\theta}_u^*$, and degradation may appear: If $P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_l^*)) < P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_u^*))$, then for a given misspecified model, $\exists l$, s.t.

$$\lim_{u\to\infty} P_e(l, u) = P\{P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_l^*)) < P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_u^*))\} > 0.$$

That is, for incorrect models, SSL yields degradation with positive probability as $u \to \infty$.

C. Imperfect models

1) Sinha and Belkin [8]: The authors focus on the situation when the correct model assumption is only satisfied to a certain degree of precision, either because the assumed model is correct but the dataset is imperfect or because the assumed model does not follow the generative model. For the purpose of this paper, we aim for the latter case. There, ϵ is defined as a perturbation size, i.e. a rough measure that indicates to what extent the true model differs from the assumed model.

It is proved that, under the assumption of having two equiprobable spherical Gaussian mixture components as generative models, as labelled examples are added to a training set with infinite unlabelled records, the probability of error is reduced exponentially in the number of labelled examples (as argued in previous works) but only until $e_B + \epsilon$. After that, the perturbation ϵ is only reduced polynomially fast in l. Moreover, they also state that, for a positive perturbation size, there is a number of unlabelled examples that beyond which any extra additions do not decrease the probability of error.

2) Chen and Li [9]: Although the authors extend the findings of [8] by assuming an imperfect model, it is better to remark their efforts to theoretically deal with the multi-class framework. Under the correct model assumption, they show that labelled examples reduce the probability of error exponentially fast, as happens in binary problems. They also proposed an approximated algorithm (called Voting) that can utilise the unlabelled data efficiently, i.e. achieving a fast convergence rate. Although, to the best of our knowledge, it is the only theoretical algorithm proposed for the multi-class framework, it does not achieve the optimal probability of error as we demonstrate throughout the paper.

IV. Semi-supervised learning in binary problems

In this section, we review the key works [1] and [2] highlighting why new strategies must be adopted in the multi-class scenario since they cannot be straightforwardly generalised.

A. Obtaining the binary classifier

Under the assumptions of (i) learning the correct model, (ii) having the unlabelled samples distributed according to the identifiable mixture density $f(\mathbf{x}) = \eta_1 f_1(\mathbf{x}) + \eta_2 f_2(\mathbf{x})$, and (iii) having $f_1(\cdot)$, $f_2(\cdot)$, η_1, and η_2 $(= 1 - \eta_1)$ unknown, the authors define a procedure (see Algorithm 1) to obtain the optimal binary classifier in SSL.

Algorithm 1 Optimal theoretical procedure for SSL binary problems [1] [2]

```
1: LEARNING TASK:
     • Stage 1 Use unlabelled set $\mathcal{U}$ to obtain $f(\mathbf{x})$ and,
       by identifiability, a permutation of its components
       ($f_{\pi(1)}(\cdot)$, $f_{\pi(2)}(\cdot)$, $\eta_{\pi(1)}$, and $\eta_{\pi(2)}$).
     • Stage 2 By means of the likelihood ratio test and the
       labelled set $\mathcal{L}$, determine the correspondence between
       the real classes and the current mixture components:
       $\hat{\pi}(1) = 1$ and $\hat{\pi}(2) = 2$, or $\hat{\pi}(2) = 1$ and $\hat{\pi}(1) = 2$.
2: CLASSIFICATION TASK:
     • Stage 3 Assign the sample $\mathbf{x}^{(0)}$ to the class induced
       by the BDR using the learned model.
```

The procedure can be divided into two major parts: (a) the *learning task*, where a model is learnt using the training dataset D, and (b) the *classification task*, where the unseen instances are classified according to the previously learnt model. The learning task is split into two stages. First, the components of the mixture are identified by means of the unlabelled subset $\mathcal{U}$ (Stage 1), and then, the labelled subset $\mathcal{L}$ is used in the likelihood ratio test to assign a class to each component (Stage 2). Finally, the classification task is composed of just one stage, namely Stage 3, in which the BDR is used to determine the class of the unseen instance given the assignment of the two previously made mixture components.

According to [1], this procedure is optimal, i.e. it achieves the highest lower bound of the probability of error of any semi-supervised classifier, since all the three stages are optimal. By means of identifiability and the correct model assumption, the recovered components are a permutation of the components of the unknown real model. The likelihood ratio test is optimal for two simple hypotheses and the BDR (eq. (3)) is the optimal classification rule. Unfortunately, this optimal procedure cannot be directly transferred to the multi-class scenario. Although, both Stage 1 and Stage 3 can be straightforwardly used to deal with $K \geq 2$ classes, the optimality of Stage 2 can only be guaranteed for $K = 2$. In the multi-class framework, new procedures must be proposed for Stage 2 as we need to deal with more than two simple hypotheses. In Section V, we tackle this problem.

B. Minimum number of labelled examples

Labelled records are needed to correctly determine the correspondence between the classes and the decomposed mixture components (Stage 2 of Algorithm 1). But, how many labelled examples are needed to carry out such a task?

It is shown in [1] that, for the case of $K = 2$, just one labelled example is enough. Once the two components have been identified (as $f_{\pi(1)}(\cdot)$ and $f_{\pi(2)}(\cdot)$) in Stage 1, with just one labelled datum $(\mathbf{x}, c_i)$, the correspondence π can be, correctly or incorrectly, uniquely determined. By means of the likelihood ratio test (Stage 2), the component that is maximum ($f_{\pi(j)}(\cdot)$) in the region R_j where the instance lies is labelled with the label of the instance ($\pi(j) = i$). Then, the other component is labelled by a process of elimination. But is just one labelled example enough to assign a label to each component in the multi-class paradigm? The answer is no. With more than two classes and a labelled datum, we can only identify just one component, the one where the datum seems to belong to. For the rest of the components, there is not enough information in the subset $\mathcal{L}$. So, how much labelled data is needed to uniquely determine the correspondence π of a K class problem? We deal with this issue in Section VI.

C. Probability of error

Under the proposed binary framework, the probability of error, $P_e(l, u)$, is calculated in [1]. First of all, the authors prove for the case of binary problems that ($P_e(0, u) = 1/2, \forall u \geq 0$ (Theorem 1)). Then, they stated that with infinite labelled examples the Bayes error is reached ($P_e(\infty, u) = e_B, \forall u \geq 0$), and that the probability of error of having just one labelled example and no unlabelled data is $P_e(1, 0) \leq 2\eta_1\eta_2 \leq 1/2$.

After all these specific scenarios, the authors make use of Algorithm 1 to study the value of $P_e(l, \infty)$. First, they analyse the case of $P_e(1, \infty)$, where only one labelled datum plus infinite unlabelled records are available. Under the correct model assumption and by identifiability, Stage 1 cannot lead to a classification error. Therefore, in this case, a classification error only occurs when either Stage 2 or Stage 3 yields an incorrect answer, i.e. either (i) the classes of the mixture components are reversed or either (ii) the BDR misclassifies the instance. When both Stage 2 and Stage 3 result in wrong answers (the classes of the mixtures are reversed and the BDR misclassifies the instance), both mistakes cancel each other out in the 2-class scenario. Let the event $A \triangleq \{\text{error in Stage } 2\}$, then $P(A) = e_B$ and the probability of error is calculated as:

$$\begin{aligned} P_e(1, \infty) = P(\hat{c}^{(0)} \neq c^{(0)}) = \\ = P(\hat{c}^{(0)} \neq c^{(0)}|A)P(A) + P(\hat{c}^{(0)} \neq c^{(0)}|\bar{A})P(\bar{A}) = \\ = (1 - e_B)e_B + e_B(1 - e_B) = 2e_B(1 - e_B). \end{aligned}$$

In general, for the case of l labelled examples, the authors follow a similar reasoning to the calculation of $P_e(1, \infty)$, i.e. determining when an error is committed in just one of the two previously exposed stages (Stage 2 and Stage 3). However, in this general case, $P(A)$ does not coincide with e_B, and therefore it must be calculated. Under these premises, they reach the conclusion that the probability of error is:

$$P_e(l, \infty) - e_B = exp\Big\{ - lZ + o(l)\Big\},$$

where $Z = -\log \Big\{ 2\sqrt{\eta_1 \eta_2} \int \sqrt{f_1(\mathbf{x}) f_2(\mathbf{x}) d\mathbf{x}} \Big\}$ is the Bhattacharyya distance between the densities $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ multiplied by a term equal to $\log{(2\sqrt{\eta_1 \eta_2})}$. In [2], they extend their work to the case of $u < \infty$, reaching to

$$P_e(l, u) - e_B = O\Big(\frac{1}{u}\Big) + exp\Big\{ - lZ + o(l)\Big\}.$$

The authors conclude that, in the case of binary problems, it turns out that unlabelled samples are only polynomially valuable, whilst labelled samples are exponentially valuable in reducing the error. So then, what is the probability of error, $P_e(l, \infty)$, when the binary class constraint is relaxed? Does this conclusion still hold in those cases? In Section VII, our main objective is to address these.

V. Semi-supervised multi-class learning strategies

Guided by the aforementioned concerns, we now tackle the first of them; proposing a strategy for Stage 2 which is able to determine a correspondence π by using a labelled set $\mathcal{L}$ with $K \geq 2$ classes, i.e. *to assign each mixture component to a specific class*. In the following subsections, we first introduce Voting [9] as the only method for Stage 2 where the binary constraint is relaxed which has already been proposed in the literature. However, since it does not make optimal usage of the data, we propose PC$_{\text{SSL}}$, an optimal multi-class learning strategy for Stage 2, which is a natural extension and generalisation of the one proposed by [1]. So, in the studied scenario, the whole multi-class procedure remains as can be seen in Algorithm 2.

Algorithm 2 Theoretical procedure for SSL multi-class problems

```
1: LEARNING TASK:
     • Stage 1 Use unlabelled set $\mathcal{U}$ to obtain $f(\mathbf{x})$ and,
       by identifiability, a permutation of its components
       ($f_{\pi(j)}(\cdot)$, and $\eta_{\pi(j)}, j = 1, ..., K$).
     • Stage 2 Use the labelled set $\mathcal{L}$ to determine the
       correspondence between the classes and the mixture
       components, i.e. the permutation of the components $\hat{\pi}$,
       by means of the semi-supervised multi-class learning
       procedure: (i) Voting [9], or (ii) our proposal PC$_{\text{SSL}}$
2: CLASSIFICATION TASK:
     • Stage 3 Assign the sample $\mathbf{x}^{(0)}$ to the class induced
       by the BDR using the learned model.
```

A. Voting

In [9], the authors propose Voting as a simple method to determine the permutation π by extending the majority vote method for binary problems [16] to the multi-class framework.

There, it is assumed that the regions R_j (see equation 5) are known by identifiability, and that the observations of $\mathcal{L}$, i.e. $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(l)}\}$, can be split into K different subsets named $\mathcal{L}_i$, for $i = \{1, ..., K\}$, such that, each $\mathcal{L}_i$ stands for the set containing all the observations in $\mathcal{L}$ of the class value c_i. Under these premises, the learning method is as follows: First, it counts the labels in each region and then, assigns the permutation which maximises the total number of counts. Formally, it can be defined as follows:

$$\hat{\pi}_v = \arg\max_{\pi} V(\pi; \mathcal{L}) = \arg\max_{\pi} \sum_{i=1}^{K} |\mathcal{L}_i \cap R_{\pi^{-1}(i)}|. \quad (7)$$

where $|\mathcal{L}_i \cap R_{\pi^{-1}(i)}|$ is the number of examples of class c_i found in the region $R_{\pi^{-1}(i)}$. Although Voting is asymptotically optimal (as $l \to \infty$), it does not make optimal usage of the dataset when l is relatively small, the natural domain for SSL.

B. PC$_{SSL}$

Due to the aforementioned drawbacks of the Voting procedure, we propose a new theoretical SSL strategy which makes optimal usage of the labelled data. It is named PC$_{\text{SSL}}$ (**P**ermutation of **C**omponents in **S**emi-**S**upervised **L**earning), and it uses the principle of maximum likelihood to determine the label permutation π of the previously decomposed components. It not only coincides with the method for Stage 2 of Cover and Castelli for $K = 2$ ([1] and [2]), but also it is a natural extension of that method to the multi-class framework. Formally, the learning strategy is as follows:

$$\hat{\pi}_p = \arg\max_{\pi} L(\pi; \mathcal{L}) = \arg\max_{\pi} \prod_{i=1}^{K} \prod_{\mathbf{x} \in \mathcal{L}_i} \eta_i f_{\pi^{-1}(i)}(\mathbf{x}). \quad (8)$$

Briefly, PC$_{\text{SSL}}$ works as follows: it returns the correspondence π between the classes and the identified components with the highest likelihood function $L(\pi; \mathcal{L})$. The following theorem proves the optimality of our proposal:

Theorem 2. ***(Optimality of PC$_{SSL}$)*** *PC$_{SSL}$ is an optimum learning procedure for Stage 2 of Algorithm 2.*

Proof: Let $\pi^* = \arg\max_\pi P(\pi|\mathcal{L})$ be the BDR for classifying a labelled subset into one of the $K!$ different possible permutations. Since it is the optimal classifier, PC$_{\text{SSL}}$ can be proved to be optimum if both classifiers are equivalent, i.e. $\pi^* = \hat{\pi}_p, \forall \mathcal{L}$. To prove this statement, we reduce the BDR to PC$_{\text{SSL}}$ by rewriting the optimal rule as

$$\pi^* = \arg\max_\pi \text{Pr}\{\pi, \mathcal{L}\} = \arg\max_\pi f(\mathcal{L}|\pi)P(\pi),$$

where the notation $\text{Pr}\{\cdot\}$ is used to omit measure-theoretical details for the sake of clarity. Regarding $f(\mathcal{L}|\pi)$, as *max* measures of disjoint events are independent and fixing π, we reach the conclusion that it is equal to the likelihood:

$$\begin{aligned} f(\mathcal{L}|\pi) &= \prod_{i=1}^{K} \eta_i f(\mathcal{L}_i|\pi) = \prod_{i=1}^{K} \prod_{\mathbf{x}\in\mathcal{L}_i} \eta_i f(\mathbf{x}|\pi^{-1}(i)) \\ &= \prod_{i=1}^{K} \prod_{\mathbf{x}\in\mathcal{L}_i} \eta_i f_{\pi^{-1}(i)}(\mathbf{x}) = L(\pi;\mathcal{L}) \end{aligned}$$

Concerning the model priors $P(\pi)$, it is reasonable to assume them to be uniformly distributed (as in the key work of Cover and Castelli [1]), which holds when the components are indexed in a random uniform manner. Therefore,

$$\pi^* = \arg\max_\pi f(\mathcal{L}|\pi)P(\pi) = \arg\max_\pi L(\pi;\mathcal{L}) = \hat{\pi}_p$$

■

C. Computational complexities of both procedures

While the solution of eq. (7) (Voting) and eq. (8) (PC$_{\text{SSL}}$) for a given $\mathcal{L}$ can be straightforwardly obtained by an exhaustive search over the $K!$ factorial permutations, this process can be simplified, in terms of computational complexity, by rewriting both equations as linear assignment problems and, then, using the Hungarian [22] to find the solution. Therefore, we consider $[n_{ij}^V = |\mathcal{L}_i \cap R_j|]$ as the cost matrix for a Voting strategy where each element n_{ij}^V represents the number of labelled examples with class value i appearing in R_j and $N^P = [n_{ij}^P = \sum_{\mathbf{x}\in\mathcal{L}_i} \log f_{\pi^{-1}(i)}(\mathbf{x})]$ as the square matrix representing the cost matrix of PC$_{\text{SSL}}$, where each element n_{ij}^P is the log-likelihood of labelled examples with class value i regarding the component $f_{\pi^{-1}(i)}$. Then, by applying the Hungarian algorithm over these cost matrices, the optimal assignment[4] π of labels, given a cost matrix, is achieved in polynomial time ($O(K^3)$). This transformation also solves the original ambiguity of Voting; in [9], further details are not provided about how Voting deals with the ties.

VI. Minimum number of labelled examples

SSL is usually applied in domains where labelled data are very expensive and/or difficult to obtain, but crucial (eq. (6)). For that reason, we think it is necessary to tackle the second issue of Section IV: the minimal number of labelled data needed in Stage 2 of Algorithm 2 to unambiguously determine a permutation. Under the proposed framework, this issue can be translated into the calculation of the minimum number of labelled data needed to uniquely determine one possible permutation π without leaving any possibility to chance. Note that, when there is ambiguity, e_B cannot be reached.

As stated, in binary problems, just one labelled example is enough. It can also be easily seen that, in general, $(K-1)$ labelled instances with different label values are needed to do so and the remaining component is determined by a process of elimination. However, we cannot ensure having $(K-1)$ different labels in a particular labelled set $\mathcal{L}$ due to the randomness of the data [17]. Hence, for multi-class problems, expectations must be taken. We need to calculate l_K, i.e. *the expected minimum number of instances needed to have a labelled set with $(K-1)$ different class values among them.*

A. The expected minimum number of labelled examples

First, we determine l_K under the assumption of having all class priors equiprobable. This calculation is given by:

Theorem 3. ***(Minimum number of labelled examples)*** *Let the family of mixtures $\mathcal{F}$ be linearly independent. Let $K \leq \infty$ be the number of classes and the number of mixture components. Let the class priors be equiprobable, i.e. $\forall i, \eta_i = \eta = 1/K$. Then, the expected minimum number of labelled instances needed to uniquely determine the class labels of the K components of a mixture is :*

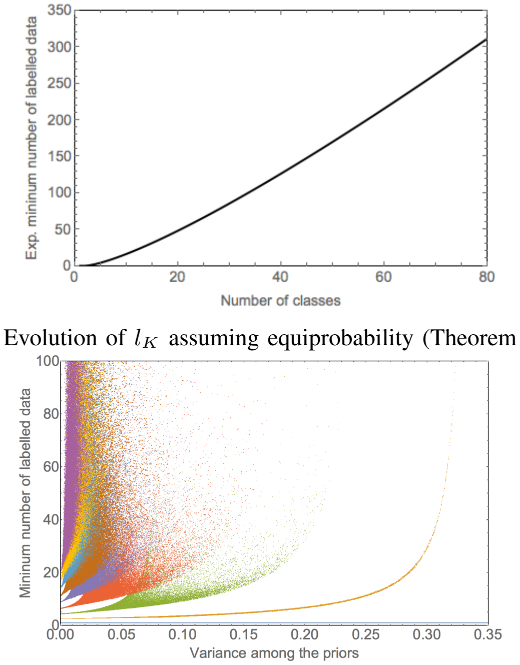
$$l_K = \begin{cases} 1 & \text{if } K = 2 \\ \sum_{j=2}^{K-1} (-1)^j \binom{K}{j} (j-1)\times & \text{if } K > 2 \\ \times\Big(\frac{K-j}{K}\Big)^{(K-2)} \Big(K - 1 + \frac{(K-j)}{j}\Big) & \end{cases} \tag{9}$$

Figure 1a shows the growth of l_K for $K = \{2, \ldots, 80\}$ when the priors are equiprobable, i.e. $\eta_i = 1/K, \forall i$. It can be seen that it grows linearly in the number of classes K. This growth is due to the fact that this assumption among the priors is a hard constraint for the minimum labelled examples required. However, we want to remark the main benefit of calculating l_K for equiprobable priors; it is the lower expected bound of labelled examples needed for any possible configuration of K different class priors. For that reason, in practise, the study of l_K gains great importance for high values of K, such as in the recently proposed highly multi-class scenario [18], where $K > 1,000$. In [18], the authors deal with the problem of image classification in a supervised manner. However, since huge amounts of unlabelled images can be easily gathered, it is a matter of time to make use of unlabelled data in highly multi-class problems, as in [4] [19]. For such problems, l_K can be of vital importance for being a lower bound of the required labelled data. As an illustration, Table II presents, for each dataset used in [18], its correspondent l_K.

B. Relations between the class priors and l_K

Now, we relax the assumption of equiprobability. In the first place, we start calculating l_K for ternary problems:

[4] Do not confuse the optimal solution for a linear assignment problem with the optimal probability of error. Under this setting, PC$_{\text{SSL}}$ remains as an optimal algorithm and Voting as a sub-optimal algorithm.



(a) Evolution of l_K assuming equiprobability (Theorem 3).

(b) The growth of $l_2, \dots, l_{10}$ (lower populations represents lower values of K) as $Var(\boldsymbol{\eta})$ increases (Monte Carlo method).

Figure 1: Minimal number of labelled data, l_K.

Problem	K	l_k
16K ImageNet	$15,589$	$143,911$
22K ImageNet	$21,841$	$208,992$
21K WebData	$21,171$	$201,921$
97K WebData	$96,812$	1.07×10^6

TABLE II: l_K for highly multi-class problems [18].

Theorem 4. ***(Minimum labelled examples for ternary problems)*** *Let $f(\mathbf{x})$ be identifiable and let $K = 3$ be the number of classes with priors $\eta_i > 0$, $\sum_{i=1}^{3} \eta_i = 1$. Then, the expected minimum number of labelled instances needed to uniquely determine the class labels of the components of the mixture is*

$$l_3 = 2 + \sum_{i=1}^{3} \frac{\eta_i^2}{1 - \eta_i}. \tag{10}$$

When we want to determine l_K for $K \geq 4$ with non-equiprobable class priors, it becomes intractable. Therefore, in order to avoid such a combinatorial explosion and to obtain an idea of the evolution of l_K with respect to the priors, we perform an empirical study to determine the growth of l_K when the class priors are non-equiprobable for the cases $K = \{2, \dots, 10\}$; We generate a population of size $50,000$ independent samplings by a Dirichlet distribution with all its K hyper parameters set to 1. Since the Dirichlet distribution is the conjugate of the multinomial distribution [20], each sample of the population represents a vector of class prior probabilities $\boldsymbol{\eta} = (\eta_1, \dots, \eta_K)$ uniformly distributed along the domain of $\boldsymbol{\eta}$. Then, for each sample $\boldsymbol{\eta}$, we calculate l_K by means of a Monte Carlo method by averaging the minimum number of instances needed to have $(K-1)$ different classes over $10,000$ independent samplings. In order to be able to show the results in a two-dimensional figure, we also calculate the variance of each sample $\boldsymbol{\eta}$, i.e. $Var(\boldsymbol{\eta}) = \frac{1}{K}\sum_{i=1}^{K}(\eta_i - \bar{\eta})^2$, where $\bar{\eta} = \frac{1}{K}\sum_{i=1}^{K}\eta_i$. Note that the variance is highly correlated to the degree of imbalance among the priors. Then, Figure 1b shows the result of this simulation; how l_K grows as the variance is increased. There, it can be seen that the lowest value of l_K for every K always fits with the equiprobability ($Var(\boldsymbol{\eta}) = 0$), and from that point all the l_K values exponentially grow as the variance is increased. Only for $K = 2$, it remains constant. Here, it can be clearly noticed that the multi-class framework is much harder, at least in the number of labelled examples needed, than the binary scenario.

VII. Probability of error in the multi-class framework

In this section, we deal with the last highlighted concern of Section IV; determining the probability of error $P_e(l, \infty)$ in the multi-class scenario under the correct model assumption.

In binary problems, the probability of error is calculated by exploiting the inherent characteristic of Algorithm 1: a classification error happens when either Stage 2 or 3 of Algorithm 1 yields an incorrect answer; but when both stages result in wrong answers, the mistakes cancel each other out [1], [2]. Unfortunately, this characteristic does not apply for the multi-class scenario. Here, the casuistry gets more complex; a classification error also occurs when either Stage 2 or Stage 3 of Algorithm 2 yields an incorrect answer. However, when both stages result in wrong answers, the mistakes do not necessarily cancel each other out. What's more, most of the mistakes in both stages lead to a final misclassification. Driven by these thoughts, we calculate the probability of error when a determined labelled subset $\mathcal{L}$ is given to derive the obtained results to the case when just the number of labelled records l is given. The following lemma formulates the probability of error for any learning procedure for Stage 2, including Voting and PC$_{\text{SSL}}$, when the BDR is applied over a returned permutation:

Lemma 1. *Let $\mathcal{L}$ be a labelled subset distributed according to a generative model $f(\mathbf{x}, c)$. Let the marginal of $f(\mathbf{x}, c)$ on $\mathbf{x}$ be an identifiable mixture density $f(\mathbf{x}) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x})$ which represents the distribution of the infinite unlabelled records. Let π be the correspondence returned by the learning procedure $\Pi(\cdot)$, (Stage 2 of Algorithm 2), and let $R_{\pi^{-1}(i)}$ be defined as in eq. (5). Then, the probability of committing an error in classifying an unseen instance with the BDR after assuming the correspondence π is*

$$P(e|\pi) = 1 - \sum_{i=1}^{K} \eta_i \int_{R_{\pi^{-1}(i)}} f_i(\mathbf{x}) d\mathbf{x}. \tag{11}$$

Although the previous lemma formulates the probability of error of Stage 3 in Algorithm 2 independently of the learning procedure used for Stage 2, we are interesting in calculating the probability of error of the whole procedure for a given number of labelled data l:

Theorem 5. ***(Probability of error)*** *The probability of error of classifying an unseen instance in the multi-class scenario,* *given l labelled records and infinite unlabelled records, is*[5]*:*

$$P_e(l,\infty) = \sum_{\pi \in S_K} P(\Pi_l = \pi) P(e|\pi), \tag{12}$$

where $P(\Pi_l = \pi)$ denotes the probability of choosing, using the learning procedure $\Pi(\cdot)$, the permutation π given l labelled data (Stage 2 of Algorithm 2) and $P(e|\pi)$ the probability of misclassification with the BDR after assuming the correspondence π (Stage 3 of Algorithm 2). Finally, S_K represents the set of all possible permutations of size K representing all the correspondences between labels and components.

Proof: First, we define $\mathbb{L} = \{\mathcal{L} \mid |\mathcal{L}| = l\}$ as the set containing all the possible labelled subsets with cardinality l and formulate $P_e(l,\infty)$ as

$$P_e(l,\infty) = \int_{\mathbb{L}} P(e|\mathcal{L}) P(\mathcal{L}) d\mathcal{L}.$$

Unfortunately, the number of labelled sets with cardinality l is infinite (except for the case of $l = 0$). Therefore, we need to rewrite this equation by partitioning $\mathbb{L}$ into several disjoint sets $\mathbb{L}_\pi$, i.e. $\mathbb{L} = \bigcup_{\pi \in S_K} \mathbb{L}_\pi \wedge \forall a, b, \mathbb{L}_{\pi_a} \cap \mathbb{L}_{\pi_b} = \emptyset$. Each $\mathbb{L}_\pi$ stands for $\{\mathcal{L} \mid |\mathcal{L}| = l \wedge \Pi(\mathcal{L}) = \pi\}$. Then, by the distribution property, the probability can be rewritten as

$$P_e(l,\infty) = \sum_{\pi \in S_K} \int_{\mathbb{L}_\pi} P(e|\mathcal{L}) P(\mathcal{L}) d\mathcal{L},$$

In this case, the probability $P(e|\mathcal{L}), \forall \mathcal{L} \in \mathbb{L}_\pi$ will be equal to the $P(e|\pi)$ (eq. (11)) since the returned permutation is π. Due to the fact that $P(e|\pi)$ is constant, it can be extracted from the integrand as a common factor. Finally, as $P(\Pi_l = \pi)$ is, by definition $\int_{\mathbb{L}_\pi} P(\mathcal{L}) d\mathcal{L}$, we rewrite the formula as that presented in the theorem. ■

However, we are interested in calculating, under certain assumptions, the convergence rate of the probability of error in the multi-class framework. When possible, we also calculate a formula depending on l and K. For that reason, in the following sections we calculate it for two scenarios; (i) when there are no pairwise intersections among the components and (ii) when the components intersect among themselves.

A. Mutually disjoint components

When the pairwise intersection among the components is empty, there is no chance of committing an error using the BDR in the supervised scenario, i.e. $e_B = 0$. However, in the SSL framework, a classification error may occur when the correspondence π is determined (Stage 2).

In order to calculate the error, we take advantage of the main property of this scenario; with just one labelled datum of the class c_i in the labelled subset, we can unequivocally determine $\pi_c(j)$. Therefore, the calculation of the error turns into the calculation of the probability that a certain number $z \le \min(K, l)$ of labels appear in l labelled records. Under the assumption of having all priors equiprobable, the probability of error is calculated based on this reasoning as follows:

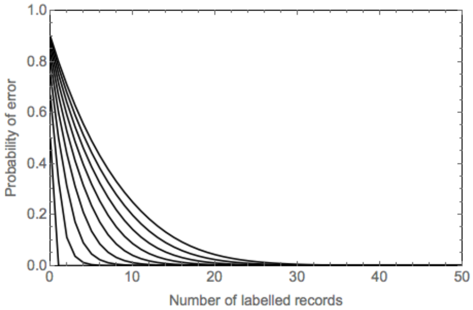


Figure 2: $P_e(l,\infty)$ for $K = \{2,\ldots,10\}$ (lower lines are lower values of K) for mutually disjoint components ($e_B = 0$).

Theorem 6. ***(Probability of error with zero Bayes error)*** *Let the mixture density $f(\mathbf{x})$ be an identifiable mixture. Let K be the number of classes and the number of mixture components. Let the class priors be equiprobable, i.e. $\forall i, \eta_i = \eta = \frac{1}{K}$. Then, the probability of error $P_e(l,\infty)$, given $l > 0$ labelled records and infinite unlabelled records when the components are mutually disjoint is given by:*

$$P_e(l,\infty) = \sum_{z=1}^{\min(K-2,l)} \frac{P_{K,z} S_2(l,z)}{PR_{K,l}} \Big(1 - \frac{z+1}{K}\Big), \tag{13}$$

where $P_{K,z}$ and $PR_{K,l}$ are the number of z-permutations without repetition and l-permutations with repetition of K, respectively. $S_2(l,z)$ is the Stirling number of 2nd kind [21].

Corollary 1. *When the components are mutually disjoint, $P_e(l,\infty)$ converges to 0 exponentially fast in the sense of that*

$$o\left(\left(\frac{K-1}{K}\right)^l\right). \tag{14}$$

Figure 2 illustrates the variation of the error under the assumptions of models with equiprobable priors and $e_B = 0$ for K from 2 to 10. There, it can be seen that the probability of error converges exponentially fast to zero in l. Also, note that, in this scenario, both Voting and PC_{SSL} are equivalent. Both of them obtain the optimal probability of error (eq. (13)).

B. Mutually non-disjoint components

When the $e_B > 0$, we provide an upper bound (Theorem 7) of $P_e(l,\infty)$ in order to determine the convergence rate in l of the optimal probability of error (Corollary 2). Here, we assume having all priors equiprobable, $\forall i, \eta_i = \eta = 1/K$.

Theorem 7. ***(Upper bound of the error)*** *When the components are not mutually disjoint and all the priors are equiprobable, the optimal probability of error of a model composed by K different identifiable mixture components is upper bounded by*

$$P_e(l,\infty) - e_B \le 2\exp\Big\{\frac{-l\lambda^2}{2K}\Big\}, \tag{15}$$

where $\lambda \in (0,1]$ depends on the degree of intersection among

[5] Note that eq. (12) fits with the one proposed for binary problems in [1] (pp. 107, eq. (7)): $P(\Pi_l = \pi_c)P(e|\pi_c) + P(\Pi_l = \bar{\pi}_c)P(e|\bar{\pi}_c)$.

the components of the mixture distribution and is defined by

$$\lambda = \frac{1}{K}\min_{j}\Big\{\int_{R_j} f_i(\mathbf{x})d\mathbf{x} - \max_{z\neq i}\int_{R_j} f_z(\mathbf{x})d\mathbf{x}\Big\}. \quad (16)$$

Corollary 2. *The probability of error, $P_e(l,\infty)$, decreases to e_B, at least, exponentially fast in the number of labelled data.*

The previous calculi prove that the optimal probability of error converges exponentially fast in l multiplied by a constant $\lambda \in (0,1]$. The latter only depends on the intrinsic characteristics of the components of the mixture; whilst models with mutually disjoint components show a value of $\lambda = 1$, models with a high level of overlapping show values of $\lambda \sim 0$. Note that, for values of λ close to 0, the decrease of the probability of error will be slower since, in problems with a high intersection of components, the process of discriminating the classes is intricate. In those cases, more labelled data will be required. However, is this upper bound good enough? Can the probability of error decrease faster? To answer these questions, we provide a lower bound of the optimal probability of error assuming the particular scenario of having a model composed by Gaussian mixture components with the same variance, i.e. $\boldsymbol{\theta} = \{\mu_1,\ldots,\mu_K,\sigma\}$ and each $\boldsymbol{\theta}_i = \{\mu_i,\sigma\}$.

Theorem 8. ***(Lower bound of the error)*** *Assuming that the model is a Gaussian identifiable mixture of K components with the same variance (σ) and equiprobable priors. Let δ_M be the largest distance between the means of two components and $Q(\cdot)$ a polynomial of degree 3. Then, the probability of error for non-disjoint components is lower bounded by*

$$P_e(l,\infty) - e_B \geq \frac{K!(1-e_B)-(K-1)!}{\big(1+\exp\{Q(\frac{\delta_M}{2\sigma}\sqrt{l})\}\big)^{(K!-1)}}. \quad (17)$$

It can be easily noticed that, in the case of assuming a Gaussian mixture, both bounds are quite close, leaving not much room for improvement in the upper bound; they converge exponentially fast in l to e_B. Note also that, in this scenario, $\delta_M \in (0,\infty)$ plays the role of the constant λ in the general solution; values of δ_M close to zero represent problems with a high intersection of components and a slower decrease of the probability of error. This can give us an idea that the optimal probability of error without any assumptions on the model will also converge exponentially fast, not faster. In the general case, it cannot decrease faster than this specific scenario.

VIII. Experimental studies

In the previous sections, we have proposed an optimal learning procedure, PC$_{\text{SSL}}$, for the multi-class problem in the SSL scenario. By Theorem 2, when the correct model assumption is met, any learning procedure, including the previously proposed Voting strategy [9], will achieve an upper or equal probability of error than PC$_{\text{SSL}}$. However, under the same assumption, does PC$_{\text{SSL}}$ significantly outperform Voting, or do both algorithms share a similar probability of error? Moreover, another interesting question arises in this framework; how does PC$_{\text{SSL}}$ behave when the correct model assumption is not met?

To answer these questions a *generative model* with the following characteristics is assumed: since it must be simply enough to be able to fully interpret the results and complex enough to be able to represent real world problems, we assume a generative model composed by K univariate Gaussian identifiable mixture components with unit variances and whose means are separated by a fixed factor $\delta \in \mathbb{R}$, i.e. $\boldsymbol{\theta}_i = \{\mu_i,\sigma_i\}$, where $\mu_i = \delta(i-1)$ and $\sigma_i = 1$. This factor determines the degree of overlapping among the classes. Also, we assume equiprobable class priors[6]. Regarding the *learning procedures*, we make use of both PC$_{\text{SSL}}$ and Voting, as they are, to the best of our knowledge, the only two theoretical procedures proposed in the literature for this problem. The behaviour of both learning procedures is simulated by a Monte Carlo method; the probability of error of each procedure and each value of l is estimated by averaging the resulting probability of error over $10,000$ independent trials (labelled datasets)[7].

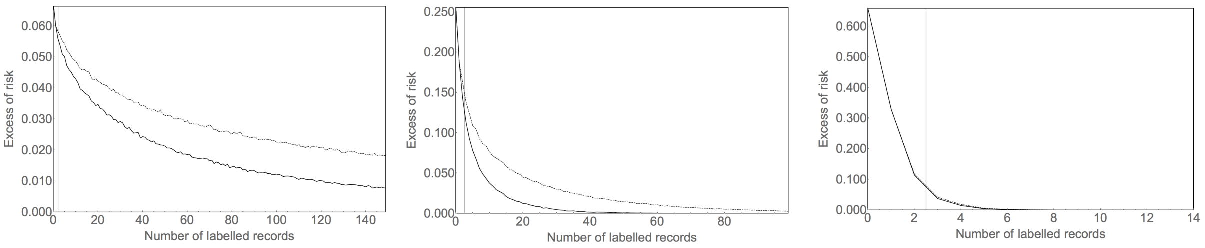
Then, the *first experiment* is carried out to address the first question, i.e. determining whether a significant difference between PC$_{\text{SSL}}$ and Voting exists. To do so, we have studied the behaviour of both procedures assuming the previous generative model for values[8] of $K = \{3,4,5,6\}$. However, due to the limiting space, in this paper, we only show the results for the ternary problems. Higher values of K can be found in Appendix B. Specifically, the probability of error of both PC$_{\text{SSL}}$ ($P_e^P(l,\infty) = P_e(l,\infty)$) and Voting ($P_e^V(l,\infty)$) learning algorithms has been calculated for $l = \{0,\ldots,l_{max}\}$ for three different levels of intersection among the components, $\delta = \{0.25,1,5\}$. These three different δ values can be assumed to correspond to complex, medium and easy problems.

Figure 3 shows the behaviour of the probability of error for PC$_{\text{SSL}}$ and Voting as l increases in the studied ternary problem. Specifically, the different levels of intersection $\delta = \{0.25,1,5\}$ are represented by Figure 3a, Figure 3b, and Figure 3c, respectively. All the figures share the same shape. The x-axis represents the number of labelled data l and the y-axis represents the excess of risk of both procedures [2], i.e. $P_e^V(l,\infty) - e_B$ or $P_e^P(l,\infty) - e_B$. Note that, the x-axis is differently scaled for each problem due to the fact that complex problems require more labelled data (eq. (15)). Moreover, since e_B varies for different values of δ, the y-axis is also not equally scaled for the three scenarios. The corresponding Bayes error values for each $\delta = \{0.25,1,5\}$ are $e_B = \{0.6004, 0.4114, 0.0083\}$, respectively. Then, the upper decreasing curve (grey colour) is the excess of risk of Voting and the lower decreasing curve (black colour) corresponds to PC$_{\text{SSL}}$. The vertical line is l_K. As can be seen, the experiment coincides with the theoretical advances proposed in the paper: (i) the probability of error of both learning algorithms decreases exponentially fast in l (Theorem 7) and PC$_{\text{SSL}}$ always dominates Voting. It always achieves a lower (or equal) probability of error (Theorem 2). (ii) When $e_B \sim 0$, i.e. higher values of δ, both algorithms behave similarly in

[6] Note that if we consider unequal standard deviation, multivariate features, other geometry or non-normal probability densities, it may not be possible to perform all the calculations, e.g. the Bayes error.

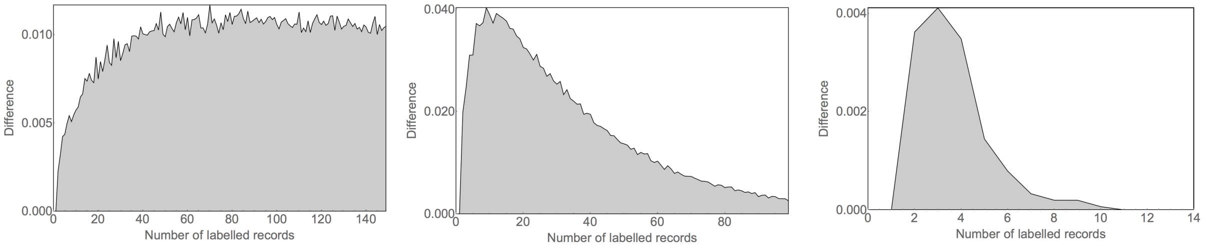
[7] For the sake of honesty, the same datasets are sampled for each procedure and each set of parameters. Moreover, the cases where the correspondence cannot unambiguously be determined are equally resolved for both procedures.

[8] Both learning algorithms are equivalent for binary problems.



(a) Complex problem ($\delta = 0.25$). (b) Medium problem ($\delta = 1$). (c) Easy problem ($\delta = 5$)

Figure 3: Probability of error of Voting [9] (upper) and PC$_{\text{SSL}}$ (lower curve) for $K = 3$ ($l_K = 2.5$).



(a) Complex problem ($\delta = 0.25$). (b) Medium problem ($\delta = 1$). (c) Easy problem ($\delta = 5$)

Figure 4: Absolute differences between Voting and PC$_{\text{SSL}}$, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 3$.

terms of probability of error (Theorem 6). (iii) In the opposite case, i.e. when δ is small, the room for improvement is quite narrow (e.g. $e_0 - e_B \sim 0.06$, for $\delta = 0.25$) and the complexity of the problem is really high. There, the probability of error of any algorithm will show a slower decrease in l (Theorem 7 and 8) and more labelled data will be required to achieve the best classifier. (iv) Finally, the results show that e_B is never achieved with less than l_K labelled examples (Theorem 3).

In order to properly quantify the magnitude of the absolute difference between the two theoretical SSL procedures, we also introduce Figure 4. There, the differences between the probabilities of error of both Voting and PC$_{\text{SSL}}$ are shown for each l. Figures 4a, 4b, and 4c represent the previously defined complex, medium and easy problems, respectively. The y-axis in each figure is scaled between 0 and the highest difference found in the simulation. In general, the differences between the procedure show a similar shape throughout the problems. First, the difference between the probability of error of both theoretical procedures is 0 for both $l = 0$ (Theorem 1) and $l = 1$. After that, it grows until a determined value of l. Finally, beyond that point, the difference starts to decrease to 0, the point where Voting reaches e_B. Additionally, the results also reveal that, although the absolute differences vary for determined values of δ, the relative differences (w.r.t. the available room for improvement, i.e. $e_0 - e_B$) are greater for lower values of δ. Then, we can conclude that PC$_{\text{SSL}}$ achieves a much better relative performance than Voting for low values of δ.

Appendix B shows that the results for that higher values of K ($l_{max} = 50$) are similar to $K = 3$. The main difference is that, for higher multi-class problems, the behaviour of $K = 3$ is horizontally stretched when $l > 1$ due to the fact that both procedures need a higher value of l to reach e_B.

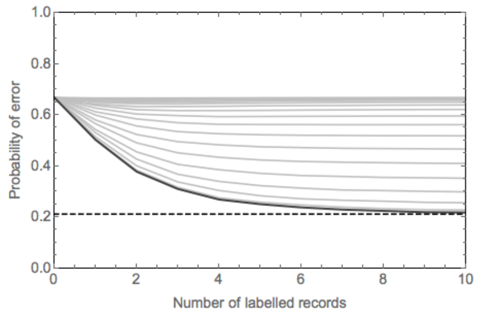


Figure 5: Evolution of the probability of error of PC$_{\text{SSL}}$ when the correct model assumption is not met.

The *second experiment* is devoted to studying the behaviour of PC$_{\text{SSL}}$ when the correct model assumption is not fulfilled, that is, when there is not enough unlabelled data to make a good estimation of the mixture density. To do so, we simulate that an incorrect model is obtained by a simple mechanism: the learnt model is also a K univariate Gaussian identifiable mixture components with unit variances and whose means are separated by a fixed factor δ. The class priors of this problem are also equiprobable. However, it is shifted a υ factor to the left. In this setting, υ varies in an arithmetic progression with a fixed difference of 0.25 from 0 to 5. Figure 5 sums up the behaviour of PC$_{\text{SSL}}$ for $K = 3$ and $\delta = 2.5$ (similar behaviours are found for different configurations). There, the black curve represents the correct model ($\upsilon = 0$), the grey curves correspond to different values of $\upsilon > 0$ (lower lines represent lower values of υ), and the horizontal dashed line is e_B. As can be seen, when this assumption does not hold, e_B can no be reached. When there is a slight difference among the models, reasonable performance can still be achieved, and the labelled data exponentially reduces the probability of error up to a difference ϵ between the asymptotic value of $\mathrm{PC}_{\mathrm{SSL}}$ and e_B [8]. This asymptotic value coincides with the unsupervised MLE, discussed in [15]. However, when υ grows, the reduction displays a more linear behaviour and the difference ϵ becomes higher. At the extreme, here $\upsilon = 5$, the probability of error, practically remains constant in l. This means that, in extreme cases of model misspecification, $P_e(l, u), \forall l, u \sim (K-1)/K$, i.e. the use of the labelled data does not reduce the probability of error [4] [15].

IX. Remarks on the problem-solving in semi-supervised learning and open challenges

Although our main aim is to theoretically study the probability of error in the multi-class scenario, we want not only to discuss the potential impact of the theories presented in the designing of new practical learning algorithms, but also some challenges appearing in both theoretical and practical SSL scenarios. Thus, imagine we want to face a real-world problem using the theoretical advances presented in this paper. There, we basically face three different key questions:

1) How much unlabelled data should we gather? Throughout the paper, we highlight the importance of the labelled records; if the correct assumption is met, they always help in making the labelled examples to reduce the probability of error faster than in supervised learning [4]. When $u = \infty$, the generative model can almost surely be recovered, therefore, the correct model assumption is met. However, in practical SSL, u is always finite. Moreover, there are also generative models which are not identifiable or, even, they do not follow mixture densities. For these practical cases, more general assumptions are used in order to support that the correct model can be learned. The assumptions are the following: smoothness assumption, cluster assumption and manifold assumption[9] [3]. However, these assumptions are hard to check in practise. They can only be tested by a trial-and-error procedure. If these assumptions are not met, the use of unlabelled data cannot guarantee any significant advantages over learning a purely supervised learning problem [23], so SSL techniques are not a good choice to solve the problem. In the opposite case, unlabelled data may help the performance of the classifier. There, we recommend the use of as much unlabelled data available so that a solid estimation of the generative model can be obtained by the learning algorithm and the correct model assumption can be met. Provided we have a good estimation of the model we can overtake, or even mitigate, the problems shown in the second experiment (incorrect model ass.). We emphasise that the problem of meeting the correct model assumption is still a challenging crucial issue. Another possible challenge for future work regarding the unlabelled data could be solved by [8], but assuming the correct model: Is there any number beyond which any extra additions of unlabelled data do not decrease the probability of error? In other words, which real number, in practise, corresponds to the infinite number of unlabelled records, broadly used in theoretical works.

2) How much labelled data is required? In order to avoid making assumptions about the generative model, when the labelled data are neither expensive nor difficult to obtain, we strongly believe that supervised learning techniques are more appropriate. In the opposite case, if the correct model assumption is met, we have proved that, in general, e_B is never achieved for labelled sets with a cardinality lower than l_K as expressed in Theorem 3. This holds true independently of the degree of imbalance and the degree of intersection among the components. Therefore, a higher number of labelled records must be collected. However, when the degree of imbalance grows, more labelled are required for the same purpose (l_K for non-equiprobable priors). Analogously, for problems with a high intersection, the decrease of the probability of error is slower and, although l_K expects that e_B can be reached, a much higher number of labelled data is probably required to reach it. So, we can conclude that this question is still a challenging issue. For this reason, we think that it is interesting to, in the future, propose sample complexities for l, not only on the unbalanced degree of the priors, but also on the complexity and dimensionality of the feature space. Sample complexities seem to be crucial in SSL, where labelled data are scarce.

3) Which SSL learning procedure can be used? In cases where the family of the generative model is known and the number of unlabelled examples is enough to obtain a good estimation, Algorithm 2 can be directly applied to the problem. There, both $\mathrm{PC}_{\mathrm{SSL}}$ and Voting [9] can be used as learning procedures. However, PC_{SSL} seems to be a more appropriate choice, not only due to its theoretical properties, but also for matching the time complexity of Voting. On the contrary, when the family of the generative model is unknown, we cannot use the generative densities. In those cases, we can use the theoretical advances of this paper to design a practical algorithm for Stage 2. For problems with linear decision boundaries, such as the Gaussian classification of the experiments, a simple procedure for determining the components can be proposed based on the nearest-centroid classifier [24]. However, instead of classifying the data, the labelled samples can be used to determine the label of the centroids. Formally, by sphering each centroid and classifying it according to the class values of the labelled data in that sphered space. In a manner analogous to the theoretical methodology proposed in this paper, in this very case, we can also follow a Voting methodology by counting the majority class of labelled data in the sphered space or the PC_{SSL}, determining the minimum distance in the possible permutations. This can be another interesting potential future work; proposing practical learning procedures for both linear and non-linear decision boundaries. Finally, and within this framework, it could be also interesting to investigate a competitive, or even optimal, procedure to correctly specify the classifier, using labelled data, when the correct model assumption is not met, similarly to [8] and [9].

X. Conclusion

In this paper, we perform a study on the SSL multi-class framework, since most of the works deal with just binary

[9] **Smoothness assumption:** Points which are close to each other are likely to share a label. **Cluster assumption:** The data tend to form discrete clusters, and points in the same cluster are more likely to share a label. **Manifold assumption:** The data lie approximately on a manifold of much lower dimension than the input space.

problems [1] [2]. For that reason, we take it a step further by extending the work of Castelli and Cover [1] [2] to the multi-class paradigm. Particularly, we consider the key problem in SSL of classifying an unseen instance $\mathbf{x}^{(0)}$ into one of K different classes, using a training dataset composed of l labelled records and $u = \infty$ unlabelled examples. However, the previous studies do not straightforwardly work for multi-class problems, so, in this paper, we make three main contributions: (i) PC_{SSL}, an optimal theoretical multi-class learning algorithm for SSL problems, is proposed. (ii) We investigate the expected minimum number, l_K, of labelled data needed to determine the K decision regions. (iii) We study the optimal probability of error when the binary constraint is relaxed, concluding that labelled data exponentially reduces the probability of error. A discussion on the impact of our proposals in solving real-world problems finalises the paper.

Acknowledgment

This work is partially supported by the Basque Government (IT609-13), the Spanish Ministry of Science and Innovation (TIN2013-41272P and MEC-FPU AP2008-00766), the Carlos III Health Institute (COMBIOMED), Diputación Foral de Gipuzkoa (2011-CIEN-000060-01), the European Commission (NICaiA PIRSESGA-2009-247619), and Diputación Foral de Bizkaia (6-12-TK-2011-014). We also thank Aritz Pérez, Ekhiñe Irurozki and Leticia Hernando for their comments.

References

[1] V. Castelli and T. Cover, "On the exponential value of labeled samples," *Pattern Recognition Letters*, vol. 16, pp. 105–111, 1995.

[2] ——, "The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2102–2117, 1996.

[3] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised learning*. The MIT Press, 2006.

[4] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: Theory, algorithms and their application to human-computer interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1553–1567, 2004.

[5] R. Xu, G. Anagnostopoulos, and D. Whunsch II, "Multi-class cancer classification by semi-supervised ellipsoid artmap with gene expression data," in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2004.

[6] J. Ortigosa-Hernández, J. Rodríguez, L. Alzate, M. Lucania, I. Inza, and J. A. Lozano, "Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers," *Neurocomputing*, vol. 92, pp. 98–115, 2012.

[7] J. Ratsaby and S. Venkatesh, "Learning from a mixture of labeled and unlabeled examples with parametric side information," in *Proceedings of the eighth annual conference on Computational learning theory (COLT '95)*, 1995, pp. 412–417.

[8] K. Sinha and M. Belkin, "The value of labeled and unlabeled examples when the model is imperfect," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. The MIT Press, 2008.

[9] H. Chen and L. Li, "Semisupervised multicategory classification with imperfect model," *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1594–1603, 2009.

[10] G. M. Tallis and P. Chesson, "Identifiability of mixtures," *Journal of the Australian Mathematical Society (Series A)*, no. 32, pp. 339–348, 1982.

[11] B. Everitt and D. Hand, *Finite Mixture Distributions*. Chapman and Hall, 1981.

[12] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, 1985.

[13] B. Grun and F. Leisch, "Identifiability of finite mixtures of multinomial logit models with varying and fixed effects," Department of Statistics, University of Munich, Tech. Rep. 024, 2008.

[14] T. Zhang and F. Oles, "A probability analysis on the value of unlabeled data for classification problems," in *Proceedings of the International Conference on Machine Learning (ICML'2000)*, 2000, pp. 1191–1198.

[15] T. Yang and C. Priebe, "The effect of model misspecification on semi-supervised classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2093–2103, 2011.

[16] P. Rigollet, "Generalization error bounds in semi-supervised classification under the cluster assumption," *Journal of Machine Learning Research*, vol. 8, pp. 1369–1392, 2007.

[17] P. Fox-Roberts and R. E., "Unbiased generative semi-supervised learning," *Journal of Machine Learning Research*, vol. 15, pp. 367–443, 2014.

[18] M. Gupta, S. Bengio, and J. Weston, "Training highly multiclass classifiers," *Journal of Machine Learning Research*, vol. 15, pp. 1461–1492, 2014.

[19] H. Wang, H. Huang, and C. Ding, "Image annotation using multi-label correlated greens function," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009, pp. 2029–2034.

[20] J. M. Bernardo, M. H. DeGroot, D. Lindley, and A. F. M. Smith, *Bayesian Statistics*. Valencia University Press, 1980.

[21] R. P. Stanley, *Enumerative combinatorics*. Wadsworth Publ. Co., 1986.

[22] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, no. 2, pp. 83–95, 1955.

[23] S. Ben-David, T. Lu, and D. Pal, "Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning," in *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, 2008.

[24] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

Jonathan Ortigosa-Hernández recieved an MSc degree in Computer Science from *The University of the Basque Country* in 2008, and a BSc in Informatics from *Coventry University* in the same year. Since 2009, he has been a PhD student in *The University of the Basque Country* and a member of the *Intelligent Systems Group* research team. His research interests are multi-dimensional classification, semi-supervised learning, and unbalanced problems. (*http://www.sc.ehu.es/ccwbayes/members/jonathan*)

Iñaki Inza received his PhD in Computer Science from *The University of the Basque Country* in 2002 and he is a member of the *Intelligent Systems Group* research team. He is an associate professor in *The University of the Basque Country*. Some of his main interests are in feature selection methods, Bayesian networks and applications to biological domains. He has 8 book chapters in 6 books and 31 publications. (*www.sc.ehu.es/ccwbayes/members/inaki.htm*)

Jose A. Lozano received a PhD degree in Computer Science from the *University of the Basque Country* in 1998. Since 2008 he is a full professor in the same university, where he leads the *Intelligent Systems Group*. He is the co-author of more than 50 ISI journal publications and co-editor of 3 books. His major research interests include machine learning, pattern analysis, evolutionary computation, data mining and metaheuristic algorithms. Prof. Lozano is a member of the editorial board of six journals. (*www.sc.ehu.es/ccwbayes/members/jalozano*).

Appendices for the article entitled "*Semi-supervised multi-class classification problems with scarcity of labelled data*"

Jonathan Ortigosa-Hernández, Iñaki Inza, and Jose A. Lozano

Contents

1 Appendix A. Mathematical proofs 2

1.1 Theorem 1 2

1.2 Theorem 2 3

1.3 Theorem 3 3

1.4 Theorem 4 4

1.5 Lemma 1 4

1.6 Theorem 5 5

1.7 Theorem 6 5

1.8 Corollary 1 6

1.9 Theorem 7 7

1.10 Corollary 2 8

1.11 Theorem 8 8

2 Appendix B. Supplementary data for the empirical study 12

3 Appendix C. Source code 15

3.1 l_k-related functions 15

3.2 Functions related to the probability of error 16

3.3 Examples of the executions of the package 18

1 Appendix A. Mathematical proofs

This appendix provides the mathematical proofs of the theoretical results in our paper **"Semi-supervised multi-class classification problems with scarcity of labelled data: A theoretical study"**

1.1 Theorem 1

Theorem 1. ***(Probability of error with no labelled data)***[1] *The probability of error of classifying a new sample $(\mathbf{x}^{(0)}, c^{(0)})$ of a K-class problem with any classifier learnt with no labelled examples and any $u \geq 0$ number of unlabelled samples coincides with the probability of error random classifier. In both cases, it remains constant to*

$$P_e(0,u) = e_0 = \frac{(K-1)}{K},\ \forall u \geq 0. \tag{1}$$

Proof.

- **When $f(\mathbf{x})$ is unknown:** When there are not enough unlabelled records ($u < \infty$) to determine the mixture density (and its components), the class $c^{(0)}$ of the unseen distance must be determined by the uniformly random classifier. In such a case, let the event A be defined as the probability of correctly choosing the right class for $c^{(0)}$ over a choice of K different classes; $P(A) = 1/K$. Then, the probability of committing an error is

$$P_e(0,u) = 1 - P(A) = \frac{(K-1)}{K}, \forall u < \infty.$$

- **When $f(\mathbf{x})$ is known:** With ∞ unlabelled examples, the mixture is identifiable up to a permutation π of the components. Let the observation $\mathbf{x}^{(0)}$ be drawn from the j-th decomposed component, $f_{\pi(j)}(\cdot)$, i the unknown true label such that $\pi_c(j) = i$, and let us define the following two events:

$$B_1 \triangleq \{\eta_{\pi_c(j)} f_{\pi_c(j)}(\mathbf{x}^{(0)}) - \max_{j' \neq j} \eta_{\pi_c(j')} f_{\pi_c(j')}(\mathbf{x}^{(0)}) > 0\}$$
$$B_2 \triangleq \{\eta_{\pi_c(j)} f_{\pi_c(j)}(\mathbf{x}^{(0)}) - \max_{j' \neq j} \eta_{\pi_c(j')} f_{\pi_c(j')}(\mathbf{x}^{(0)}) < 0\}$$

B_1 is the event which represents achieving a correct answer in the application of the Bayes decision rule over $\mathbf{x}^{(0)}$ and B_2 represents the opposite- By the definition, $P(B_1) = (1 - e_B)$ and $P(B_2) = e_B$. Then, the probability of error is

$$\begin{aligned} P_e(0,\infty) &= P(\hat{c}^{(0)} \neq c^{(0)}) = 1 - P(\hat{c}^{(0)} = c^{(0)}) = \\ &= 1 - P(\hat{c}^{(0)} = c^{(0)}|B_1)P(B_1) - P(\hat{c}^{(0)} = c^{(0)}|B_2)P(B_2) = \\ &= 1 - P(\pi_a)(1 - e_B) - P(\pi_a)e_B. \end{aligned}$$

[1] This holds true independently of the value of the class priors $\eta_1, \ldots, \eta_K$.

where π_a and π_b are two permutations such that $\pi_a(j) = i$ and $\pi_b(j') = i$.As no labelled data are provided to determine the correspondence π, it has to be randomly chosen. Then, the probability of choosing those permutations is $P(\pi_a) = P(\pi_b) = (K-1)!/K! = 1/K$. After substituting these probabilities in the previous formula and after some algebra, we obtain the same expression:

$$P_e(0,\infty) = \frac{(K-1)}{K}.$$

□

1.2 Theorem 2

Theorem 2. ***(Optimality of PC$_{SSL}$)*** *PC$_{SSL}$ is an optimum learning procedure for Stage 2 of Algorithm 2.*

Proof. The proof of this theorem can be found in the manuscript. □

1.3 Theorem 3

Theorem 3. ***(Minimum number of labelled examples)*** *Let the family of mixtures $\mathcal{F}$ be linearly independent. Let $K \leq \infty$ be the number of classes and the number of mixture components. Let the class priors be equiprobable, i.e. $\forall i, \eta_i = \eta = 1/K$. Then, the expected minimum number of labelled instances needed to uniquely determine the class labels of the K components of a mixture is :*

$$l_K = \begin{cases} 1 & \textit{if } K = 2 \\ \sum_{j=2}^{K-1} (-1)^j \binom{K}{j} (j-1) \times & \textit{if } K > 2 \\ \times \Big(\frac{K-j}{K}\Big)^{(K-2)} \Big(K - 1 + \frac{(K-j)}{j}\Big) & \end{cases} \tag{2}$$

Proof. The proof for $K = 2$ can be found in [1]. For higher values of K, let L_K be a random variable representing the minimum number of instances needed to obtain examples of $(K-1)$ different classes. Assuming equiprobability, $P(L_K = l)$, for $l \geq (K-1)$, is a fraction whose numerator is the number of favourable cases and whose denominator is the number of all possible cases:

$$P(L_K = l) = \frac{P_K S_2(l-1, K-2)}{PR_{K,l}},$$

where P_K is the number of permutations of K elements, $S_2(\cdot,\cdot)$ is the Stirling number of second kind, and $PR_{K,l}$ is the number of l-permutations of K elements with repetition (formulae in [2]). Then, we calculate the expectation of the random variable L_K in l as $l_K = E[L_K] = \sum_{l=K-1}^{\infty} lP(L_K = l)$. After some algebra, we reach equation (2), for $K > 2$. □

1.4 Theorem 4

Theorem 4. ***(Minimum labelled examples for ternary problems)*** *Let $f(\mathbf{x})$ be identifiable and let $K = 3$ be the number of classes with priors $\eta_i > 0$, $\sum_{i=1}^{3} \eta_i = 1$. Then, the expected minimum number of labelled instances needed to uniquely determine the class labels of the components of the mixture is*

$$l_3 = 2 + \sum_{i=1}^{3} \frac{\eta_i^2}{1 - \eta_i}. \tag{3}$$

Proof. Let L_3 be a random variable representing the minimum number of instances needed to obtain two different labels. Assuming each class c_i has a prior η_i and $\sum_{i=1}^{3} \eta_i = 1$, $P(L_3 = l)$ has the following form:

$$P(L_3 = l) = \eta_1^{(l-1)}(1 - \eta_1) + \eta_2^{(l-1)}(1 - \eta_2) + \eta_3^{(l-1)}(1 - \eta_3).$$

Which stands for the probability of having $(l - 1)$ labelled examples of one class, and just one example of one of the two other classes. Then, we calculate the expectation of L_3 in l. After some algebra, we reach the equation (3). □

1.5 Lemma 1

Lemma 1. *Let $\mathcal{L}$ be a labelled subset distributed according to a generative model $f(\mathbf{x}, c)$. Let the marginal of $f(\mathbf{x}, c)$ on $\mathbf{x}$ be an identifiable mixture density $f(\mathbf{x}) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x})$ which represents the distribution of the infinite unlabelled records. Let π be the correspondence returned by the learning procedure $\Pi(\cdot)$, (Stage 2 of Algorithm 2), and let $R_{\pi^{-1}(i)}$ be defined as*

$$R_{\pi^{-1}(i)} = \{\mathbf{x} : \eta_{\pi^{-1}(i)} f_{\pi^{-1}(i)}(\mathbf{x}) - \max_{i' \neq i} \eta_{\pi^{-1}(i')} f_{\pi^{-1}(i')}(\mathbf{x}) > 0\}.$$

Then, the probability of committing an error in classifying an unseen instance with the BDR after assuming the correspondence π is

$$P(e|\pi) = 1 - \sum_{i=1}^{K} \eta_i \int_{R_{\pi^{-1}(i)}} f_i(\mathbf{x}) d\mathbf{x}. \tag{4}$$

Proof. According to Algorithm 2, a correct prediction occurs depending on whether the optimal classification rule, which assumes a learnt correspondence π, over the unseen instance $\mathbf{x}$, hits the real class value (c_i). If the region where $f_i(\mathbf{x})$ is maximum is R_j (it holds that $i = \pi_c(j)$), there are only two cases in which the real class is correctly predicted:

$$\begin{array}{llll} \text{Case 1: if } \mathbf{x} \in R_j & \wedge & \pi(j) & = i \\ \text{Case 2: if } \mathbf{x} \in R_{j' \neq j} & \wedge & \pi(j') & = i \end{array}$$

As can be seen, in both cases, the region to where $\mathbf{x}$ belongs can be named as $\pi^{-1}(i)$, which is equal to j in Case 1, and to j' in Case 2. Therefore, the probability of correctly classifying $\mathbf{x}$ is just the probability of $\mathbf{x}$ being in the region labelled as i, i.e. $R_{\pi^{-1}(i)}$ and this formula can be easily generalised to all the possible values of i in the range $\{1, \dots, K\}$ as:

$$P(\bar{e}|\pi) = \sum_{i=1}^{K} \eta_i \int_{R_{\pi^{-1}(i)}} f_i(\mathbf{x}) d\mathbf{x},$$

After that we can reach formula (4) taking into account that the probability of error is the opposite to the probability of a correct classification, $P(\bar{e}|\pi) = 1 - P(e|\pi)$. □

1.6 Theorem 5

Theorem 5. ***(Probability of error)*** *The probability of error of classifying an unseen instance in the multi-class scenario, given l labelled records and infinite unlabelled records, is:*

$$P_e(l, \infty) = \sum_{\pi \in S_K} P(\Pi_l = \pi) P(e|\pi), \tag{5}$$

where $P(\Pi_l = \pi)$ denotes the probability of choosing, using the learning procedure $\Pi(\cdot)$, the permutation π given l labelled data (Stage 2 of Algorithm 2) and $P(e|\pi)$ the probability of misclassification with the BDR after assuming the correspondence π (Stage 3 of Algorithm 2)[2]. Finally, S_K represents the set of all possible permutations of size K representing all the correspondences between labels and components.

Proof. The mathematical proof of this theorem is in the manuscript. □

1.7 Theorem 6

Theorem 6. ***(Probability of error with zero Bayes error)*** *Let the mixture density $f(\mathbf{x})$ be an identifiable mixture. Let K be the number of classes and the number of mixture components. Let the class priors be equiprobable, i.e. $\forall i, \eta_i = \eta = \frac{1}{K}$. Then, the probability of error $P_e(l, \infty)$, given $l > 0$ labelled records and infinite unlabelled records when the components are mutually disjoint is given by:*

$$P_e(l, \infty) = \sum_{z=1}^{\min(K-2,l)} \frac{P_{K,z} S_2(l, z)}{PR_{K,l}} \Big(1 - \frac{z+1}{K}\Big), \tag{6}$$

where $P_{K,z}$ and $PR_{K,l}$ are the number of z-permutations without repetition and l-permutations with repetition of K, respectively. $S_2(l, z)$ is the Stirling number of 2^{nd} kind[2].

[2] Note that eq. (5) fits with the one proposed for binary problems in [1] (pp. 107, eq. (7)): $P(\Pi_l = \pi_c)P(e|\pi_c) + P(\Pi_l = \bar{\pi}_c)P(e|\bar{\pi}_c)$.

Proof. To determine the probability of error, we just need to calculate the probability of finding $z \leq \min(K, l)$ different labels in l. When $z \geq (K-1)$ (see definition of l_K) we reach the real model, so, the probability of error is e_B, which is zero. Therefore, we can define the probability as follows:

$$P_e(l, \infty) = \sum_{z=1}^{\min(K-2,l)} P(\Psi_l = z) P(e|z), \tag{7}$$

where Ψ_l is a random variable representing the number of different labels, z in l and $P(e|z)$ is the probability of committing a classification error knowing the real correspondence of z labels with their components.

Regarding $P(\Psi_l = z)$, as all the priors are equiprobable; it is a fraction whose numerator is the number of selecting just z classes of K multiplied by the way of ordering them, and whose denominator is the number of all possible cases:

$$P(\Psi_l = z) = \frac{P_{K,z} S_2(l, z)}{PR_{K,l}}.$$

Then, $P(e|z)$ can be decomposed into the sum of the probability of misclassifying when we find a particular label among the labelled subset and the probability of misclassifying it when we do not have that label to identify a mixture: $P(e|z) = \Pr\{i \in z\} \times \Pr\{e|i\} + \Pr\{i \notin z\} \times \Pr\{e|i\}$, where $\{i \in z\}$ corresponds to the event that the unseen instance has the same label as one of the labels that appears in the labelled subset, $\{i \notin z\}$ is the opposite, and $\Pr\{e|i\}$ is the probability of misclassifying an instance of class i. By solving the previous formula, we obtain that $P(e|z)$ is equal to

$$\frac{z}{K} \times 0 + \frac{K-z}{K} \times \frac{(K-z)! - (K-z-1)!}{(K-z)!} = 1 - \frac{z+1}{K}. \tag{8}$$

Finally, by substituting the calculations of $P(\Psi_l = z)$ and $P(e|z)$ in equation (7), we reach formula (6). □

1.8 Corollary 1

Corollary 1. *When the components are mutually disjoint, $P_e(l, \infty)$ converges to* 0 *exponentially fast in the sense of that*

$$o\left(\left(\frac{K-1}{K}\right)^l\right). \tag{9}$$

Proof. It is trivial to calculate the convergence order from the previous theorem. □

1.9 Theorem 7

Theorem 7. ***(Upper bound of the error)*** *When the components are not mutually disjoint and all the priors are equiprobable, the optimal probability of error of a model composed by K different identifiable mixture components is upper bounded by*

$$P_e(l,\infty) - e_B \leq 2\exp\Big\{\frac{-l\lambda^2}{2K}\Big\}, \tag{10}$$

where $\lambda \in (0,1]$ depends on the degree of intersection among the components of the mixture distribution and is defined by

$$\lambda = \frac{1}{K}\min_j \Big\{\int_{R_j} f_i(\mathbf{x})d\mathbf{x} - \max_{z\neq i}\int_{R_j} f_z(\mathbf{x})d\mathbf{x}\Big\}. \tag{11}$$

Proof. Let the Voting learning procedure [3] be defined as the function $\Sigma : \mathbb{L} \to S_K$, where $\mathcal{L} \in \mathbb{L}$ is the labelled set and $\hat{\pi}_v \in S_K$ is the permutation of components returned by Voting. In [3], the *excess of risk* of the Voting procedure, $(\varepsilon(\Sigma))$, is defined as

$$E_{\mathbb{L}}\Big[\sum_{j=1}^{K}\int_{R_j}\big(\eta_{\pi_c(j)}f_{\pi_c(j)}(\mathbf{x}) - \eta_{\hat{\pi}_v(j)}f_{\hat{\pi}_v(j)}(\mathbf{x})\big)d\mathbf{x}\Big], \tag{12}$$

where $E_{\mathbb{L}}[\cdot]$ is the expectation with respect to the labelled sample and, $\pi_c(j)$ and $\hat{\pi}_v(j)$ correspond to the true class and the Voting bet of the j-th component, respectively. In that paper, they prove that $\varepsilon(\Sigma) \leq 2\exp\{-l\lambda^2/2K\}$.

Then, we just need to prove that $P_e(l,\infty) - e_B \leq \varepsilon(\Sigma_l)$. By means of the linearity property of both the integral and the expectation over equation (12), we reach

$$E_{\mathbb{L}}\Big[\sum_{j=1}^{K}\int_{R_j}\eta_{\pi_c(j)}f_{\pi_c(j)}(\mathbf{x})d\mathbf{x}\Big] - E_{\mathbb{L}}\Big[\sum_{j=1}^{K}\int_{R_j}\eta_{\hat{\pi}_v(j)}f_{\hat{\pi}_v(j)}(\mathbf{x})d\mathbf{x}\Big].$$

There, the first term is equal to $(1-e_B)$ since $\pi_c(j) = i$ (equation (4)). Then, the second is equal to $\int_{\mathbb{L}}(1-P(e|\mathcal{L}))P(\mathcal{L})d\mathcal{L}$ (where Voting is implicitly contained), which, following the same reasoning as in the proof of theorem 5, it is equal to

$$P_e^V(l,\infty) = \sum_{\pi\in S_K} P(\Sigma_l = \pi)P(e|\pi).$$

Note that, here, $P_e^V(l,\infty)$ is used instead of $P_e(l,\infty)$ to denote that the Voting classifier is used, not the optimal one. Then, by Theorem 2 ($P_e^V(l,\infty) \geq P_e(l,\infty), \forall l$), we can reach the conclusion of equation (10); $P_e(l,\infty)$ is upper bounded by $\varepsilon(\Sigma)$, which, in turn, is upper bounded by $2\exp\{-l\lambda^2/2K\}$. □

1.10 Corollary 2

Corollary 2. *The probability of error, $P_e(l, \infty)$, decreases to e_B, at least, exponentially fast in the number of labelled data.*

Proof. It is trivial to calculate the convergence order from Theorem 7 of the manuscript. □

1.11 Theorem 8

Lemma 2. *Assuming all the priors to be equiprobable and that the model is a Gaussian identifiable mixture of K components with the same variance (σ), let δ_M be the largest distance between the means of two components and let $\Phi(\cdot)$ be the CDF of a standard Gaussian distribution. Then, it holds that*

$$\min_{\pi} P(\Pi_l = \pi) \geq \Big(1 - \Phi\big(\frac{\delta_M}{2\sigma}\sqrt{l}\big)\Big)^{(K!-1)}. \tag{13}$$

Proof. First, $P(\Pi_l = \pi)$ can be rewritten as:

$$\sum_{(l_1,\ldots,l_K)\in\mathcal{G}_K^l} P((l_1, \ldots, l_K)) \times P(\Pi((l_1, \ldots, l_K)) = \pi), \tag{14}$$

which is the decomposition of the probability in terms of the number of labelled examples of each class c_i in l, i.e. $l_i, \forall\ 1 \leq i \leq K$. There, $P((l_1, \ldots, l_K))$ is the probability of having the distribution of labelled samples $(l_1, \ldots, l_K)$ in l, $P(\Pi((l_1, \ldots, l_k)) = \pi)$ is the probability of obtaining the permutation π with a determined distribution of labelled samples. $\mathcal{G}_K^l$ is the set containing all possible distributions of labels defined as the set containing all the integer partitions of l in exactly K addenda, but including zeros and taking into account the order of addenda. Formally, it is defined as

$$\mathcal{G}_K^l \triangleq \{(\gamma_1, \gamma_2, \ldots, \gamma_K) | \sum_{z=1}^{K} \gamma_z = l \wedge \gamma_z \in \{0, 1, \ldots, l\}, \forall z\}.$$

Then, we define $P(\Pi((l_1, \ldots, l_k)) = \pi)$ in terms of the components of the mixture as

$$P\left(\frac{\displaystyle\prod_{i=1}^{K}\prod_{a=1}^{l_i} \eta f_{\pi^{-1}(i)}(x_a)}{\displaystyle\arg\max_{\tau\neq\pi}\{\prod_{i=1}^{K}\prod_{a=1}^{l_i} \eta f_{\tau^{-1}(i)}(x_a)\}} \geq 1\right),$$

where $f_j(x)$ is the density function of the component j. For simplicity of notation, from now on, we rename the numerator as f_π^l and the denominator as $\arg\max_{\tau\neq\pi}\{f_\tau^l\}$.

Then, by defining the permutation π_D as the furthest permutation to π_c, i.e. $\pi_D = \arg\max_\pi \sum_{i=1}^K |\mu_{\pi^{-1}(i)} - \mu_{\pi_c^{-1}(i)}|$, it holds that

$$\min_\pi P\Big(\frac{f_\pi^l}{\arg\max\limits_{\tau\neq\pi}\{f_\tau^l\}} \geq 1\Big) = P\Big(\frac{f_{\pi_D}^l}{\arg\max\limits_{\tau\neq\pi_D}\{f_\tau^l\}} \geq 1\Big). \tag{15}$$

As there is no independency between the permutations, we cannot express the second term of the formula (15) as a product of π_D being greater than or equal to any other permutation τ. For that reason, we use the chain rule to decompose the formula in such a way that the first term of the chain has, $\forall l > 0$, the lowest probability:

$$P\Big(\frac{f_{\pi_D}^l}{\arg\max_{\tau\neq\pi_D}\{f_\tau^l\}} \geq 1\Big) = P\big(f_{\pi_D}^l \geq f_{\pi_c}^l\big) \times \prod_{j=2}^{K!} P\Big(f_{\pi_D}^l \geq f_{\pi_j}^l | f_{\pi_D}^l \geq f_{\pi_c}^l, \bigcap_{m=2}^{j-1} f_{\pi_D}^l \geq f_{\pi_m}^l\Big). \tag{16}$$

Since we assume a mixture of Gaussian components ($\boldsymbol{\theta} = \{\mu_1, \ldots, \mu_K, \sigma\}$), doing some calculations over the first term of the chain in formula (16) we reach the conclusion that it is equal to

$$\begin{aligned} P\big(f_{\pi_D}^l \geq f_{\pi_c}^l\big) \qquad & = 1 - \Phi\left(\frac{1}{2\sigma}\sqrt{\sum_{i=1}^K l_i(\mu_{\pi_D^{-1}(i)} - \mu_{\pi_c^{-1}(i)})^2}\right) \\ \text{and bounded to} \qquad & \\ & \geq 1 - \Phi\left(\frac{1}{2\sigma}\sqrt{\sum_{i=1}^K l_i\delta_M^2}\right) = 1 - \Phi\Big(\frac{\delta_M}{2\sigma}\sqrt{l}\Big). \end{aligned} \tag{17}$$

where $\delta_M = \max\{(\mu_{\pi_D^{-1}(i)} - \mu_{\pi_c^{-1}(i)})^2\}$ is the largest distance between two means. As equation (17) is, by definition of π_D, the lowest term in the product of equation (16), we can lower bound it by substituting the product by equation (17) to the $(K! - 1)$ (number of terms in the chain rule) power. Then, substituting this value in equation (14), we reach formula (13): $\min_\pi P(\Pi_l = \pi)$ is greater than or equal to

$$\Big(1 - \Phi\big(\frac{\delta_M}{2\sigma}\sqrt{l}\big)\Big)^{(K!-1)} \overbrace{\sum_{\mathcal{G}_K^l} P((l_1, \ldots, l_k)}^{1}. \tag{18}$$

□

Theorem 8. ***(Lower bound of the error)*** *Assuming that the model is a Gaussian identifiable mixture of K components with the same variance (σ) and equiprobable priors. Let δ_M be the largest distance between the means of two components and $Q(\cdot)$ a polynomial of degree 3. Then, the probability of error for non-disjoint components is lower bounded by*

$$P_e(l, \infty) - e_B \geq \frac{K!(1 - e_B) - (K-1)!}{\big(1 + \exp\{Q(\frac{\delta_M}{2\sigma}\sqrt{l})\}\big)^{(K!-1)}}. \tag{19}$$

Proof. First, we decompose formula (10) of the main manuscript as follows:

$$\begin{aligned} P_e(l,\infty) = \quad & P(\Pi_l = \pi_c)P(e|\pi_c)+ \\ + \sum_{\pi \in S_K \setminus \pi_c} & P(\Pi_l = \pi)P(e|\pi). \end{aligned} \tag{20}$$

It can be easily seen that, when l increases, whilst $P(\Pi_l = \pi_c)$ grows to 1, the remaining $P(\Pi_l = \pi), \forall \pi \neq \pi_c$ decrease to 0.

Since $P(e|\pi_c)$ is, by definition, equal to e_B and $P(\Pi_l = \pi_c) = 1 - \sum_{\pi \in S_K \setminus \pi_c} P(\Pi_l = \pi)$, by just substituting equation (13) of the previous lemma in (20) and subtracting e_B in both terms, we can lower bound the error $(P_e(l,\infty) - e_B)$ by

$$\Big(1 - \Phi\big(\frac{\delta_M}{2\sigma}\sqrt{l}\big)\Big)^{(K!-1)}(1-e_B) \sum_{\pi \in S_K \setminus \pi_c} P(e|\pi). \tag{21}$$

There, we start by calculating $\sum_{\pi \in S_K \setminus \pi_c} P(e|\pi)$. Note that:

$$\sum_{\pi \in S_K \setminus \pi_c} P(e|\pi) = \sum_{\pi \in S_K} P(e|\pi) - e_B \tag{22}$$

$$\sum_{\pi \in S_K} P(e|\pi) = K! - \sum_{\pi \in S_K} P(\overline{e}|\pi) \tag{23}$$

By substituting formula (9) of Lemma 1 (in the manuscript) in equation (23), we obtain

$$\sum_{\pi \in S_K} P(e|\pi) = K! - \sum_{\pi \in S_K} \sum_{i=1}^{K} \frac{1}{K} \int\limits_{R_{\pi^{-1}(i)}} f_i(\mathbf{x})d\mathbf{x} \tag{24}$$

By distributive property of addition and the fact that $(K-1)!$ permutations of S_K share the same element in the same position, formula (24) can be rewritten as

$$\sum_{\pi \in S_K} P(e|\pi) = K! - (K-1)! \overbrace{\frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{K} \int_{R_j} f_i(\mathbf{x})d\mathbf{x}}^{1},$$

where $j = \pi^{-1}(i)$. Then, we can substitute the result in formula (22) obtaining

$$\sum_{\pi \in S_K \setminus \pi_c} P(e|\pi) = K! - (K-1)! - e_B. \tag{25}$$

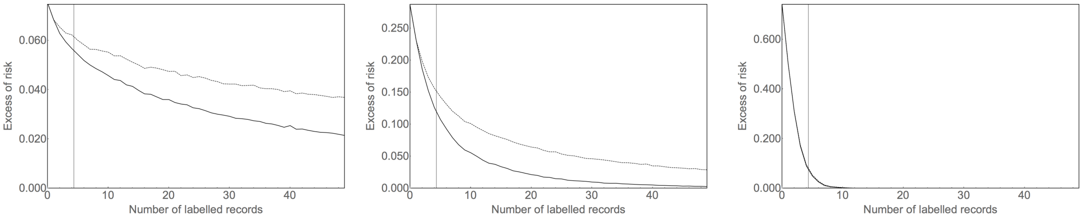
Secondly, as there is no closed form expression for the normal cumulative density function, we approximate $\Phi(x)$ by an inverse exponential as proposed by Page in [4]:

$$\Phi(x) \sim \Phi^{\text{Page}}(x) = 1 - (1 + \exp\{Q(x)\})^{-1}, \tag{26}$$

where $Q(x) = 1.5976x + 0.070565992x^3$ and the absolute error $\epsilon = |\Phi(x) - \Phi^{\text{Page}}(x)| \leq 1.4 \times 10^{-4}, \forall x \geq 0$. Then, by substituting the formulas (26) and (25) in (21) and after some algebra, we reach formula (19). $\square$

2 Appendix B. Supplementary data for the empirical study

The results in this supplementary data for $K = \{4, 5, 6\}$ show a similar style to that presented in the manuscript for $K = 3$. The generative model and the strategies used are the same as in Section VIII (manuscript). However, in these problems, we set $l_{max} = 50$. Whilst Figure 1, Figure 3 and Figure 5 shows the excess of risk of both Voting and PC_{SSL} for the values of $K = \{4, 5, 6\}$, respectively, Figure 2, Figure 4 and Figure 6 present the absolute differences between the probability of error of both learning procedures for the respective values of $K = \{4, 5, 6\}$. In each figure, the subfigure (a) stands for complex problems ($\delta = 0.25$), subfigure (b) for problems showing medium complexity ($\delta = 1$) and subfigure (c) shows the performance for easy problems showing a small degree of intersection among the components ($\delta = 5$).

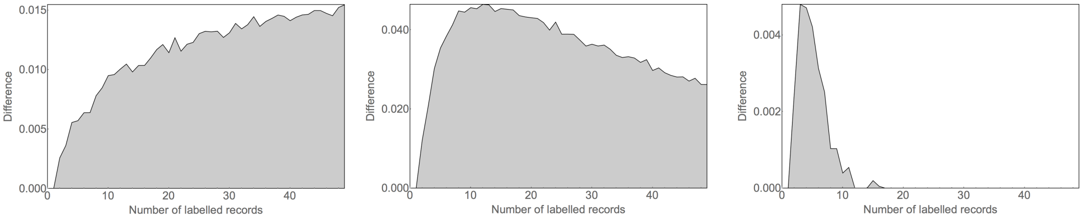


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 1: Probability of error of Voting [3] (upper) and PC_{SSL} (lower curve) for $K = 4$ ($l_K = 4.33$).

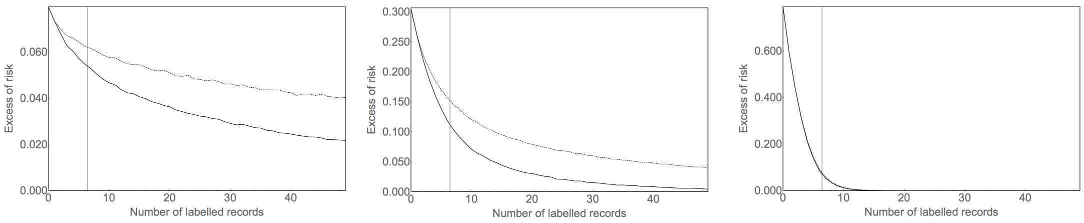


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

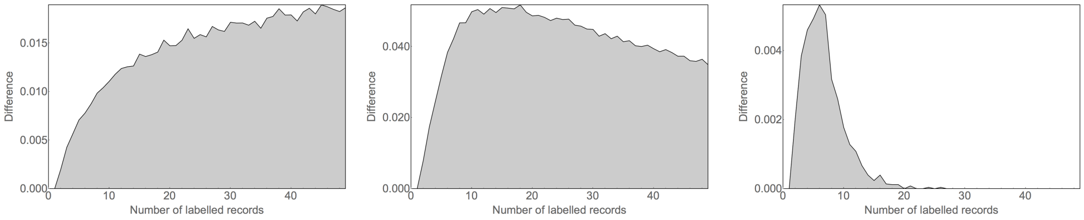
(c) Easy ($\delta = 5$)

Figure 2: Absolute differences between Voting [3] and PC_{SSL}, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 4$.



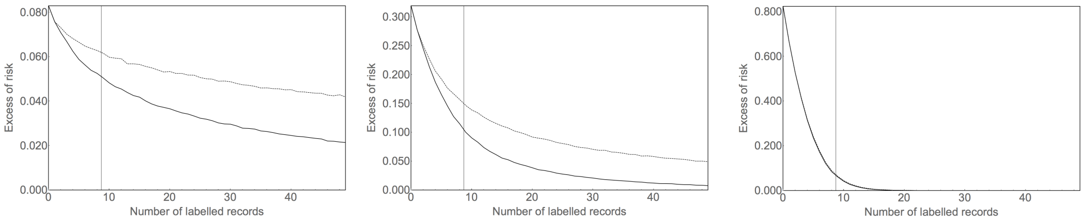
(a) Complex ($\delta = 0.25$). (b) Medium ($\delta = 1$). (c) Easy ($\delta = 5$)

Figure 3: Probability of error of Voting [3] (upper) and PC_{SSL} (lower curve) for $K = 5$ ($l_K = 6.42$).



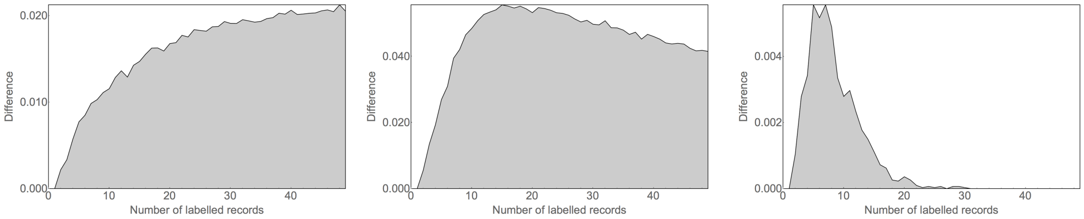
(a) Complex ($\delta = 0.25$). (b) Medium ($\delta = 1$). (c) Easy ($\delta = 5$)

Figure 4: Absolute differences between Voting [3] and PC_{SSL}, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 5$.



(a) Complex ($\delta = 0.25$). (b) Medium ($\delta = 1$). (c) Easy ($\delta = 5$)

Figure 5: Probability of error of Voting [3] (upper) and PC_{SSL} (lower curve) for $K = 6$ ($l_K = 8.7$).



(a) Complex ($\delta = 0.25$). (b) Medium ($\delta = 1$). (c) Easy ($\delta = 5$)

Figure 6: Absolute differences between Voting [3] and $\mathrm{PC}_{\mathrm{SSL}}$, i.e. $P_e^V(l,\infty) - P_e^P(l,\infty)$, for $K = 6$.

3 Appendix C. Source code

The Mathematica package [5] available to download from our website contains the formulae presented in the manuscript. Besides, it also contains some experiments used to study the behaviour of the probability of error of both Voting [3] and PC$_{\mathrm{SSL}}$ learning algorithms.

3.1 l_k-related functions

1) The function **`LKEQ[K]`** calculates the l_K value for a problem with **`K`** equiprobable clases.

——**`LKEQ[K]`** ——

Input parameters:

`K` Number of classes.

Output:

A real number, l_k.

---

2) The function **`LKEQPlot[maxK]`** prints a figure of the l_K values for problems of $\{1..$**`maxK`**$\}$ equiprobable classes. The Figure 2a of the manuscript has been created with this function.

——**`LKEQPlot[maxK]`** ——

Input parameters:

`maxK` Number of classes.

Output:

Plot in the standard output.

---

3) **`LKPriors[priors, nRep]`** use a Monte Carlo method with **`nRep`** repetitions to approximate l_k for a problem with **`Length[priors]`** non-equiprobable priors given by the variable **`priors`**.

——**`LKPriors[priors, nRep]`** ——

Input parameters:

`priors` List containing K class priors.
`nRep` Number of repetitions.

Output:

A real number, l_k.

---

4) **`LKSampling[K, samplesize, nRep]`** applies **`LKPriors[priors, nRep]`** with **`nRep`** repetitions over a population of class priors of size **`samplesize`** which are generated from a Dirichlet distribution with all the alpha hyper-parameters equal to 1.

——`LKSampling[K, samplesize, nRep]` ————

Input parameters:

`K`	Number of classes.
`samplesize`	Population size.
`nRep`	Number of repetitions for `LKPriors[...]`.

Output:

A list of l_k, one per each sample of the population.

---

5) The eq. (8) of Theorem 4 corresponds to `L3[n1,n2,n3]`. It calculates l_3 for a ternary problem with priors `n1`, `n2` and `n3`.

——`L3[n1,n2,n3]` ————

Input parameters:

`n1`	Class prior of the class c_1
`n2`	Class prior of the class c_2
`n3`	Class prior of the class c_3

Output:

A real number, l_3.

---

6) `L3Plot[]` plots `L3[n1,n2,n3]` assuming that `n1`$= \eta_1$ and `n2, n3`$= (1 - \eta_1)/(K - 1)$.

——`L3Plot[]` ————

Input parameters:

−−

Output:

Plot in the standard output.

3.2 Functions related to the probability of error

7) The function `ZeroEB[K,maxL]` calculates the probability of error for $l = 0..$`maxL` for a `K`-class problem when there is no intersection among the components.

——`ZeroEB[K,maxL]` ————

Input parameters:

`K`	Number of classes.
`maxL`	Maximum number of labelled examples.

Output:

Summary of results in the standard output.

---

8) **`MCPCSSL[K,distance,sigma,maxL,nRep]`** uses a Monte Carlo method with **`nRep`** repetitions to approximate $P_e(l,\infty)$ for $l = \{0..\mathtt{maxL}\}$ assuming a mixture of `K` Gaussian. **`distance`** represents the distance between the adjacent means and **`sigma`** is the variance.

`MCPCSSL[K,distance,sigma,maxL,nRep]`

Input parameters:		
	`K`	Number of classes.
	`distance`	Distance between the means of adjacent components.
	`sigma`	Variance of the Gaussian components.
	`maxL`	Maximum number of labelled examples.
	`nRep`	Number of repetitions for the Monte Carlo method.
Output:		
		Summary of results in the standard output.

9) The function **`MCPCSSLBiased[K,distance,sigma,maxL,bias, nRep]`** uses a Monte Carlo method with **`nRep`** repetitions to approximate $P_e(l,\infty)$ for $l = \{0..\mathtt{maxL}\}$ assuming a a generative model composed of a mixture of `K` Gaussian. **`distance`** represents the distance between the adjacent means and **`sigma`** is the variance. In this simulation the learnt model is also a mixture of Gaussian components, but they are shifted a factor **`bias`** to the right, i.e. $\hat{\mu}_i - \mu_i = \mathtt{bias}$. This function corresponds to the second experiment of the manuscript.

`MCPCSSLBiased[K,distance,sigma,maxL,bias, nRep]`

Input parameters:		
	`K`	Number of classes.
	`distance`	Distance between the means of adjacent components.
	`sigma`	Variance of the Gaussian components.
	`maxL`	Maximum number of labelled examples.
	`bias`	Bias between the learnt and the generative models.
	`nRep`	Number of repetitions for the Monte Carlo method.
Output:		
		Summary of results in the standard output.

10) **`MCVOTING[K,distance,sigma,maxL,bias, nRep]`** uses a Monte Carlo method with **`nRep`** repetitions to approximate the probability of error of Voting for $l = \{0..\mathtt{maxL}\}$ assuming a mixture of `K` Gaussian. **`distance`** represents the distance between the adjacent means and **`sigma`** is the variance.

——**`MCVOTING[K,distance,sigma,maxL,bias, nRep]`**——

Input parameters:

`K`	Number of classes.
`distance`	Distance between the means of adjacent components.
`sigma`	Variance of the Gaussian components.
`maxL`	Maximum number of labelled examples.
`nRep`	Number of repetitions for the Monte Carlo method.

Output:

Summary of results in the standard output.

---

11) **`MCComparison[K,distance,sigma,maxL,bias, nRep]`** uses a Monte Carlo method with **`nRep`** repetitions to approximate the probability of error of both $\mathrm{PC}_{\mathrm{SSL}}$ and Voting for $l = \{0..\texttt{maxL}\}$ assuming a mixture of `K` Gaussian. **`distance`** represents the distance between the adjacent means and **`sigma`** is the variance.

——**`MCComparison[K,distance,sigma,maxL,bias, nRep]`**——

Input parameters:

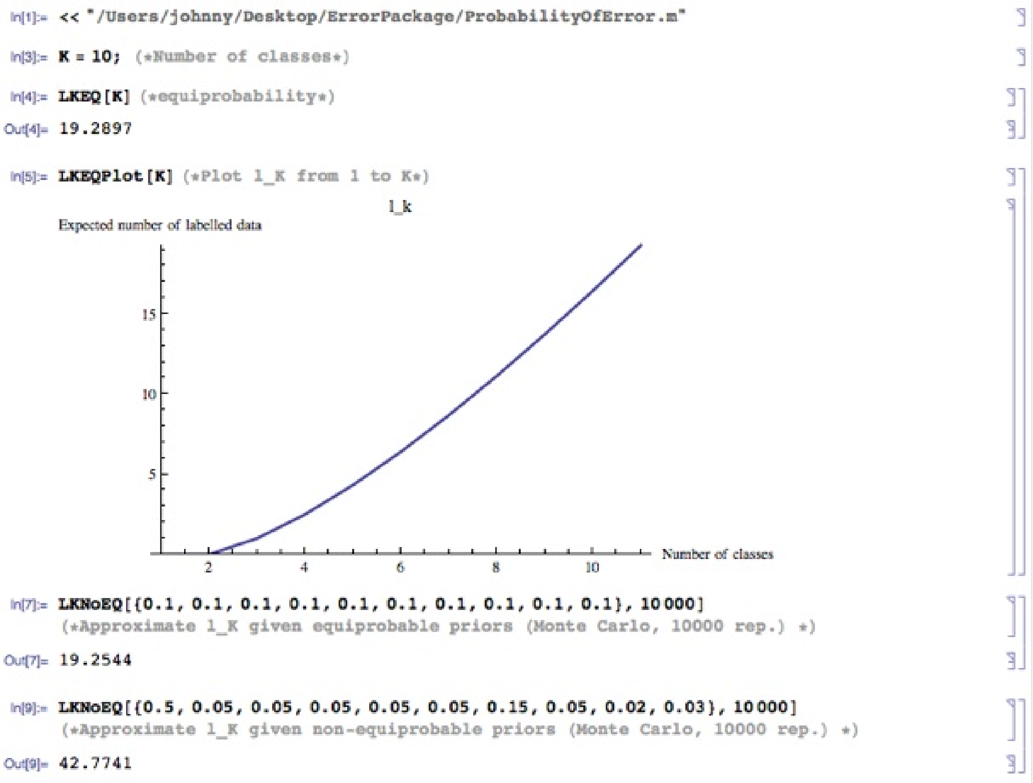
`K`	Number of classes.
`distance`	Distance between the means of adjacent components.
`sigma`	Variance of the Gaussian components.
`maxL`	Maximum number of labelled examples.
`nRep`	Number of repetitions for the Monte Carlo method.

Output:

Summary of results in the standard output.

3.3 Examples of the executions of the package

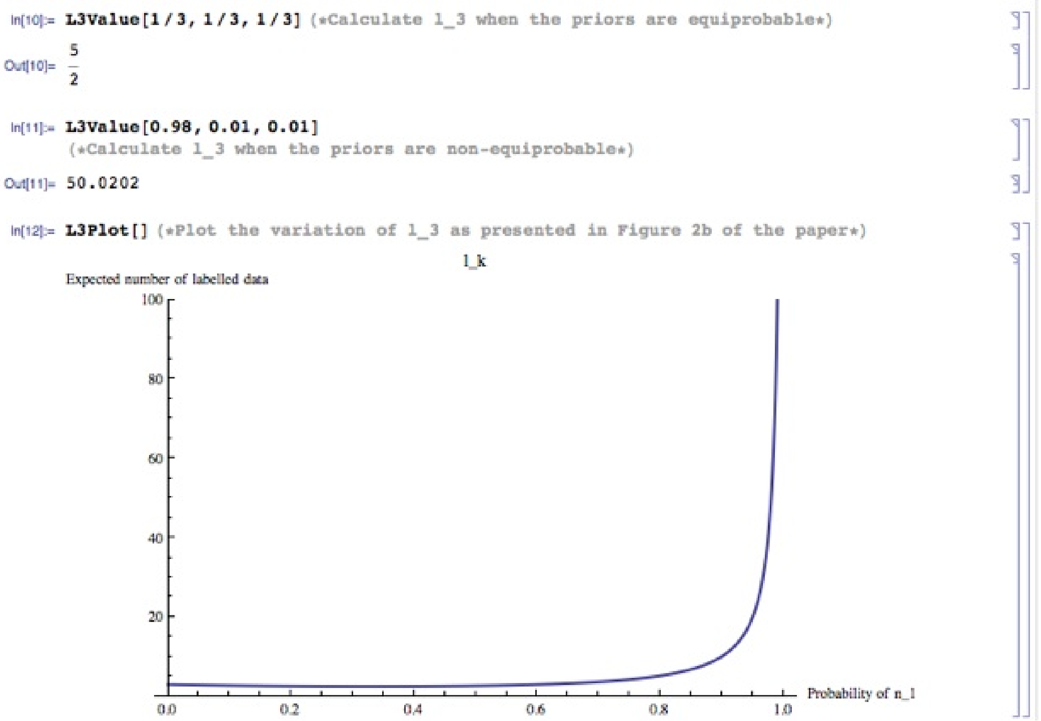
Figure 7 and Figure 8 shows two executions of the package to calculate l_K and l_3.



```
In[1]:= << "/Users/johnny/Desktop/ErrorPackage/ProbabilityOfError.m"

In[3]:= K = 10; (*Number of classes*)

In[4]:= LKEQ[K] (*equiprobability*)

Out[4]= 19.2897

In[5]:= LKEQPlot[K] (*Plot l_K from 1 to K*)
```

```
In[7]:= LKNoEQ[{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1}, 10000]
        (*Approximate l_K given equiprobable priors (Monte Carlo, 10000 rep.) *)

Out[7]= 19.2544

In[9]:= LKNoEQ[{0.5, 0.05, 0.05, 0.05, 0.05, 0.05, 0.15, 0.05, 0.02, 0.03}, 10000]
        (*Approximate l_K given non-equiprobable priors (Monte Carlo, 10000 rep.) *)

Out[9]= 42.7741
```

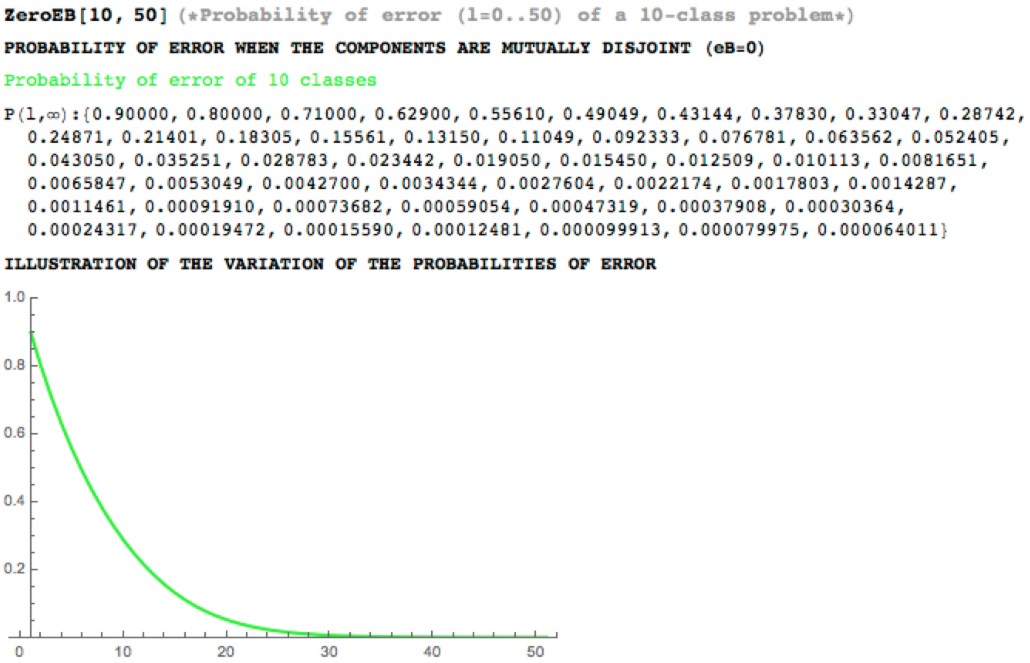
Figure 7: An example of the use of the l_k-related functions.



```
In[10]:= L3Value[1/3, 1/3, 1/3] (*Calculate l_3 when the priors are equiprobable*)

         5
Out[10]= -
         2

In[11]:= L3Value[0.98, 0.01, 0.01]
         (*Calculate l_3 when the priors are non-equiprobable*)

Out[11]= 50.0202

In[12]:= L3Plot[] (*Plot the variation of l_3 as presented in Figure 2b of the paper*)
```

Figure 8: An example of the calculation of l_3.

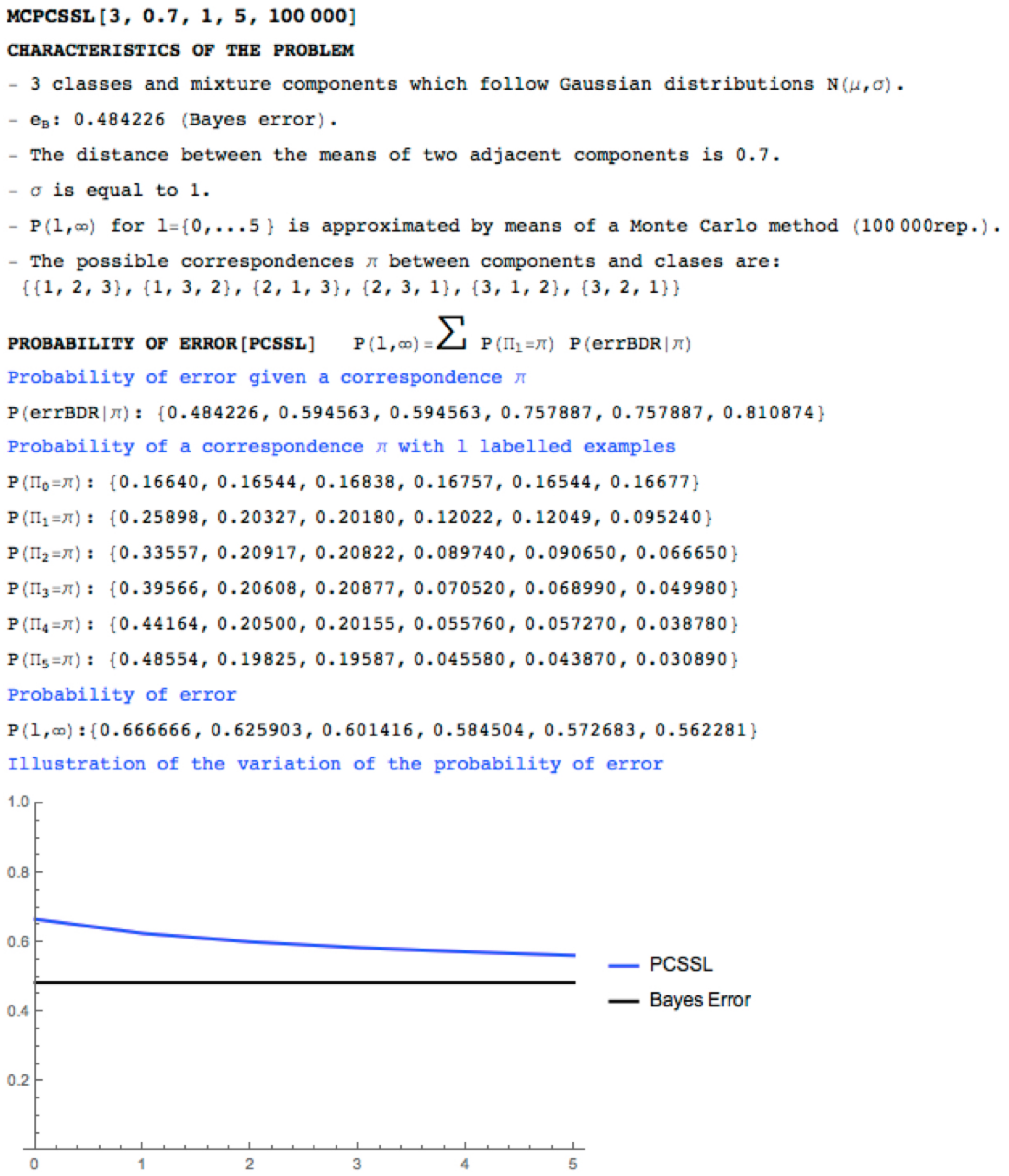
Figure 9 shows an example of how, in cases of mutually non-intersecting components, the probability of error is calculated.



```
ZeroEB[10, 50] (*Probability of error (l=0..50) of a 10-class problem*)
PROBABILITY OF ERROR WHEN THE COMPONENTS ARE MUTUALLY DISJOINT (eB=0)
Probability of error of 10 classes
P(l,∞):{0.90000, 0.80000, 0.71000, 0.62900, 0.55610, 0.49049, 0.43144, 0.37830, 0.33047, 0.28742,
  0.24871, 0.21401, 0.18305, 0.15561, 0.13150, 0.11049, 0.092333, 0.076781, 0.063562, 0.052405,
  0.043050, 0.035251, 0.028783, 0.023442, 0.019050, 0.015450, 0.012509, 0.010113, 0.0081651,
  0.0065847, 0.0053049, 0.0042700, 0.0034344, 0.0027604, 0.0022174, 0.0017803, 0.0014287,
  0.0011461, 0.00091910, 0.00073682, 0.00059054, 0.00047319, 0.00037908, 0.00030364,
  0.00024317, 0.00019472, 0.00015590, 0.00012481, 0.000099913, 0.000079975, 0.000064011}
ILLUSTRATION OF THE VARIATION OF THE PROBABILITIES OF ERROR
```

Figure 9: An execution of `ZeroEB[K,maxL]`.

Whilst Figure 10 shows how the probability of error of PC_{SSL} is calculated with the Mathematica package, Figure 8 provides an example of the calculation of the probability of error of PC_{SSL} when the correct model assumption is not met.



MCPCSSL[3, 0.7, 1, 5, 100 000]

CHARACTERISTICS OF THE PROBLEM

- 3 classes and mixture components which follow Gaussian distributions N(μ,σ).
- e_B: 0.484226 (Bayes error).
- The distance between the means of two adjacent components is 0.7.
- σ is equal to 1.
- P(l,∞) for l={0,...5 } is approximated by means of a Monte Carlo method (100 000rep.).
- The possible correspondences π between components and clases are:
{{1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}}

PROBABILITY OF ERROR[PCSSL] $P(l,\infty)=\sum P(\Pi_l=\pi)\ P(\mathrm{errBDR}|\pi)$

Probability of error given a correspondence π

P(errBDR|π): {0.484226, 0.594563, 0.594563, 0.757887, 0.757887, 0.810874}

Probability of a correspondence π with l labelled examples

P(Π_0=π): {0.16640, 0.16544, 0.16838, 0.16757, 0.16544, 0.16677}

P(Π_1=π): {0.25898, 0.20327, 0.20180, 0.12022, 0.12049, 0.095240}

P(Π_2=π): {0.33557, 0.20917, 0.20822, 0.089740, 0.090650, 0.066650}

P(Π_3=π): {0.39566, 0.20608, 0.20877, 0.070520, 0.068990, 0.049980}

P(Π_4=π): {0.44164, 0.20500, 0.20155, 0.055760, 0.057270, 0.038780}

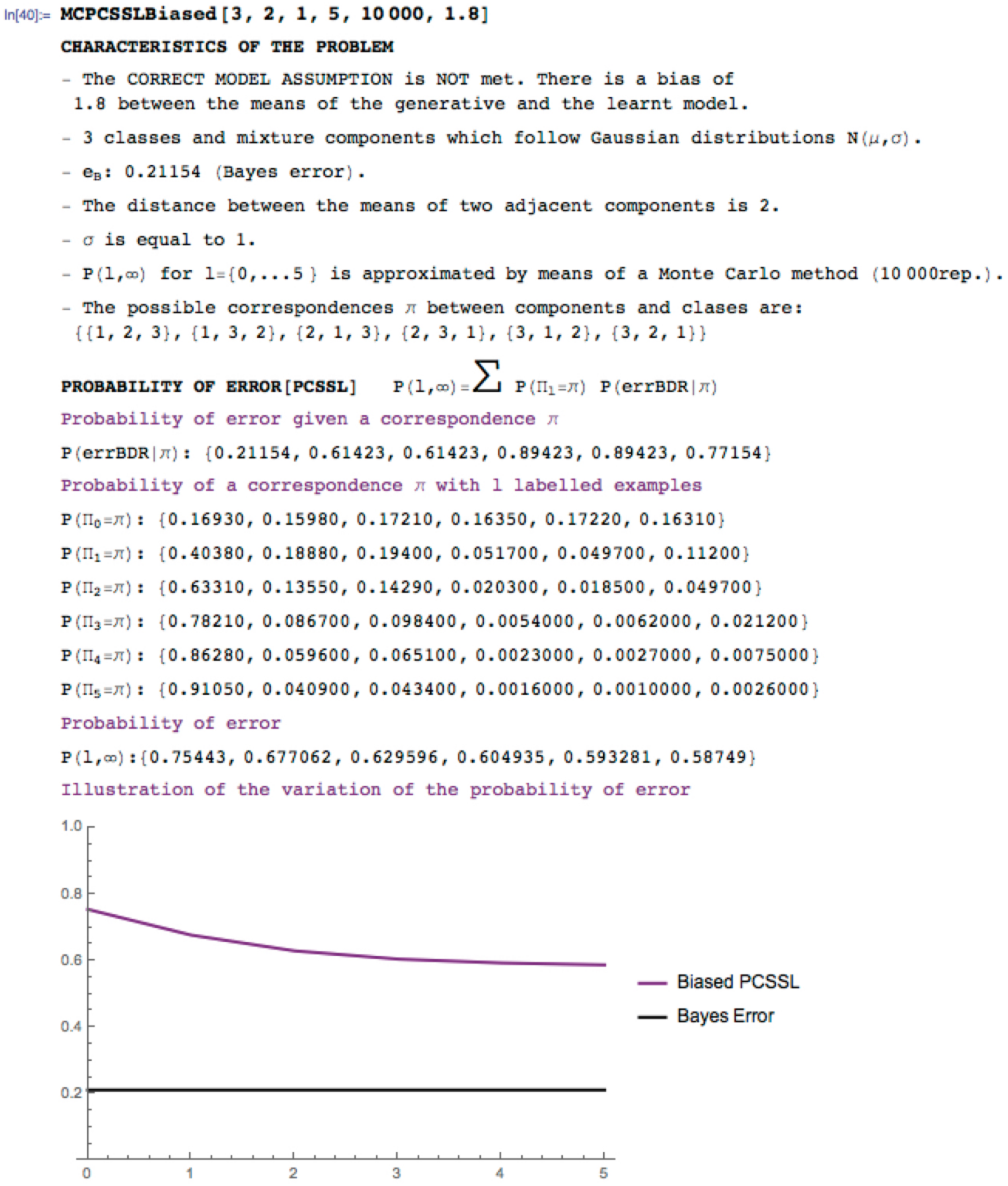
P(Π_5=π): {0.48554, 0.19825, 0.19587, 0.045580, 0.043870, 0.030890}

Probability of error

P(l,∞):{0.666666, 0.625903, 0.601416, 0.584504, 0.572683, 0.562281}

Illustration of the variation of the probability of error

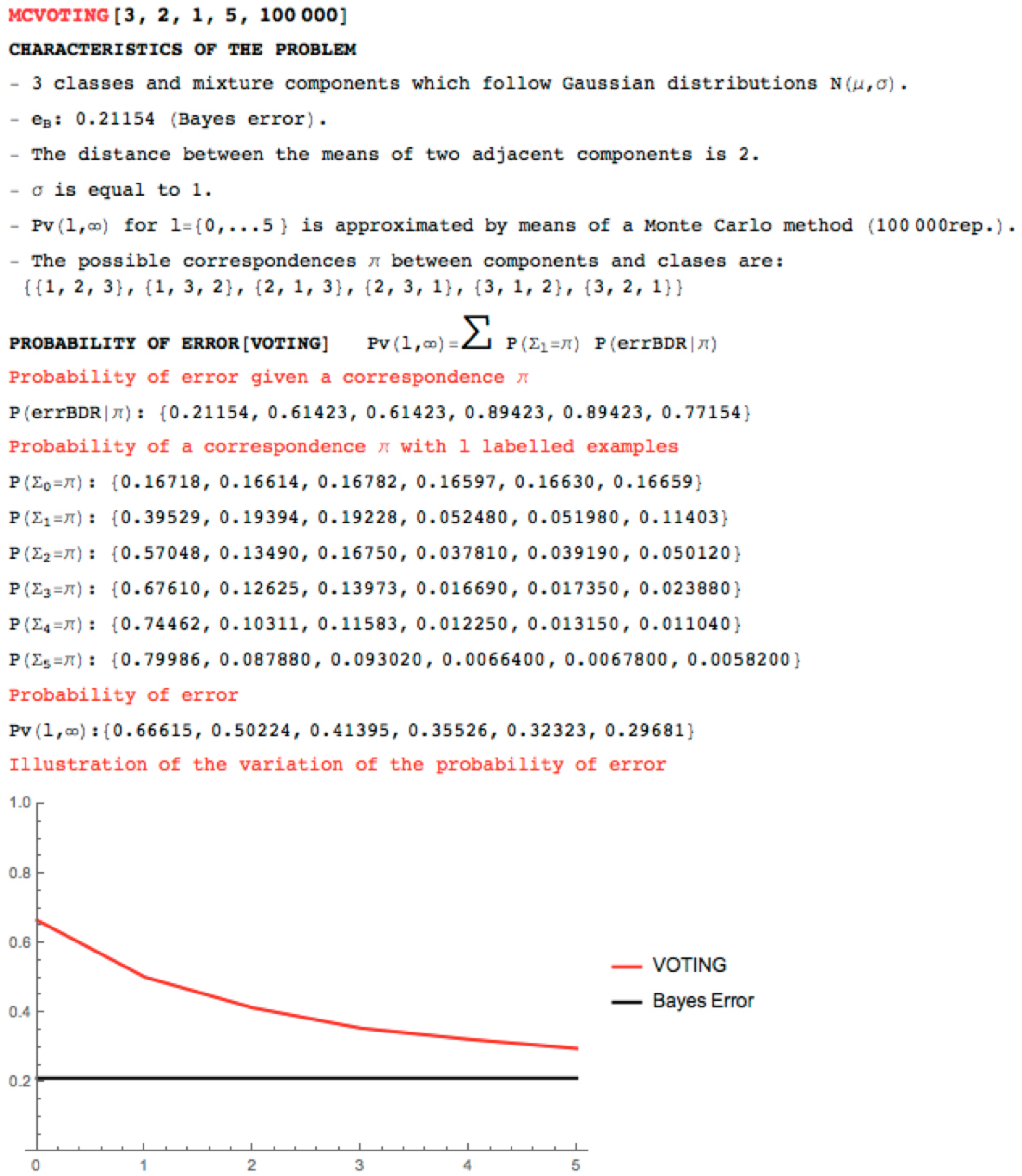
Figure 10: Calculating the optimal probability of error using PC_{SSL}.



```
In[40]:= MCPCSSLBiased[3, 2, 1, 5, 10 000, 1.8]
         CHARACTERISTICS OF THE PROBLEM
         - The CORRECT MODEL ASSUMPTION is NOT met. There is a bias of
          1.8 between the means of the generative and the learnt model.
         - 3 classes and mixture components which follow Gaussian distributions N(μ,σ).
         - eB: 0.21154 (Bayes error).
         - The distance between the means of two adjacent components is 2.
         - σ is equal to 1.
         - P(l,∞) for l={0,...5 } is approximated by means of a Monte Carlo method (10 000rep.).
         - The possible correspondences π between components and clases are:
          {{1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}}

         PROBABILITY OF ERROR[PCSSL]    P(l,∞)=∑ P(Π1=π) P(errBDR|π)
         Probability of error given a correspondence π
         P(errBDR|π): {0.21154, 0.61423, 0.61423, 0.89423, 0.89423, 0.77154}
         Probability of a correspondence π with l labelled examples
         P(Π0=π): {0.16930, 0.15980, 0.17210, 0.16350, 0.17220, 0.16310}
         P(Π1=π): {0.40380, 0.18880, 0.19400, 0.051700, 0.049700, 0.11200}
         P(Π2=π): {0.63310, 0.13550, 0.14290, 0.020300, 0.018500, 0.049700}
         P(Π3=π): {0.78210, 0.086700, 0.098400, 0.0054000, 0.0062000, 0.021200}
         P(Π4=π): {0.86280, 0.059600, 0.065100, 0.0023000, 0.0027000, 0.0075000}
         P(Π5=π): {0.91050, 0.040900, 0.043400, 0.0016000, 0.0010000, 0.0026000}
         Probability of error
         P(l,∞):{0.75443, 0.677062, 0.629596, 0.604935, 0.593281, 0.58749}
         Illustration of the variation of the probability of error
```

Figure 11: An execution of PC_{SSL} when the correct model assumption is not met.

Then, Figure 12 shows how the probability of error of PC_{SSL} is calculated.



MCVOTING[3, 2, 1, 5, 100 000]

CHARACTERISTICS OF THE PROBLEM

- 3 classes and mixture components which follow Gaussian distributions N(μ,σ).
- e_B: 0.21154 (Bayes error).
- The distance between the means of two adjacent components is 2.
- σ is equal to 1.
- Pv(l,∞) for l={0,...5 } is approximated by means of a Monte Carlo method (100 000rep.).
- The possible correspondences π between components and clases are:
  {{1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}}

PROBABILITY OF ERROR[VOTING] $\mathrm{Pv}(l,\infty)=\sum \mathrm{P}(\Sigma_l=\pi)\ \mathrm{P}(\mathrm{errBDR}|\pi)$

Probability of error given a correspondence π

P(errBDR|π): {0.21154, 0.61423, 0.61423, 0.89423, 0.89423, 0.77154}

Probability of a correspondence π with l labelled examples

P(Σ_0=π): {0.16718, 0.16614, 0.16782, 0.16597, 0.16630, 0.16659}

P(Σ_1=π): {0.39529, 0.19394, 0.19228, 0.052480, 0.051980, 0.11403}

P(Σ_2=π): {0.57048, 0.13490, 0.16750, 0.037810, 0.039190, 0.050120}

P(Σ_3=π): {0.67610, 0.12625, 0.13973, 0.016690, 0.017350, 0.023880}

P(Σ_4=π): {0.74462, 0.10311, 0.11583, 0.012250, 0.013150, 0.011040}

P(Σ_5=π): {0.79986, 0.087880, 0.093020, 0.0066400, 0.0067800, 0.0058200}

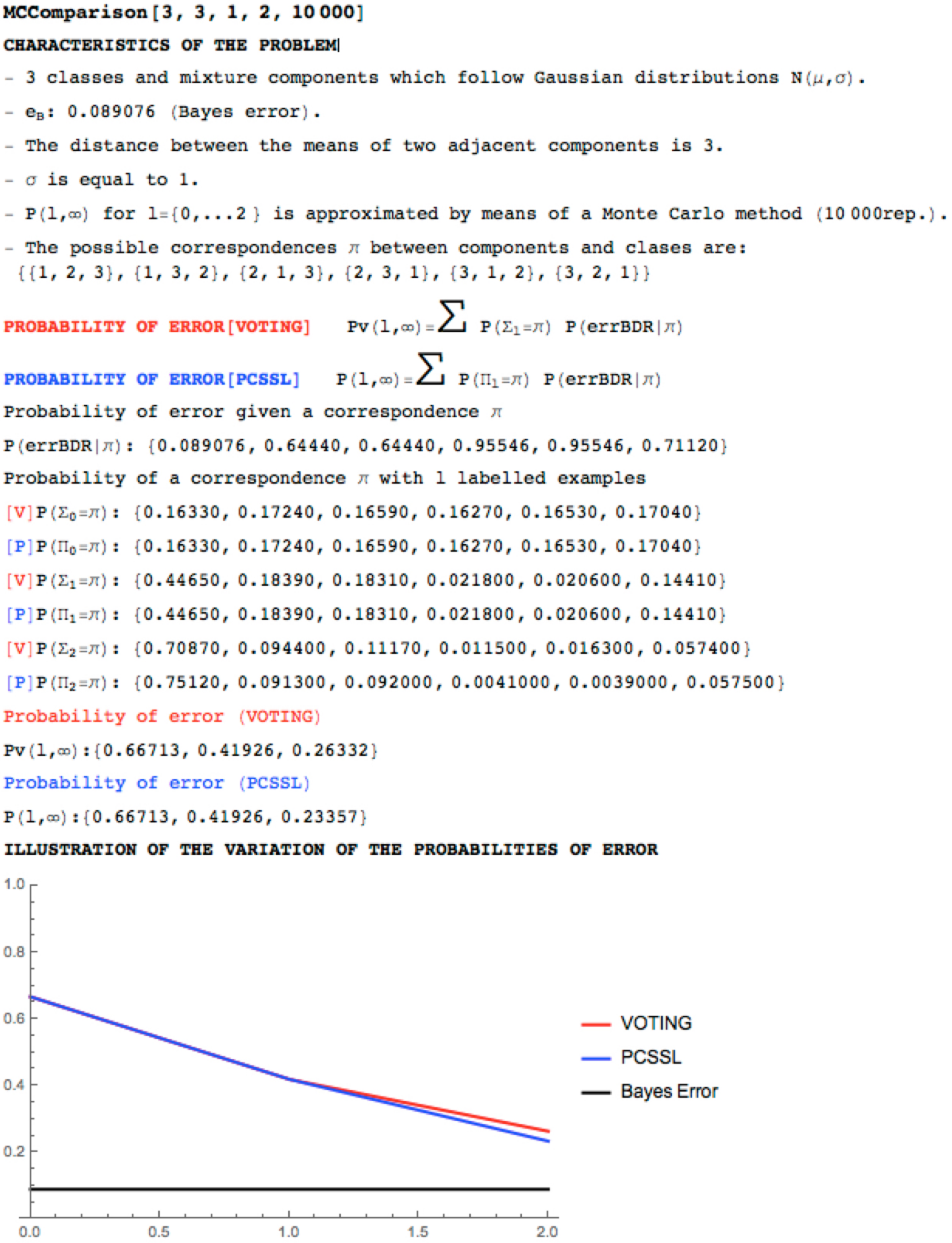
Probability of error

Pv(l,∞):{0.66615, 0.50224, 0.41395, 0.35526, 0.32323, 0.29681}

Illustration of the variation of the probability of error

Figure 12: An example of the execution of the Voting strategy.

Finally, 13 shows an example of how the comparison between both procedures can be made with the function `MCComparison`.



MCComparison[3, 3, 1, 2, 10 000]

CHARACTERISTICS OF THE PROBLEM

- 3 classes and mixture components which follow Gaussian distributions N(μ,σ).

- e_B: 0.089076 (Bayes error).

- The distance between the means of two adjacent components is 3.

- σ is equal to 1.

- P(l,∞) for l={0,...2 } is approximated by means of a Monte Carlo method (10 000rep.).

- The possible correspondences π between components and clases are:
{{1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}}

PROBABILITY OF ERROR[VOTING] $\mathrm{Pv}(l,\infty) = \sum P(\Sigma_l=\pi)\ P(\mathrm{errBDR}|\pi)$

PROBABILITY OF ERROR[PCSSL] $P(l,\infty) = \sum P(\Pi_l=\pi)\ P(\mathrm{errBDR}|\pi)$

Probability of error given a correspondence π

P(errBDR|π): {0.089076, 0.64440, 0.64440, 0.95546, 0.95546, 0.71120}

Probability of a correspondence π with l labelled examples

[V]P(Σ_0=π): {0.16330, 0.17240, 0.16590, 0.16270, 0.16530, 0.17040}

[P]P(Π_0=π): {0.16330, 0.17240, 0.16590, 0.16270, 0.16530, 0.17040}

[V]P(Σ_1=π): {0.44650, 0.18390, 0.18310, 0.021800, 0.020600, 0.14410}

[P]P(Π_1=π): {0.44650, 0.18390, 0.18310, 0.021800, 0.020600, 0.14410}

[V]P(Σ_2=π): {0.70870, 0.094400, 0.11170, 0.011500, 0.016300, 0.057400}

[P]P(Π_2=π): {0.75120, 0.091300, 0.092000, 0.0041000, 0.0039000, 0.057500}

Probability of error (VOTING)

Pv(l,∞):{0.66713, 0.41926, 0.26332}

Probability of error (PCSSL)

P(l,∞):{0.66713, 0.41926, 0.23357}

ILLUSTRATION OF THE VARIATION OF THE PROBABILITIES OF ERROR

Figure 13: Comparing both learning algorithms using the function **MCComparison[...]**.

References

[1] V. Castelli and T. Cover, "On the exponential value of labeled samples," *Pattern Recognition Letters*, vol. 16, pp. 105–111, 1995.

[2] R. P. Stanley, *Enumerative combinatorics.* Wadsworth Publ. Co., 1986.

[3] H. Chen and L. Li, "Semisupervised multicategory classification with imperfect model," *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1594–1603, 2009.

[4] E. Page, "Approximation to the cummulative normal function and its inverse for use on a pocket calculator," *Applied Statistics*, vol. 26, pp. 75–76, 1977.

[5] Wolfram Research Inc., "Mathematica," 2014.

6

Towards Competitive Classifiers for Unbalanced Classification Problems: A Study on the Performance Scores

> *When we had toiled for months over problems of integrating, Einstein used to remark:"God does not care about our mathematical difficulties; He integrates empirically".*
>
> - Leopold Infeld, *Quest: An Autobiography*

Submitted to Machine Learning
Manuscript Reference MACH-D-17-00062

Towards Competitive Classifiers for Unbalanced Multi-class Classification Problems

A Study on the Performance Scores

Jonathan Ortigosa-Hernández ·
Iñaki Inza · **Jose A. Lozano**

Abstract Although a great methodological effort has been invested in proposing competitive solutions to the class-imbalance problem, little effort has been made in pursuing a theoretical understanding of this matter. In order to shed some light on this topic, we perform, through a novel framework, an exhaustive analysis of the adequateness of the most commonly used performance scores to assess this complex scenario in binary problems for the multi-class framework. We conclude that using unweighted Hölder means with exponent $p \leq 1$ to average the recalls of all the classes produces adequate scores which are capable of determining whether a classifier is competitive. Next, we review the dominant solutions (data-sampling and cost-sensitive learning) presented in the multi-class class-imbalance literature. Since any learning task can be defined as an optimisation problem where a loss function, usually connected to a particular score, is minimised, our goal, here, is to find whether the learning tasks found in the literature are also oriented to maximise the previously detected adequate scores. We conclude that they usually maximise the unweighted Hölder mean with $p = 1$ (a-mean). Finally, we provide bounds on the values of the studied performance scores which guarantee a multi-class classifier with a higher recall than the random classifier in each and every class.

Keywords Supervised Classification · Class-Imbalance Problem · Bayes Decision Rule · Multi-class Problems · Performance Assessment · Hölder Means.

Jonathan Ortigosa-Hernández, Iñaki Inza and Jose A. Lozano
Intelligent Systems Group at the Department of Computer Science and Artificial Intelligence, Computer Science Faculty, The University of the Basque Country UPV/EHU, Donostia-San Sebastián, Spain.
E-mail: jonathan.ortigosa@ehu.es, inaki.inza@ehu.es, ja.lozano@ehu.es

Jonathan Ortigosa-Hernández
Industry 4.0 Department, Gestamp Automoción SL, Madrid, Spain.

Jose A. Lozano
Basque Center for Applied Mathematics BCAM, Bilbao, Spain.

1 Introduction

In many classification problems, there are significant differences among the probabilities of the classes, i.e. the probability of a particular example belonging to a certain class. In the literature, this situation is known as the class-imbalance problem (Kubat et al., 1997; Provost et al., 1998; Axelsson, 2000; He and Garcia, 2009; Weiss, 2013; Branco et al., 2016) and it is considered to be a major obstacle to building competitive classifiers. By *competitive classifiers* we refer to classifiers showing not only a low overall classification error, but also a balance between the prediction powers for both the most and the least probable classes.

Moreover, the investigated class-imbalance problem has somewhat torn apart the conventional approaches to solve classification problems. In traditional supervised learning approaches (Wu et al., 2007), classification accuracy is by far the most popular numerical performance score due to its theoretical foundations, its simplicity and its property of being intuitive (Provost et al., 1998). On one hand, it has been used to guide learning processes due to the fact that most learning algorithms are designed to asymptotically converge to the behaviour of the Bayes decision rule (Young and Calvert, 1974), a classifier which is optimal for this score. On the other hand, this performance score has also been broadly used in the literature to evaluate the performance of real-world classifiers (Provost et al., 1998). However, in recent years, the research community has noticed that it is not an adequate score for problems with extremely skewed class distributions; the least probable classes have very little impact on the classification accuracy when compared to the most probable classes (Gu et al., 2009). When dealing with highly unbalanced problems, this implies that (i) traditional learning approaches maximising the classification accuracy may produce dummy classifiers which always predict the most probable classes (Drummond and Holte, 2005), and that (ii) this performance score is no longer convenient for assessing real-world classifiers since a high value does not guarantee a fair prediction power for the underrepresented classes.

In view of this puzzling situation and mainly in binary classification, there has been significant methodological effort invested not only in determining which performance scores are the most appropriate to the class-imbalance scenario (Gu et al., 2009), but also in proposing new learning systems to obtain competitive classifiers for highly unbalanced classification problems (Fernández et al., 2011; Wang and Yao, 2012; López et al., 2013). However, little theoretical effort (Weiss, 2004; Drummond and Holte, 2005) has been made in pursuing a complete understanding of these topics in binary problems, and much less in the multi-class scenario. Most of the successful proposals are the result of experience, systematic studies (Japkowicz and Stephen, 2002) or just pure intuition on how special prominence can be given to the least probable classes (Wang and Yao, 2012), rather than being built on a solid theoretical foundation (He and Garcia, 2009). Table 2, in Appendix C, summarises the closest studies which can be found in the literature about the class-imbalance problems and their main differences to this manuscript. To the best of our knowledge, the following questions are still unanswered in the literature:

1. Which performance scores are adequate to determine the competitiveness of a classifier in multi-class unbalanced domains?

2. Which performance scores (loss functions) are maximised (minimised) in the most well-known learning solutions designed to deal with skewed classes?
3. Can bounds guaranteeing the competitiveness of a classifier be provided for certain adequate performance scores?

Thus, in order to shed some theoretical light on the previous three questions, in this paper, we perform the following three studies:

A first study on determining the adequate performance scores for class-imbalance scenarios in the multi-class setting. Although this issue has been long studied in the binary setting (Menon et al., 2013), to the best of our knowledge, there is still no convention on which performance scores should be used to assess the performance of the classifiers in multi-class unbalanced scenarios. Thus, firstly, our main objective is to analyse how different performance scores behave under a changing multi-class distribution when the Bayes decision rule is used as a classifier. The usage of this rule is due to the fact that, as it has been long studied in the literature e.g. (Drummond and Holte, 2005), there is a complete understanding of the deficiencies of this classifier in this complex scenario. Hence, this knowledge can be exploited in order to obtain a licit characterisation of the adequateness of the performance scores; by having a look at the fluctuation of the values of the performance scores when the class distribution varies, we can devise which scores are the most appropriate to evaluate the classifiers for unbalanced domains. However, having differences among the probabilities of the classes is not the only factor hindering the proposed solutions (López et al., 2013); other factors such as class-overlapping or the presence of noise data also modify the values of the performance scores hampering the study of the contribution of the class distribution to the performance of a learnt classifier. Thus, based on these grounds, we define a novel controlled framework where these other hindering factors can be properly cancelled so that the contribution of the class distribution to the performance scores can be legitimately quantified in isolation. Conclusions show that the performance scores which are unweighted Hölder means (Bullen, 2003) with $p \leq 1$ among the recalls (Section 2) are the most appropriate to evaluate the competitiveness of classifiers in unbalanced multi-class problems. In this regard, misclassifying the least probable classes is highly penalised.

A second study on discovering the performance scores maximised in the most predominant solutions of the class-imbalance literature. Since it is known that the typical supervised learning techniques are unable to deal with the class-imbalance problem (Drummond and Holte, 2005) and given the large number of publications reporting positive results (Batista et al., 2004; Liu and Zhou, 2006), we analyse the most common approaches to the class-imbalance problem, which are easily extensible to the multi-class setting, by assuming that different performance scores than classification accuracy are maximised in those solutions. Our main conclusion after analysing data sampling and cost sensitive learning techniques is that they output classifiers which are maximised for the unweighted Hölder mean among the recalls with $p = 1$ (a-mean) and that they have an asymptotic behaviour close to the Bayes decision rule for an equiprobable class distribution (an optimal rule for that performance score). Moreover, we also show that, in scenarios showing skewed class distributions, this rule outperforms the well-known Bayes decision rule in terms of the values of the adequate scores detected in the first study. Yet, the usage of any other unweighted Hölder mean with $p < 1$ to define new learning procedures for dealing with unbalance problems is also appropriate.

A third study on proposing bounds for the performance scores to determine the competitiveness of multi-class classifiers in unbalanced domains. For this purpose, we delimit the definition of competitiveness of a given solution to a classifier having higher recalls than the random classifier in each and every class. Thus, we can provide two different practical bounds for the values of the performance scores expressed as unweighted Hölder means among the recalls with $p \in \mathbb{R} \cup \{+\infty, -\infty\}$; a bound for the lowest value of the performance score ensuring a competitive solution, and a bound for the highest value of the score indicating an incompetent solution, i.e. the random classifier obtains a better recall in at least one class value. Here, our conclusions are also consistent with the first study; since the distance between both bounds decreases along with p (rapidly for $p \leq 1$), using Hölder means with $p \leq 1$ is presumed to be adequate for determining the competitiveness of a classifier. In fact, both bounds coincide in $p = -\infty$. These bounds may ease the interpretations made by the machine learning practitioners when comparing different multi-class classifiers in unbalanced domains.

The rest of the paper is organised as follows: First, Section 2 introduces the general definitions and notation. Then, Section 3, Section 4, and Section 5 expose the above mentioned first, second, and third studies, respectively. Section 6 discusses the limitations of our research and discusses the potential lines of future work. Finally, Section 7 sums up the paper.

2 General Definitions and Notation

2.1 Hölder Means

As we make use of the Hölder means (Bullen, 2003), a.k.a. generalised means or power means, throughout the whole paper, we start by defining them as follows:

Definition 1 Let $\mathbf{a} = (a_1, \ldots, a_n)$ be a series of n positive real numbers with non-negative weights $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$ s.t. $\sum_i \zeta_i = 1$, then, a Hölder mean is a mean of the form

$$M_p(\mathbf{a}, \boldsymbol{\zeta}) = \left(\sum_{i=1}^{n} \zeta_i a_i^p \right)^{\frac{1}{p}} . \tag{1}$$

Here, $p \in \mathbb{R} \cup \{+\infty, -\infty\}$ is an affinely extended real number. This family of functions has several interesting properties:

1. **Hölder mean inequality:** $M_p(\mathbf{a}, \boldsymbol{\zeta}) \geq M_q(\mathbf{a}, \boldsymbol{\zeta})$ if $p > q$. The equality only holds for the case of $a_i = a_j, \forall i, j$.
2. **Inclusion of the Pythagorean means[1]:** The arithmetic mean corresponds to the case $p = 1$, the harmonic mean to $p = -1$, and the geometric mean is the limit of mean with an exponent p approaching to 0, i.e. $\lim_{p \to 0} M_p(\mathbf{a}, \boldsymbol{\zeta})$.
3. **Extremes:** $\lim_{p \to +\infty} M_p(\mathbf{a}, \boldsymbol{\zeta}) = \max\{\zeta_1 a_1, \ldots, \zeta_n a_n\}$ and $\lim_{p \to -\infty} M_p(\mathbf{a}, \boldsymbol{\zeta}) = \min\{\zeta_1 a_1, \ldots, \zeta_n a_n\}$.

[1] The classical definition of these means can be found in Table 1.

2.2 Unbalanced K-class Classification Problem

Let γ_K be a K-class classification problem with a generative model given by the generalised joint probability density function

$$\rho(\mathbf{x}, c|\boldsymbol{\theta}) = p(c)\rho(\mathbf{x}|c, \boldsymbol{\theta}). \tag{2}$$

Under the assumption of belonging to a given family of probability distributions, let $\rho(\mathbf{x}|c, \boldsymbol{\theta})$ be the conditional distribution of the feature space and let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ stand for the set of the model parameters which unambiguously determine the conditional probability distributions. Here, each $\boldsymbol{\theta}_i$ represents the set of parameters for the distribution of each class c_i. Also, let $p(c)$ be the distribution of the class probabilities. For simplicity of notation, henceforth, we denote $p(c)$ by $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_K)$, where each $\eta_i = p(c_i)$ is the probability of the categorical class c_i. Additionally, for convenience, hereafter, the special case of equiprobability, i.e. $\forall i, \eta_i = 1/K$, is denoted by $\mathbf{e}$. Therefore, according to (He and Garcia, 2009), a K-class classification problem is balanced if it exhibits a uniform class distribution. Otherwise, it is considered to be unbalanced. Formally,

$$\gamma_K \text{ is balanced } \iff \boldsymbol{\eta} = \mathbf{e}.$$

When the model is unbalanced, in the multi-class scenario ($K > 2$), we can differentiate two major types of class-imbalance (Wang and Yao, 2012; Ortigosa-Hernández et al., 2017): (i) multi-majority unbalanced problems, i.e. when most of the classes have a higher probability than equiprobability, and (ii) multi-minority unbalanced problems, i.e. when most of the class probabilities are below the equiprobability. Formally, let $\{\mathbb{M}, \mathbb{m}\}$ be a partition of the set $\{1, \ldots, K\}$ such that, $\forall i \in \mathbb{M}, \eta_i \geq 1/K$ (overrepresented) and $\forall j \in \mathbb{m}, \eta_j < 1/K$ (underrepresented), then

$$\gamma_K \text{ is multi-majority } \iff |\mathbb{M}| \geq K/2, \text{ and } \gamma_K \text{ is multi-minority } \iff |\mathbb{m}| > K/2.$$

By means of this definition, it can be presumed that having a balanced scenario is a hard condition to ensure in real-world problems. Here, $\boldsymbol{\eta}$ usually differs from equiprobability. For that reason, it is imperative to theoretically study not only the impact of the class distribution on the competitiveness of the proposed solutions and which performance scores are able to measure that impact, but also define more adequate learning systems which can effectively learn from highly skewed class distributions, i.e. $\exists i$ s.t. $(\eta_i \sim 1 \vee \eta_i \sim 0)$.

2.3 Traditional Supervised Learning Approaches

Solving a K-class classification problem, regardless of its imbalance extent, is equivalent to learning a function Ψ, known as a classifier, that maps a vector of observations $\mathbf{x}$, drawn from the generative function of eq. (2), into a categorical class c_i. In the supervised learning approach, the learning process is carried out by means of an optimisation algorithm, which provided with some labelled training data, drawn also from eq. (2), attempts to infer a function Ψ that minimises a certain loss function (Bartlett et al., 1996). Here, most of the traditional learning algorithms use the 0-1 loss or a surrogate loss which, by providing an upper bound for it, is also expected to minimise the 0-1 loss (Kanamori et al., 2013). This loss is also referred to as the misclassification error which can, in fact, be directly calculated by (1 − classification accuracy). This implies that those algorithms inherently maximise the classification accuracy and, therefore, they should asymptotically obtain classifiers close to the Bayes decision rule, a classifier which always obtains the highest classification accuracy for every K-class classification problem. In consequence, in this paper, we assume a framework where the generative function is known so that the Bayes decision rule (Young and Calvert, 1974) can be directly used as a representative of the classifiers resulting from the traditional approaches. By means of this approach, we can make use of the knowledge of prior works on the deficiencies of the traditional approaches in solving unbalanced problems (Axelsson, 2000; Weiss and Provost, 2003; Prati et al., 2004; Drummond and Holte, 2005; Prati et al., 2015) to complement our studies on class-imbalance.

Definition 2 Assuming $\rho(\mathbf{x}|c,\boldsymbol{\theta})$ and $\boldsymbol{\eta}$ to be known, the *Bayes decision rule* (BDR) is given by

$$\hat{c}_B = \arg\max_i \eta_i \rho(\mathbf{x}|c_i,\boldsymbol{\theta}_i). \tag{3}$$

Here, $\hat{c}_B$ is the categorical class assigned by the BDR to the observation $\mathbf{x}$. This rule has a corresponding probability of error

$$e_B = 1 - \sum_{i=1}^{K} \eta_i \int\limits_{\Omega_i} \rho(\mathbf{x}|c_i,\boldsymbol{\theta}_i) d\mathbf{x} \tag{4}$$

which is called the Bayes error and is the highest lower bound of the probability of error of any classifier. Here,

$$\Omega_i = \{\mathbf{x} : \eta_i \rho(\mathbf{x}|c_i,\boldsymbol{\theta}_i) - \max_{i' \neq i} \eta_{i'} \rho(\mathbf{x}|c_{i'},\boldsymbol{\theta}_{i'}) > 0\} \tag{5}$$

is the region where $\eta_i \rho(\mathbf{x}|c_i,\boldsymbol{\theta}_i)$ is maximum and, so, the instances are assigned to the class c_i by the BDR, for all i.

2.4 Performance Scores

Classifiers often produce misclassifications, and optimal classifiers are not exceptions. Thus, once a classifier is constructed, its associated discerning skill needs to be measured. When the generative function is assumed to be known, a common tool used for visualising the performance of a classifier is the true confusion matrix of a given classifier Ψ (Koço and Capponi, 2013). It is a square matrix of size K containing the mathematical expectations of classifying, with a classifier Ψ, as class c_i (rows) an example of class c_j (columns). By convection, the rows correspond to the predicted classes, while the columns to the actual classes (Elkan, 2001) Therefore, formally, the *true confusion matrix*[2] of the BDR is defined as $\mathbf{A}_B = [a_{i,j}]_{1 \leq i,j \leq K}$, where each $a_{i,j}$ is calculated by:

$$a_{i,j} = \mathbb{E}_{\mathbf{x}|c=c_j}[\hat{c}_B = c_i] = \eta_j \int\limits_{\Omega_i} \rho(\mathbf{x}|c_j,\boldsymbol{\theta}_j) d\mathbf{x}. \tag{6}$$

[2] In most the real world applications, where $\rho(\mathbf{x}, c)$ is unknown, an estimation of the true confusion matrix is utilised instead. It is usually referred to as the empirical confusion matrix, or simply as the confusion matrix.

Here, $\mathbb{E}[\cdot]$ stands for the mathematical expectation and Ω_i is defined as in eq. (5). It can be easily noticed that, assuming that the family of distributions for the feature space is fixed, the calculation of the true confusion matrix depends on only three parameters; the used classifier ($\Psi = B$, the BDR is assumed here), the class distribution ($\boldsymbol{\eta}$) and the parameters of the generative function ($\boldsymbol{\theta}$).

As stated in (Koço and Capponi, 2013), the confusion matrix is one of the most informative performance summaries that a multi-class learning system can rely on. Among other information, it contains how much the classifier is accurate on each class, and the way it tends to get confused among classes. However, it is often tedious not only determining the overall behaviour of a classifier from the confusion matrix, but also comparing several classifiers. Therefore, quantity measures which summarise the confusion matrix are often preferred in the literature (Japkowicz and Shah, 2011) and, therefore, used in this paper. These measures are known as performance scores (Santafe et al., 2015). Since *the best behaviour* may vary from one kind of problem to another, there are many different and diverse performance scores in the community. This diversity may, at times, obscure important information on the hypotheses or algorithms under consideration (Gu et al., 2009). Thus, it is fundamental to check in advance the adequateness of a performance score for assessing a determined classification problem so that the validity of the obtained results can be ensured. In this paper, we check this issue for the case of the multi-class class-imbalance domain: by studying the implication of the class distribution on the "already known" behaviour of the BDR as it is perceived by several numerical scores, it can be directly determined which performance scores are inadequate for excluding important information about the behaviour of the classifier. It is worth noticing that, since few papers have dealt with the class-imbalance problem in the multi-class framework without partitioning the problems into several independent binary problems, to the best of our knowledge, there is still no convention in which performance score should be used (Wang and Yao, 2012). This paper tries to shed some light on this issue by performing several theoretical studies on numerical performance scores.

Formally, we define a numerical performance score as $\mathcal{S}_{\Psi}(\boldsymbol{\eta}, \boldsymbol{\theta})$. Since it is single number summarising the confusion matrix, it also depends on the same parameters $\boldsymbol{\eta}$, $\boldsymbol{\theta}$ and $\Psi = B$. Then, the numerical performance scores can be mainly divided into two different groups: local scores, which only focus on the behaviour of one target class, and global scores, which summarise the performance of the classifier taking into account its behaviour in all classes. The list of performance scores considered and their formal definitions can be found in Table 1.

We use two well-known scores as *local performance scores*: precision $\mathcal{P}^i$ and recall $\mathcal{R}^i$. Whilst $\mathcal{P}^i$ assesses exactness, i.e. to what extent the predictions of a certain class c_i are correct, $\mathcal{R}^i$ assesses completeness, i.e. to what extent all examples of a certain class c_i are classified as so. Unfortunately, for their local property, they lose the global picture of the performance of the classifier; they are more useful when combined with other scores or applied to all classes (Gu et al., 2009). Therefore, most of the *global scores* are just functions which, by taking local performance scores applied to some/all classes, summarise the behaviour of the classifier according to a determined subjective criterion. In these studies, we have selected the global scores principally used or mentioned in the class-imbalance literature, which can be expressed as Hölder means (Bullen, 2003) among the recalls. The classification accuracy, $\mathcal{A}cc$, is a weighted Hölder mean with parameters $p = 1$ and

	Name	Notation	Formula
Local scores	Precision	$\mathcal{P}^i$	$a_{i,i}\left(\sum_{j=1}^{K} a_{j,i}\right)^{-1}$
	Recall	$\mathcal{R}^i$	$a_{i,i}\left(\sum_{j=1}^{K} a_{i,j}\right)^{-1}$
Global scores	Classification accuracy	$\mathcal{A}cc$	$\sum_{i=1}^{K} \eta_i \mathcal{R}^i$
	Arithmetic mean among the recalls (*a-mean*)	$\mathcal{A}$	$\sum_{i=1}^{K} \frac{1}{K}\mathcal{R}^i$
	Geometric mean among the recalls (*g-mean*)	$\mathcal{G}$	$\sqrt[K]{\prod_{i=1}^{K} \mathcal{R}^i}$
	Harmonic mean among the recalls (*h-mean*)	$\mathcal{H}$	$K\left(\sum_{i=1}^{K} \frac{1}{\mathcal{R}^i}\right)^{-1}$
	Maximum value among the recalls	max	$\max_i \mathcal{R}^i$
	Minimum value among the recalls	min	$\min_i \mathcal{R}^i$

Table 1: Numerical performance scores (by convention $0/0 = 1$).

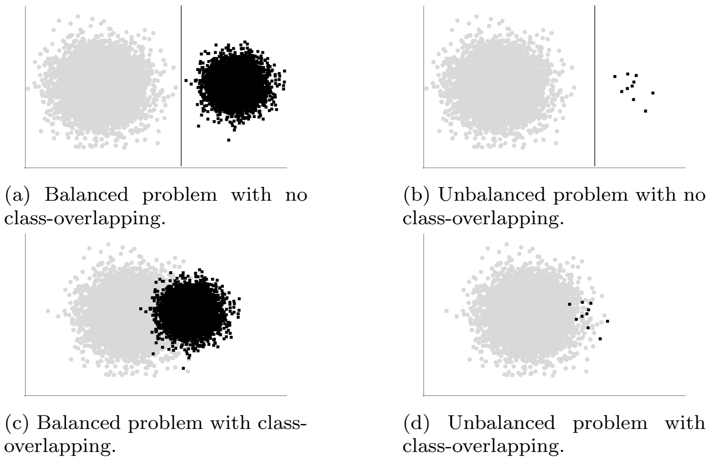
$\zeta_i = \eta_i$. As previously mentioned, the performance in underrepresented classes has very little impact on the measure when compared to the overrepresented classes (Gu et al., 2009; Di Martino et al., 2013). Due to this, most of the scores for unbalanced learning average over the recalls without weighting these values on the class probabilities. Therefore, all classes, over and underrepresented, share a common consideration in the score. The most-used scores in the class-imbalance literature are the Pythagorean means – arithmetic ($\mathcal{A}$), geometric ($\mathcal{G}$), and harmonic ($\mathcal{H}$) – over the recalls of the K-classes (Menon et al., 2013), which can be directly expressed as unweighted Hölder means with $p = 1$, $p = 0$ and $p = -1$, respectively. In the literature, they are referred to as a-mean, g-mean, and h-mean, respectively. Moreover, we also consider the extreme values of these unweighted Hölder means ($p = \infty$ and $p = -\infty$). They correspond to the maximum recall, *max*, and the minimum recall, *min* (among the classes), respectively. We believe that, since the are also used for multi-class problems in (Wang and Yao, 2012), they can also give valuable information in this complex scenario.

In (Menon et al., 2013), it is stated that the BDR is not an optimal decision rule for $\mathcal{A}$, i.e. a higher value for the score may be obtained with other classifiers. Unfortunately, in the literature, no further information is provided on either which the optimal classifier for $\mathcal{A}$ is or on whether the BDR is optimal for the other two broadly used unweighted Hölder means ($\mathcal{G}$ and $\mathcal{H}$) or for the extreme means. In the following sections, we also shed light on these questions.

3 First Study: Adequate Numerical Performance Scores for Multi-class Unbalanced Problems

In this section, our goal is to find out *which performance scores are adequate to determine the competitiveness of a classifier in the multi-class unbalanced domain.* Particularly, we seek to determine which performance scores succeed in expressing the long-studied performance detriment (Japkowicz and Stephen, 2002; Weiss, 2013) resulting from learning, in a classical manner, well-defined categories in moderate unbalanced scenarios.

3.1 A Novel Framework to Marginalise the Effect of the Class Distribution on the Performance Scores



(a) Balanced problem with no class-overlapping.

(b) Unbalanced problem with no class-overlapping.

(c) Balanced problem with class-overlapping.

(d) Unbalanced problem with class-overlapping.

Fig. 1: Relation between class-overlapping and class-imbalance.

Unfortunately, some authors (Denil and Trappenberg, 2010; López et al., 2013) have stated that the class distribution is not the only factor hindering the predictive power of the classifiers. These other factors, which appear in both balanced and unbalanced scenarios, are listed in (López et al., 2013) as the degree of overlapping among the classes (Prati et al., 2004), the training size (Kalayeh and Landgrebe, 1983), the noise in the data (Angluin and Laird, 1988), the presence of small disjuncts and the dispersity of some classes (Holte et al., 1989), among others (Ho and Basu, 2002). Moreover, these authors also argue that (i) the hindering factors have strong interdependences among them, and that (ii) these interdependences can modify the contribution of the class distribution to the behaviour of the classifier hampering its study. Thus, it seems that it is not trivial to isolate the implication of the class distribution on the performance of the classifier so that, therefore, the adequateness of the numerical performance scores to capture this implication can be determined. Special care must be taken to marginalise out all the rest of the hindering factors. Fortunately, all of these causes, with the exception of class-overlapping, are only dependent on the nature of the training dataset. Since the BDR only depends on the generative function, this classifier allows us to learn well-defined concepts and, therefore, to omit practically all these factors which may modify the real impact that the imbalance extent has on the performance of the classifier. However, the effect of the class-overlapping is harder to eliminate; it depends on the local probability distributions of the feature space (Denil and Trappenberg, 2010), i.e. class-overlapping, like class-imbalance, is an intrinsic characteristic of the generative model. Hence, the overlapping-imbalance dependence must be exhaustively studied in order to find a legitimate manner to remove the class-overlapping from this puzzling situation.

3.1.1 Class-overlapping and Class-imbalance Relationship

First, we set up an example, which is similar to the experiment performed in (Prati et al., 2004), in order to clarify the aforementioned dependency and how both the class-overlapping and class-imbalance factors may hinder the behaviour of the inferred classifier. Figure 1 shows four binary problems sharing two different degrees of class-imbalance and two different degrees of class-overlapping. For each problem, the data has been created by sampling two bivariate Gaussian distributions, one distribution for each class, i.e. $\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}, i = \{1, 2\}$. To simulate no class-overlapping, we choose two Gaussians whose means are far from each other (Figures 1a and 1b), and to simulate the opposite, we shorten the distance between these means (Figures 1c and 1d). Regarding the class-imbalance setting, it is simulated by sampling $1,000$ instances of each class for balanced problems (Figures 1a and 1c), and by sampling $1,000$ instances for the majority class and only 10 samples for the minority, for the case of unbalanced problems (Figures 1b and 1d). By just having an overall look at Figure 1, it can be easily noticed how the combination of both factors (class-imbalance and class-overlapping) has a straight effect on the issue of discriminating among the classes and, therefore, on the performance of the inferred classifier. When there is no class-overlapping, as shown in the first row of the figures, the class distribution does not hinder the predictive power of the resulting classifier. In both scenarios, a simple and perfect discriminant linear classifier can be easily drawn. This classifier is represented by a straight continuous line in the figure. However, in the second row, the situation is completely different. When both classes are balanced (Figure 1c), a classifier with a tolerable recall for both classes can be learned. Unfortunately, in the unbalanced case (Figure 1d), the lack of enough examples for the minority class hinders the process of discriminating that class; any intuitively chosen classifier will be incompetent, i.e. it will have a low recall for the minority class. This example concurs with the literature (Prati et al., 2004), where it is stated that (i) only when the class-overlapping is non-zero, the influence of the class distribution on the competitiveness of the inferred classifier is noticeable, and that (ii) the influence of class-overlapping into the learning process is even stronger than class-imbalance.

3.1.2 Isolating Class-imbalance from Class-overlapping

Whilst class-overlapping is an old stalwart in the literature for being broadly studied (Basu and Ho, 2006), to the best of our knowledge, no prior work in the literature has isolated the impact that the class distribution has on the performance of the classifiers or has explored which performance scores are able to appropriately capture this potential fluctuation of performance. In order to bridge these gaps, we propose a function which, by properly cancelling the effect of the class-overlapping, returns the impact of the class distribution on a chosen numerical performance score for assessing the inferred classifier (here the BDR):

Definition 3 Let $\boldsymbol{\Theta}$ represent the space of parameters for a fixed family of distributions over the feature space for classification problems with K classes. Let $\mathcal{S}_B(\boldsymbol{\eta}, \boldsymbol{\theta})$ be the value of a performance score $\mathcal{S}$ assessing the behaviour of the BDR on a K-class classification problem, γ_K, with a class distribution equal to $\boldsymbol{\eta}$ and parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Also, let $\mathcal{S}_B(\boldsymbol{e}, \boldsymbol{\theta})$ be the value of S evaluating the BDR inferred from the balanced version of γ_K. Therefore, the *influence function*, $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$, of the K-class distribution $\boldsymbol{\eta}$ on the performance score[3] $\mathcal{S}$ using the BDR as a classifier is defined as follows:

$$\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta}) = \int\limits_{\boldsymbol{\Theta}} [\mathcal{S}_B(\boldsymbol{e}, \boldsymbol{\theta}) - \mathcal{S}_B(\boldsymbol{\eta}, \boldsymbol{\theta})] d\boldsymbol{\theta}. \tag{7}$$

It can be easily seen that the previous equation fulfills our couple of objectives. First, the implication of the class-overlapping on the behaviour of the inferred classifier is taken out of the equation by means of the integration; every possible set of parameter values for the probability distributions of the generative model showing a non-zero degree of overlapping among the classes is marginalised out. As a result, the average influence of the class distribution on the behaviour of the inferred classifier, as conceived by the performance score $\mathcal{S}$, can then be quantified and studied: assuming a fixed parametric family for the local distribution, positive values of $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ denote that the performance score $\mathcal{S}$ obtains, on average, higher values for the BDR when it is used on a balanced scenario rather than when it is inferred from a class distribution $\boldsymbol{\eta}$. Negative values mean the opposite; the BDR achieves, in general, worse values for $\mathcal{S}$ in the balanced scenario.

3.2 Identifying the Adequate Performance Scores

By just plotting $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta}), \forall \boldsymbol{\eta}$ using different performance scores, we can perceive, at a glance, how these scores differ in measuring the goodness of the traditional classifiers. However, we are incapable of determining which of them are adequate to validate classifiers in unbalanced domains. Thus, in order to accomplish the latter task, we must take advantage of the prior studies (Axelsson, 2000; Drummond and Holte, 2005; Gu et al., 2009) on the expected behaviour of the popular BDR when it faces skewed class distributions. By doing so, we can determine the shape that an adequate performance score should have for the influence function in the class-imbalance spectrum. Then, this shape can be straightforwardly used as a representative case to discern which performance scores are appropriate for the class-imbalance scenario. For this purpose, we focus on the long discussed

[3] In this paper, we assume a positive correlation between the value of the performance score and the behaviour of the classifier, i.e. higher values of $\mathcal{S}$ represent higher performances. In the event of a negative correlation, the sign of $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ must be reversed.

hindering behaviours of the BDR: A good prediction power of the BDR is only guaranteed for the majority classes (Axelsson, 2000; Gu et al., 2009). Additionally, in highly unbalanced situations, the BDR often performs little better than a dummy classifier always predicting the most common classes (Drummond and Holte, 2005). This happens because the BDR suffers from one of the cornerstones of Bayesian statistics: the *base-rate fallacy* (Matthews, 1996, 1997; Axelsson, 2000). This formal fallacy states that if presented with generic information about the probability of an event (base rate) and specific information about the case in question, the mind tends to ignore the former and focus on the latter. Within the class-imbalance problem, the conditional probabilities of the minority classes are the base rate, which are often ignored by the BDR when having an overwhelming class probability for the other classes, and, thus, traditional classifiers behave like a dummy. Examples of these phenomenon when applying the Bayes theorem can be found in the cited papers.

By means of this discussion, it can be seen that (i) the highest performance of the BDR for all classes occurs when they share the same class probability, i.e. the balanced scenario. That setting has zero probability of suffering from the base-rate fallacy since the BDR is not provided with class probabilities which may obscure the conditional distribution of the feature space of any class. (ii) In any other setting the BDR has a positive probability of suffering the fallacy, which grows towards the skewed class distributions. In those extremes, the performance of the BDR should be highly penalized due to the fact that it falls into the base-rate fallacy. After this argument, we may define the shape of the influence function of an adequate performance score $\mathcal{S}$ with independence of the generative model as follows:

Properties of an adequate performance score. *A performance score $\mathcal{S}$ is successful in being adequate to determine the competitiveness of a classifier Ψ in the class-imbalance scenario if, assuming a scenario where a classifier is inferred by directly minimising a* 0-1 *loss (maximising the classification accuracy),*

(a) *its influence function is positive for almost any $\boldsymbol{\eta} \neq \mathbf{e}$, and*
(b) *it shows a negative correlation to the minority class probability, i.e. $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ grows as the Euclidean distance between $\boldsymbol{\eta}$ and $\mathbf{e}$ gets larger.*

3.2.1 Experimental Model for the Study

When the generative model is known, in theory, eq. (7) is obtainable. However, solving an integral of such characteristics independently of the parametric family is intricate. Therefore, we assume a parametric family with the following characteristics: (i) simple enough to be able to fully interpret the results and complete enough to be able to represent real world problems, (ii) a set of parameters which allows us to unambiguously represent each particular model as a single point. For these reasons, as a generative model we make use of a *univariate Gaussian identifiable mixture of components with unit variances whose means are separated by a fixed overlapping factor δ*. Under this assumption, since each $\boldsymbol{\theta}_i = \{\mu_i, \sigma_i\}$ is such that $\mu_i = (i-1)\delta$ and $\sigma_i = 1$, the parameters can be simplified to just $\boldsymbol{\theta} = \{\delta\}$ and eq. (7) be rewritten as:

$$\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta}) = \int_0^\infty [\mathcal{S}_B(\boldsymbol{e}, \delta) - \mathcal{S}_B(\boldsymbol{\eta}, \delta)] d\delta. \tag{8}$$

In Figure 2 and 3 (local and global scores for binary problems, respectively) and Figure 4 (global scores in the multi-class framework) we numerically approximate[4], using Mathematica (Wolfram Research Inc., 2014), the influence function of eq. (20), for each performance score of Table 1 and through the whole class-imbalance spectrum. For *binary problems*, we numerically approximate the following equation:

$$\mathbb{I}_2^{\mathcal{S}}(\eta) = \int_{0.01}^{10} [\mathcal{S}_B(\frac{1}{2}, \delta) - \mathcal{S}_B(\eta, \delta)] d\delta. \tag{9}$$

Here, since, in binary problems, there are only two class probabilities which are complementary to each other, we can simplify $\boldsymbol{\eta}$ to η ($\eta_1 = \eta$ and $\eta_2 = 1 - \eta$). Also, note that the integral is calculated in the domain $[0.01, 10]$, instead of $[0, \infty)$. We choose an upper limit distance of 10 because, for unit variances, it is almost equivalent to not overlapping. Regarding the lower limit, we choose 0.01 in order to avoid the singularity $\delta = 0$[5].

Regarding the *multi-class framework*, we perform the same case study. However, as this setting is more complex, several changes are made. First, for the sake of clarity in the presentation of the results, we simplify the multi-class framework to the following: $(K-1)$ classes are equiprobable among them, with probability $\eta = \frac{1}{K} - \frac{\epsilon}{K-1}$, whilst the remainder has a probability $\eta_1 = \frac{1}{K} + \epsilon$. Then, by means of just one parameter $\epsilon \in [-\frac{1}{K}, \frac{K-1}{K}]$ which determines the imbalance extent, we can easily study the hindrance produced by the class distribution in the multi-class framework and present the results in a bi-dimensional plot. Note that the value $\epsilon = 0$ corresponds to the balanced setting. Therefore, we numerically approximate the following equation:

$$\mathbb{I}_K^{\mathcal{S}}(\epsilon) = \int_{0.01}^{10} [\mathcal{S}_B(0, \delta) - \mathcal{S}_B(\epsilon, \delta)] d\delta. \tag{10}$$

Here, all the parameters[6] are the same as in binary problems except for the defined ϵ. Due to the limiting space and since similar results are obtained for any arbitrary K, in this manuscript, only the case of $K = 3$ is presented. The source code to calculate $\mathbb{I}_K^{\mathcal{S}}(\epsilon)$ for any number K of classes can be downloaded from `http://github.com/jonathanSS/ClassImbalanceStudies`.

[4] **NIntegrate** with all options set to default.

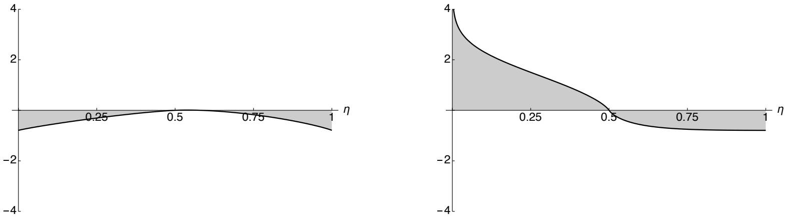
[5] Assuming an upper limit of 10 instead of infinity, the degree of overlapping (e_B) is 2.9×10^{-7} rather than 0. Analogously, assuming a lower limit of 0.01 means that, instead of having an overlapping of 0.5, we have 0.498.

[6] Assuming an upper limit of 10, the degree of overlapping for $K = \{3, 4, 5\}$ is $\{3.8, 4.3, 4.6\} \times 10^{-7}$ instead of 0. The overlapping is $\{0.64, 0.72, 0.77\}$ instead of $(K-1)/K$ for the lower limit.

3.3 Results and Discussion

3.3.1 Binary Problems

Figure 2 shows the value of the function $\mathbb{I}_2^{\mathcal{S}}(\eta)$ over the domain $0 \leq \eta \leq 1$ for the local performance scores of Table 1 when using the BDR. Specifically, the precision, $\mathcal{P}^1$ is shown in Figure 2a and the recall, $\mathcal{R}^1$, in Figure 2b, both for class c_1. Note that, for local scores, the diagrams for c_2 are omitted. This is due to the fact that they are a reflection, with respect to the imaginary vertical axis $\eta = 0.5$ of those of c_1. Next, the values of the influence function for the global performance scores for binary problems are presented in Figure 3. There, the accuracy, $\mathcal{A}cc$ (Figure 3a), the maximum recall, *max* (Figure 3b), the arithmetic mean, $\mathcal{A}$ (Figure 3c), the geometric mean, $\mathcal{G}$ (Figure 3d), the harmonic mean, $\mathcal{H}$ (Figure 3e), and the minimum recall, *min* (Figure 3f) are displayed. All plots share the same style; the x-axis represents the value of η and the y-axis, $\mathbb{I}_2^{\mathcal{S}}(\eta)$. The area between the function and the x-axis is highlighted for visual purposes.

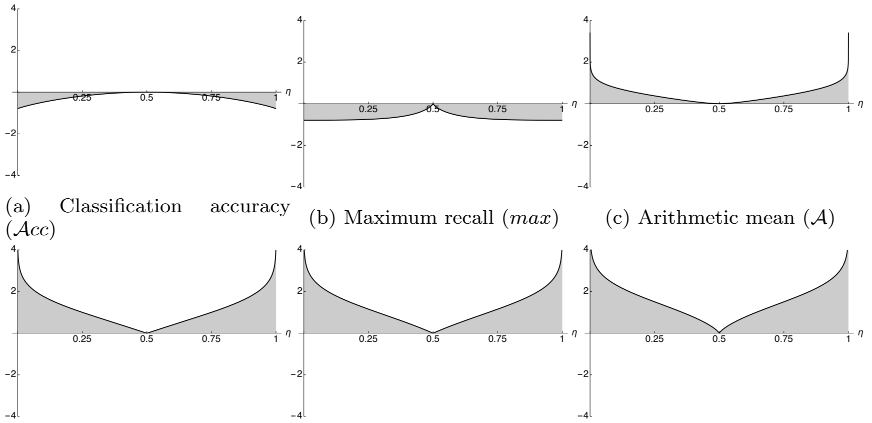


(a) Precision ($\mathcal{P}^1$)

(b) Recall ($\mathcal{R}^1$)

Fig. 2: The influence function in binary problems, $\mathbb{I}_2^{\mathcal{S}}(\eta)$, for both precision and recall throughout the range $0 \leq \eta \leq 1$ (c_1).

Several insights about **the diverse behaviour of the performance scores** can be extracted from just a glimpse at Figures 2 and Figure 3. First, regarding the *local performance scores* studied, they are completely different to each other; whilst the influence function for $\mathcal{P}^1$ shows a negative behaviour for all $\eta \neq 0.5$, $\mathbb{I}_2^{\mathcal{R}^1}(\eta)$ is positive when c_1 is the minority class and takes negative values in the opposite case. Concerning the most common *global performance scores* of Figure 3, two different major behaviours can be easily detected: on one hand, we have $\mathcal{A}cc$; its influence function shows a negative behaviour for the whole spectrum of unbalanced settings. On the other hand, we have the performance scores commonly used in unbalanced domains, $\mathcal{A}$, $\mathcal{G}$, and $\mathcal{H}$. These scores show a positive influence function for all the values of η and they share a common shape; their lowest value is in the equiprobability, and, from there, they strictly increase to the extremes ($\eta \sim 0$ or $\eta \sim 1$), where they, finally, achieve an exponential growth rate. The two extreme Hölder means introduced in this paper can also be categorised into these two groups. While *max* shares a similar behaviour to the $\mathcal{A}cc$, *min* behaves closely to the scores utilised for unbalanced data. Lastly, it is also worth noticing the similarity between the shape of the influence functions for $\mathcal{P}^1$ and $\mathcal{A}cc$ in this



(a) Classification accuracy ($\mathcal{A}cc$)

(b) Maximum recall (max)

(c) Arithmetic mean ($\mathcal{A}$)

(d) Geometric mean ($\mathcal{G}$)

(e) Harmonic mean ($\mathcal{H}$)

(f) Minimum recall (min)

Fig. 3: The influence function in binary problems, $\mathbb{I}_2^{\mathcal{S}}(\eta)$, for each global performance score throughout the range $0 \leq \eta \leq 1$.

binary setting. Despite the fact that one is a local score and the other is a global score, they share a close response to a changing class probability distribution.

Thereafter, we deal with the issue of determining the **adequateness of the studied performance scores** by comparing the influence function of these scores to the representative case, defined in Proposition 1, of the influence function that an adequate performance score should have:

– Although *local performance scores* are not sufficient to summarise the overall performance for losing the global picture, they give partial valuable advice: several global performance scores are just Hölder means (Bullen, 2003) or other averaging functions over the local performances, e.g. $\mathcal{A}cc$, $\mathcal{G}$, etc (Santafe et al., 2015). In order to study the adequateness of these scores, we focus on the most problematic section of the plots: the values of $\eta < 0.5$. There, the class c_1 is the minority class, and due to the fact that the use of the BDR is assumed, the misclassification rate of c_1 grows considerably, on average, as η decreases. As for $\mathcal{P}^1$, although it is extensively used in the literature, it is not adequate enough for this scenario since it does not penalise the decrease in the prediction power of the BDR for c_1 when this class becomes minority: $\mathbb{I}_2^{\mathcal{P}^1}(\eta)$ takes negative values for $\eta < 0.5$ (Proposition 1). This is due to the fact that, for low values of η, the BDR achieves, for $\mathcal{P}^1$, higher values than in the balanced scenario. There, it only classifies examples as c_1 if they are "undoubtedly" drawn from that minority class, i.e. examples that are far away from the density of c_2. In such schemes, the ratio of the correctly classified examples over the predicted ones is considerably high, and it increases as the distribution grows in skewness. In the extreme, we have the convention $0/0 = 1$. Due to the popularity of precision in the class-imbalance literature, further research on this topic is indispensable to determine whether it is suitable to use precision on minority classes when a different generative model is assumed. On the contrary, $\mathcal{R}^1$ shows a more appropriate description of the effect of the class distribution on the resulting classifier; $\mathbb{I}_2^{\mathcal{R}^1}(\eta)$ is positive for $\eta < 0.5$. Therefore, regarding using precision or recall as parameters for a global measure function, two conclusions can be extracted:

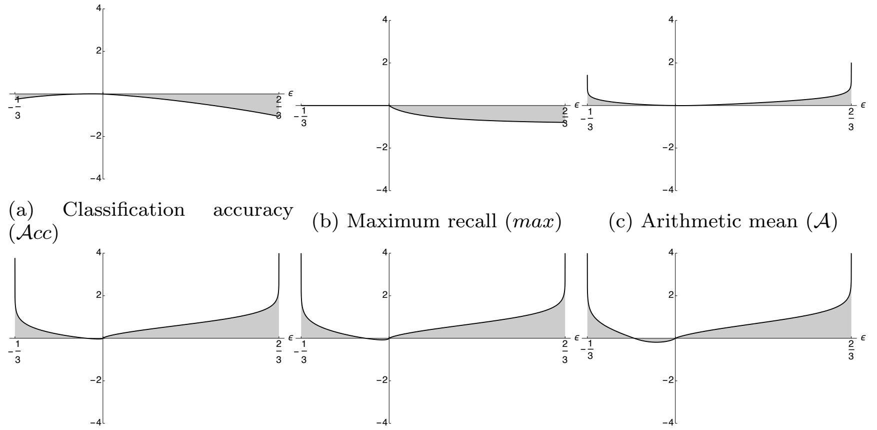
(i) Unweighted Hölder means among the precisions are inadequate due to the fact that, as c_2 is a reflection of c_1, the values of $\mathbb{I}_2^{\mathcal{S}}(\eta)$ for an average function will always be under the x-axis. Other factors are more influential in the score rather than a good prediction of the minority class.

(ii) Averaging recalls is a better choice since, for certain Hölder means, a positive value of the influence function for almost all η will be shown. This is due to the fact that the positive part of $\mathbb{I}_2^{\mathcal{S}}(\eta)$ for the recalls of c_1 and c_2 is always greater than the negative part.

– In relation to the *global performance scores*, our study supports the conclusion of the state-of-the-art literature stating that $\mathcal{A}cc$ is not an adequate score for unbalanced problems (Gu et al., 2009). Here, Proposition 1 is not required to determine the insensitivity of the classification accuracy to the class-imbalance extent. Since its influence function shares shape with the inadequate function $\mathbb{I}_2^{\mathcal{P}^1}(\eta)$, $\mathcal{A}cc$ can be directly appointed as an inadequate performance score. By extension, max is not an appropriate score as well. The reason for the inadequateness of the latter is that it only takes into account the maximum recall, which usually coincides with the recall of the majority class as the BDR favours this kind of classes. Contrastingly, the performance scores of the other behavioural group ($\mathcal{A}$, $\mathcal{G}$, $\mathcal{H}$ and min) are adequate for the class-imbalance problem since their influence functions meet the two conditions of Proposition 1. In this case, $\mathcal{A}$ is the score showing the lowest sensitivity – its influence function has the smoothest shape –, which is followed by $\mathcal{G}$, then $\mathcal{H}$ and, finally, min. There it can be seen that, in these scores, the behaviour of the BDR is penalised when the class distribution is not balanced, i.e. the misclassification of the minority class also produces high drops in these performance scores. In the extremes, we discover that the situation of the BDR acting as a dummy classifier in situations of extremely skewed class distribution is strongly penalised. There, the influence functions exponentially grow as η gets closer to either 0 or 1. In general terms, unweighted Hölder means over the recalls with $p > 1$ will be inadequate for the class-imbalance extent since greater recalls have more presence in the score than lower recalls. On the contrary, means with $p < 1$ would be adequate to determine the competitiveness of a classifier as lower recalls have more influence on the resulting score. This can be easily drawn from the Hölder mean inequality (Definition 1) for the assumed model. The general validity of this conclusion for any generative model, on the contrary, can be straightforwardly inferred from the third study of this paper (Section 5), where bounds for these performance scores ensuring competitive classifiers are proposed.

3.3.2 Multi-class Problems

Figure 4 presents results for accuracy, $\mathcal{A}cc$ (Figure 4a), maximum recall, max (Figure 4b), arithmetic mean, $\mathcal{A}$ (Figure 4c), geometric mean, $\mathcal{G}$ (Figure 4d), harmonic mean, $\mathcal{H}$ (Figure 4e), and minimum recall, min (Figure 4f). The local performance scores are omitted in the multi-class scenario since the loss of the global picture becomes aggravated when more than two classes are used. All these figures are similar to the binary functions but the x-axis shows, in the current case, the parameter ϵ over the range $-1/3 \leq \epsilon \leq 2/3$. There, the values $\epsilon < 0$ represent



(a) Classification accuracy ($\mathcal{A}cc$)

(b) Maximum recall (max)

(c) Arithmetic mean ($\mathcal{A}$)

(d) Geometric mean ($\mathcal{G}$)

(e) Harmonic mean ($\mathcal{H}$)

(f) Minimum recall (min)

Fig. 4: The influence function in ternary problems, $\mathbb{I}_3^{\mathcal{S}}(\epsilon)$, for each global performance score throughout the range $-\frac{1}{3} \leq \epsilon \leq \frac{2}{3}$.

the multi-majority version of the problem and $\epsilon > 0$ the multi-minority version. In view of the results, the previous conclusions generalise well to multi-class problems:

Regarding the **diversity of the global performance scores**, the same two major groups can be perceived; $\mathcal{A}cc$ and max on one hand, and the Pythagorean means and min on the other. In the latter group, it can be seen that, due to the Hölder mean inequality, the scores can also be arranged by their sensitivity to the class-imbalance extent as $\mathbb{I}_K^{\mathcal{A}}(\epsilon) \leq \mathbb{I}_K^{\mathcal{G}}(\epsilon) \leq \mathbb{I}_K^{\mathcal{H}}(\epsilon) \leq \mathbb{I}_K^{min}(\epsilon)$. Next, concerning the **adequateness of the performance scores**, it can be concluded that:

(i) As their influence functions violate the conditions of Proposition 1 ($\mathbb{I}_K^{\mathcal{S}}(\epsilon) \leq 0, \forall\epsilon$), $\mathcal{A}cc$ and max are inadequate scores for assessing unbalanced problems.

(ii) The unweighted Pythagorean means over the recalls and min are adequate to assess the competitiveness of classifiers in this complex domain; their influence functions truly capture the hindering behaviours of the BDR (Proposition 1).

It is also worth mentioning that the generalisation of the study for the whole set of Hölder means also applies well for multi-class problems. Moreover, it can be seen that achieving a high value of an adequate performance score for multi-minority problems is far more difficult than for multi-majority; $\mathbb{I}_K^{\mathcal{S}}(\epsilon)$ is always higher for positive values of ϵ. Finally, we want to point out the particularity that appears in Figure 4d ($\mathcal{G}$), in Figure 4e ($\mathcal{H}$) and in Figure 4f (min); $\mathbb{I}_K^{\mathcal{S}}(\epsilon)$ is below 0 for the negative values of ϵ near the balance situation, i.e. for $\epsilon \to 0^-$. In Section 4, we properly address this interesting particularity.

4 Second Study: On the Inherent Scores of the Proposals for Multi-class Unbalanced Domains

The fact that any learning task can be viewed as an optimisation problem leads us to the second unanswered question exposed in the introduction; *Since there* *are many different and diverse performance scores available, which performance scores are maximised in the most predominant learning solutions designed to deal with skewed classes? Are they the adequate performance scores detected in the first study?* To the best of our knowledge, little effort has been made in the literature towards answering these questions. Yet, the interpretation of previous works with regards to this issue seems to be in contradiction. On one hand, there are works claiming that there is no algorithmic solution to the class-imbalance problem since the BDR establishes a fundamental limit in the performance of any classifier (Drummond and Holte, 2005). The argument is that the BDR is asymptotically sought in this domain, i.e. all the solutions proposed in the class-imbalance domain can be categorised as traditional supervised learning approaches since they maximise $\mathcal{A}cc$ (by minimising the 0-1 loss). So, therefore, no competitive classifier can be proposed for these problems. On the other hand, there are a large number of methodological contributions designed to overcome the intrinsic difficulties of this intricate domain. Moreover, most of these solutions report positive results (He and Garcia, 2009). This suggests that they produce classifiers which are maximised for a more appropriate score to the class imbalance scenario than $\mathcal{A}cc$.

Therefore, in order to enlighten this apparent contradiction of statements and provide answers, we, now, theoretically scrutinise the most predominant solutions of the state-of-the-art literature. Thus, we can devise what the decision rules behind the proposals of the literature are, their competitiveness and their inherently maximised performance scores.

4.1 Major Solutions for the Class-imbalance Problem

Concerning the unbalanced learning literature, most of its methodological contributions can be mainly categorised into four main kinds of approaches: (1) data sampling (Batista et al., 2004), (2) cost-sensitive learning (Liu and Zhou, 2006), (3) algorithmic modification (Galar et al., 2012), and (4) the use of ensembles (Galar et al., 2012). In this paper, we study data sampling and cost-sensitive learning due to the following facts: (i) they dominate the current research efforts (He and Garcia, 2009). (ii) Both approaches are more transparent to the Bayesian decision theory since they never behave as blackboxes (Rodríguez et al., 2014). (iii) Additionally, there is a lack of a unified framework for the heterogeneous approaches of algorithmic modification and ensembles which impedes their categorisation and study (Galar et al., 2012). (iv) Most of the current efforts in solving unbalanced problems are focused on just binary classification problems. Thus, the effectiveness of the non-dominant (other than data-sampling and cost-sensitive learning) approaches for the multi-class framework is, in most of the proposals, still unknown (Wang and Yao, 2012).

4.1.1 Data Sampling

By means of data sampling (or rescaling) techniques, the training dataset is modified in order to provide a more balanced class distribution (Sáez et al., 2016) so that, when classical supervised algorithms (Wu et al., 2007) are used in the learning process, the resulting classifiers are not biased towards the majority classes (Chawla et al., 2002; Batista et al., 2004). In other words, this approach allows the traditional learning systems to learn from a *"safe scenario"* where the probabilities of the classes hinder the learnt classifiers in an insignificant or null manner. The two simplest methods used are *random over-sampling* (ROS) and *random under-sampling* (RUS). While ROS balances the class distribution by the random replication of the examples of the minority classes of the training dataset, the balance distribution is achieved in RUS by the random removal of examples of the overrepresented classes. From a theoretical point of view, both methods are equivalent. They balance the class distribution up to having a uniform class distribution or, at least, a hardly noticeable unbalanced distribution. Formally, data sampling methods, instead of using the generative model as defined in eq. (2), modify the training dataset in such a way that they try to learn a classifier from the following model:

$$\rho'(\mathbf{x}, c|\boldsymbol{\theta}) = p'(c)\rho(\mathbf{x}|c, \boldsymbol{\theta}). \tag{11}$$

where $p'(c)$ is a multinomial distribution near the equiprobability, i.e. close to $\mathbf{e}$. Then, by directly applying the BDR, our surrogate of the traditional learning approach, over the modified generative model, we reach the following decision rule:

$$\hat{c}_B = \arg\max_i \eta'_i \rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i), \text{ where } \eta'_i \sim \frac{1}{K}. \tag{12}$$

Thus, it can be concluded that the joint use of a data sampling technique and the BDR is practically equivalent (equal for $\boldsymbol{\eta}' = \mathbf{e}$) to use the equiprobable Bayes decision rule. The latter rule does not take into account the class distribution and it is defined as:

Definition 4 Assuming $\rho(\mathbf{x}|c, \boldsymbol{\theta})$ and $\boldsymbol{\eta}$ to be known, the *equiprobable Bayes decision rule* (EDR) is given by

$$\hat{c}_E = \arg\max_i \rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i). \tag{13}$$

Figure 5 summarises our theoretical reasoning behind the data sampling approach. However, when dealing with real-world problems, where the generative function is usually unknown, each method introduces its own set of problematic consequences that might hinder the learning task (He and Garcia, 2009); while ROS may produce over-fitting towards the minority classes, RUS may discard data which are potentially important to the classification process. In order to overcome these problems, some heuristic methods have been proposed in the literature; Tomek links (Tomek, 1976), condensed nearest neighbour rule (CNN) (Hart, 1968), one-side selection (OSS) (Kubat et al., 1997), synthetic minority oversampling techniques (SMOTE) (Chawla et al., 2002), and combinations among them (Batista et al., 2004). The main motivation behind some of these proposals is not only to balance the training data, but also to remove noisy examples lying on the wrong side of the decision region (Batista et al., 2004). Due to its nature and despite the fact that prior works (He and Garcia, 2009; Galar et al., 2012) have shown that data sampling is usually a positive practical solution, this methodology has mainly been criticised due to altering the original class distribution (He and Garcia, 2009), or even, the distribution of the feature space as happens with SMOTE.

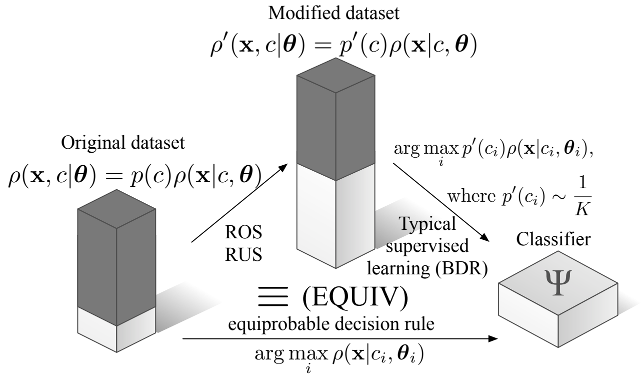

Modified dataset
$\rho'(\mathbf{x}, c|\boldsymbol{\theta}) = p'(c)\rho(\mathbf{x}|c, \boldsymbol{\theta})$
Original dataset
$\rho(\mathbf{x}, c|\boldsymbol{\theta}) = p(c)\rho(\mathbf{x}|c, \boldsymbol{\theta})$
ROS
RUS
$\arg\max_i p'(c_i)\rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i)$,
where $p'(c_i) \sim \frac{1}{K}$
Typical supervised learning (BDR)
Classifier
Ψ
$\equiv$ (EQUIV)
equiprobable decision rule
$\arg\max_i \rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i)$

Fig. 5: Data sampling techniques.

4.1.2 Cost-sensitive Learning

Cost-sensitive learning (Elkan, 2001; Zhou and Lui, 2010; Krishnapuram et al., 2011) is a paradigm which studies and solves classification problems where some types of misclassifications may be more crucial than others, e.g. rejecting a valid credit card transaction may cause an inconvenience while approving a large fraudulent transaction may have very negative consequences. Usually, the costs associated to misclassifications are given in the form of a matrix called cost matrix. It is a numerical representation of the cost of classifying examples from one class to another. Formally, the cost matrix, $\mathbf{B} = [b_{i,j}]_{1 \leq i,j \leq K}$, is a square matrix of size K where each element $b_{i,j}$ is, by convention, the cost of classifying as class c_i (rows) an example of class c_j (columns) (Elkan, 2001). In this scenario, the main objective of a cost-sensitive learning method is to learn a classifier minimising the overall cost on the training dataset so that it may, afterwards, classify new instances into the classes showing the lowest expected costs. Two main approaches can be found in the literature to deal with cost-sensitive problems (López et al., 2013): (i) *direct methods*, where cost-sensitive learning algorithms using the cost matrix are built, and (ii) *meta learning*, where preprocessing or postprocessing mechanisms for the training data or for the output, respectively, are used so that any traditional learning algorithm can be made cost-sensitive. Note that data sampling may be included in the latter (Elkan, 2001; Zhou and Lui, 2010) since, by just changing the class probabilities, the misclassification costs are modified.

In situations of disproportionate class probabilities, cost-sensitive techniques are deemed as good solutions; the proposed solutions usually assume that the cost of misclassifying an example from a minority class is higher than that of misclassifying an example from a majority class. Thereby, the ratio of correct classifications for the minority classes may be improved (Liu and Zhou, 2006; Thai-Nghe et al., 2010). Moreover, several authors (Elkan, 2001; He and Garcia, 2009) claim that cost-sensitive learning is superior to data sampling methods. However, although this framework can significantly improve the performance, it takes for granted the availability of a cost matrix and its associated cost items. Unfortunately, establishing a cost representation of a given unbalanced domain can be particularly challenging. For that reason, several proposals in the literature appoint cost matrices for class-imbalance problems. The most utilised approach is the one proposed in Japkowicz and Stephen (2002), which suggests the use of non-uniform error costs defined by means of the class-imbalance ratio presented in the dataset, i.e. $b_{i,j} = \eta_i/\eta_j$. In other words, modify the relative costs associated to misclassifying any pair of classes so that the imbalance ratio is compensated. Liu and Zhou (2006) proposed an equivalent method allowing the introduction of other costs so that the problem of simultaneously handling unequal misclassification costs and class-imbalance can be handled.

Once the cost matrix is defined and assuming the generative model to be known, an example can be directly classified by means of the BDR for unequal misclassification costs[7] (CS-BDR) (Berger, 1985). This rule minimises the expected misclassification costs as expressed in the cost matrix instead of a 0-1 loss as the traditional solutions seek.

Next, by applying the imbalance ratio compensation approach (Japkowicz and Stephen, 2002; Liu and Zhou, 2006), the CS-BDR, which is optimal for the Hölder mean with $p = 1$ and $\zeta_i = W_i\eta_i$, can be defined as

$$\hat{c}_B = \arg\max_i W_i \eta_i \rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i), \text{ where } W_i = \sum_{j=1}^{K} \frac{1}{b_{i,j}}. \tag{14}$$

In order to discover the actual decision behind the cost-sensitive methods, we now just substitute the values of the W_i in eq. (14), also reaching the EDR:

$$\hat{c}_B = \arg\max_i \eta_i \sum_{j=1}^{K} \frac{1}{b_{i,j}} \rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i) = \arg\max_i \rho(\mathbf{x}|c_i, \boldsymbol{\theta}_i). \tag{15}$$

4.2 An Analysis of the Equiprobable Decision Rule

In the previous paragraphs, it is shown that the asymptotic decision rule sought in the studied approaches is the EDR (Definition 4). Additionally, we are confident enough to think that, also for the remaining class-imbalance approaches, as long as their intention is to overcome the hindering behaviours expressed in Section 3.2 by reducing the bias of the traditional learners towards the majority classes, they will seek a decision rule which will be, if not the same, similar to the EDR. This claim is plausible based on both (i) the extended usage of the area under the ROC curve ($\mathcal{AUC}$, Fawcett (2006)) in unbalanced binary problems to assess the learnt classifiers, and (ii) the fact that the best class distribution to be used in the training dataset in order to maximise the $\mathcal{AUC}$ tends to be near the balanced scenario (Weiss and Provost, 2003). For that reason, in this section, we take a step further and study the main properties of this rule.

[7] This rule differs from the BDR used in the manuscript in the fact that our version of the BDR (Definition 2) assumes equal misclassification costs.

4.2.1 Competitiveness in the Class-Imbalance Scenario

First, we deal with the problem of determining whether the EDR is a competitive classifier to the class-imbalance domain. Opportunely, both the proposed influence function, eq. (7), and the fact that the EDR can be viewed as a particular case of the BDR can be used to shed some light on this issue. Regarding the latter, we focus on the following relationship between both decision rules:

Proposition 1 *Let $\mathbf{A}_\Psi(\boldsymbol{\eta}, \boldsymbol{\theta})$ be the true confusion matrix resulting from applying the algorithm Ψ to a classification problem γ_K with generative function equal to eq. (2) and with parameters $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$. Hence, when the generative model is known, for both the BDR ($\Psi = B$) and the EDR ($\Psi = E$), it holds true that*

$$\forall \boldsymbol{\eta}, \mathbf{A}_B(\boldsymbol{e}, \boldsymbol{\theta}) = \mathbf{A}_E(\boldsymbol{\eta}, \boldsymbol{\theta}). \tag{16}$$

This proposition[8] highlights the fact that, since the EDR is a particular case of the BDR (when the BDR is applied to the balanced version of γ_K), it results in a constant confusion matrix through the whole spectrum of the class distribution and its value is equal to the one resulting from the BDR in that particular case. Additionally, this relationship between both decision rules has an effect on the studied numerical performance scores:

Corollary 1 *Let $\mathcal{S}_\Psi(\boldsymbol{\eta}, \boldsymbol{\theta})$ be a numerical performance score summarising the true confusion matrix $\mathbf{A}_\Psi$. Then,*

$$\mathcal{S}_B(\boldsymbol{e}, \boldsymbol{\theta}) = \mathcal{S}_E(\boldsymbol{e}, \boldsymbol{\theta}) \tag{17}$$

is true in all circumstances.

Corollary 2 *When a numerical performance score $\mathbf{S}_\Psi$ summarises the true confusion matrix $\mathbf{A}_\Psi$ with independence[9] of the class distribution $\boldsymbol{\eta}$, the relation of Proposition 2 can also be held to be true for $\mathcal{S}_\Psi$. That is,*

$$\forall \boldsymbol{\eta}, \mathcal{S}_B(\boldsymbol{e}, \boldsymbol{\theta}) = \mathcal{S}_E(\boldsymbol{\eta}, \boldsymbol{\theta}). \tag{18}$$

Note that, in our framework, the numerical performance scores whose summary is independent to $\boldsymbol{\eta}$ are the local scores (recall and precision), the unweighted Hölder means among those local scores and the weighted Hölder means among those local scores whose weights are not calculated on the class distribution.

Corollary 3 *The value of classification accuracy, a score which is summarised using the distribution of the classes, for the BDR on the balanced version of a problem is equal to the value of the arithmetic mean among the recalls obtained by the EDR in γ_K showing any value of class imbalance. Formally,*

$$\forall \boldsymbol{\eta}, \mathcal{A}cc_B(\boldsymbol{e}, \boldsymbol{\theta}) = \mathcal{A}_E(\boldsymbol{\eta}, \boldsymbol{\theta}). \tag{19}$$

[8] Proofs for this proposition and its corollaries are not included in the manuscript due to their triviality; they can be easily inferred by using simple algebra from the proposed framework.

[9] No class probability is used in the calculation of the value of the score.

Now, by substituting eq. (18) in eq. (7); the influence function for the adequate performance scores can be rewritten as:

$$\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta}) = \int\limits_{\boldsymbol{\Theta}} [\mathcal{S}_E(\boldsymbol{\eta},\boldsymbol{\theta}) - \mathcal{S}_B(\boldsymbol{\eta},\boldsymbol{\theta})] d\boldsymbol{\theta}. \tag{20}$$

By means of this transformation, this new version of the influence function can be used to study whether the BDR or the EDR has, on average, a superior behaviour with regards to a determined adequate performance score ($\mathcal{A}$, $\mathcal{G}$, $\mathcal{H}$, and *min*). If the influence function is positive for the whole range of $\boldsymbol{\eta}$, then the EDR behaves, on average, better for that performance score than the BDR. The opposite case, a whole negative influence function, will show that the BDR is superior to the EDR. From just a re-examination of Figure 4 (4c-4f), where the influence function is studied in ternary problems and displayed for the performance scores, the following conclusion can be extracted: since $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ for the adequate scores is positive for "almost" the whole range of $\boldsymbol{\eta}$, *the EDR is, in general and on average, more competitive than the BDR under the assumed framework.* Therefore, the EDR tears apart the fundamental limit in the performance of every algorithm solution established by the BDR and claimed in (Drummond and Holte, 2005). Figure 4 serves as a counter example to generally prove the previous claim.

Finally, we want to remark that, although in theory there is no difference between dealing with binary or multi-class domains, in real world problems, most of the solutions seeking the EDR for unbalanced problems have been shown to be less effective or even to cause a negative effect in dealing with multiple classes (Wang and Yao, 2012; Sáez et al., 2016).

4.2.2 Optimality of the EDR

In order to determine the performance score optimised in the EDR, we also take advantage of the definition of the influence function and the relationships between both decision rules:

Theorem 1 *The equiprobable Bayes decision rule is optimal for $\mathcal{A}$, the unweighted Hölder mean among the recalls with $p = 1$.*

Proof Let the classification accuracy and the unweighted arithmetic mean among the recalls obtained by a classifier Ψ in a classification problem γ_K with a generative function determined by eq. (2) and with parameters $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ be defined as

$$\mathcal{A}cc_{\Psi}(\boldsymbol{\eta},\boldsymbol{\theta}) = \sum_{i=1}^{K} \eta_i \mathcal{R}_{\Psi}^{i}, \text{ and } \mathcal{A}_{\Psi}(\boldsymbol{\eta},\boldsymbol{\theta}) = \sum_{i=1}^{K} \frac{1}{K}\mathcal{R}_{\Psi}^{i}, \text{ respectively.}$$

Also let the BDR and the EDR be denoted as $\Psi = B$ and $\Psi = E$, respectively. Then, since the BDR obtains the optimal classification accuracy in every problem, it holds that:

$$\forall(\Psi,\boldsymbol{\eta},\boldsymbol{\theta}), \mathcal{A}cc_B(\boldsymbol{\eta},\boldsymbol{\theta}) = \max\{\mathcal{A}cc_{\Psi}(\boldsymbol{\eta},\boldsymbol{\theta})\}.$$

If the BDR is applied to the balanced version of the classification problem γ_K, i.e $\forall i, \eta_i = K^{-1}$, we obtain

$$\forall(\Psi,\boldsymbol{\eta},\boldsymbol{\theta}), \mathcal{A}cc_B(\boldsymbol{e},\boldsymbol{\theta}) = \max\{\sum_{i=1}^{K} \frac{1}{K}\mathcal{R}_{\Psi}^{i}\}.$$

Next, by the definition of $\mathcal{A}$, the max function can be rewritten as

$$\forall(\Psi, \boldsymbol{\eta}, \boldsymbol{\theta}), \mathcal{A}cc_B(\boldsymbol{e}, \boldsymbol{\theta}) = \max\{\sum_{i=1}^{K} \frac{1}{K}\mathcal{R}_{\Psi}^{i}\} = \max\{\mathcal{A}_{\Psi}(\boldsymbol{\eta}, \boldsymbol{\theta})\}.$$

Finally, by Corollary 3, we reach the conclusion that the EDR optimises the unweighted arithmetic mean among the recalls:

$$\forall(\Psi, \boldsymbol{\eta}, \boldsymbol{\theta}), \mathcal{A}_E(\boldsymbol{\eta}, \boldsymbol{\theta}) = \max\{\mathcal{A}_{\Psi}(\boldsymbol{\eta}, \boldsymbol{\theta})\}.$$

This theorem shows that the EDR achieves the lowest upper bound for $\mathcal{A}$, an adequate performance to the class imbalance extent, of any classifier. Therefore, most of the practical contributions to the class-imbalance scenario inherently also maximise this performance score. Unfortunately, this decision rule is not optimal for the other adequate performance scores ($\mathcal{G}$, $\mathcal{H}$ and min). This non-optimality can be straightforwardly proven using the influence function as expressed in eq. (20); both positive and negative values for $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$ will prove that neither the BDR nor the EDR always behave, on average, better than the other. Then, by just having a look at the previously indicated singularities of Figure 4d ($\mathcal{G}$), Figure 4e ($\mathcal{H}$) and Figure 4f (min), it can be seen that neither the EDR nor the BDR are optimal for these performance scores. For greater values of K than in Figure 4, the non-optimality holds, yet, the absolute values of these singularities are smaller. By having a glance at Figure 3, we cannot say whether the EDR is optimal for these performance scores in binary domains. However, by relaxing the geometry assumption[10] of equal unit variances in our model, we reach the same conclusion as in multi-class; *in general, for the unweighted Hölder means among the recalls with $p \neq 1$, the EDR is not an optimal decision rule.*

4.3 Discussion on Maximising Other Scores beyond $\mathcal{A}$

Several ideas regarding whether the remaining unweighted Hölder means among the recalls can be used to direct the definition of the forthcoming learning solutions for the class-imbalance domain can be extracted from the analyses performed in the previous sections:

(i) Provided that learning algorithms grouped by the EDR report positive results in the literature, and that the Hölder means with $p < 1$ have a higher sensitivity to the imbalance extent than $\mathcal{A}$, visibly, classifiers more adequate to deal with the class-imbalance related problems can be proposed[11]. In this range of p, (optimal) classifiers maximising $\mathcal{G}$, $\mathcal{H}$ and min, among other means, could be proposed. The optimal classifier for the latter will be the most restrictive that can be defined in this framework due to the fact that it ensures than the minimum recall of all classes must be the highest. Probably, a desired property (inferred from (Fernández et al., 2011)) within this complex domain.

[10] In Figure 3, the behaviour of the EDR is, in fact, optimal for any unweighted Hölder mean with $p \leq 1$. This is because the geometry created in the model has equal variances and the same distance between adjacent means and the Hölder mean equality; the overlapping area for each feature space is equal for both classes. Unfortunately, even in this geometric scenario, for $p > 1$, the EDR is not optimal.

[11] Since they have greater values for $\mathbb{I}_K^{\mathcal{S}}(\boldsymbol{\eta})$, a classifier maximising these scores will behave, on average, better than a classifier maximising $\mathcal{A}$.

(ii) On the contrary, the use of Hölder means with $p > 1$ in the definition of learning systems is not an adequate solution. These scores favour the greater recalls, one class will always have more chances to be selected in the classification. Moreover, this class rarely coincides with the minority class. In the extreme, we have the unweighted Hölder mean with $p = \infty$, whose optimal classifier is the one which classifies every instance to just one class. Therefore, this set of performance scores must be avoided to define loss functions in unbalanced domains, or at least, they must be used with special care.

In conclusion, any practical classifier Ψ resulting from the maximisation of a Hölder mean with $p \leq 1$ over a training sample is a better option than a classifier learnt in the traditional supervised framework due to the fact that it will favour the recalls of the minority classes. These classifiers will have less probability of, in cases of skewed class distributions, behaving like a dummy classifier. Therefore, *optimising adequate scores*[12] *to the class-imbalance extent may produce propitious classifiers reactive to the class-imbalance related problems.*

5 Third Study: Bounds for the Competitiveness of a Classifier in the Class-Imbalance Scenario

Virtually, every paper proposing a class-imbalance solution has the exact same experimental setup (Prati et al., 2015): "A proposed method is compared against one or two previously proposed methods over a dozen or so datasets. Although this experimental setup is reasonable to support an argument that the new method is as good as or better than the state of the art, it still leaves many unanswered questions". Among them, we can find the question of whether the proposed solution is able to produce competitive classifiers: if the precedent solutions are not competitive (in terms of our definition), the proposal might be uncompetitive as well. For that reason, in this study, we focus on the last question of the introduction: *Can bounds guaranteeing the competitiveness of a classifier be provided for the value of certain adequate performance scores?* In particular, since they have gained notorious importance throughout the paper, we focus on the values of the unweighted Hölder means among the recalls. We refer to this value as $\mathcal{S}_{\Psi}^{p}$, where p stands for the exponent of the mean used to assess a given classifier Ψ.

Next, in order to answer the question, we rewrite the definition of "competitiveness of a classifier" as *a competitive classifier is a classifier whose expected behaviour is superior to the expected behaviour of an already known baseline classifier*. Thus, by just instantiating the expressions "*expected behaviour*" and "*baseline classifier*", and after some algebra, practical bounds for $\mathcal{S}_{\Psi}^{p}$ can be given in order to ensure the adequateness of Ψ for any given unbalanced problem. Regarding the former concept, the expected behaviour cannot be determined by the direct use of the inherently maximised performance score as it would not be legitimate; Ψ might be favoured in the comparison. For that reason, we rely on common sense and on previous experience in the class-imbalance domain to define the term. Within this domain, it is interesting to obtain classifiers achieving great recalls for the minority

[12] The interested reader can find, in the appendix, an example of how the decision regions for the different decision rules optimising the studied global performance scores are located in a ternary problem generated from a Gaussian mixture model.

classes while maintaining adequate recalls for the majority ones (Gu et al., 2009), a fairly complicated task. Therefore, the competitiveness of the target classifier can be translated into obtaining greater or equal recalls for both the minority and majority classes than a baseline classifier for which prior knowledge is available. Finally, concerning the second term, we make use of the most-utilised base classifier for establishing the lower expected behaviour than a competitive classifier must obtain; the classifier representing the random guessing. Formally:

Definition 5 The classifier representing the random guessing of K different classes is known as the uniformly random classifier (RAND) and it is given by

$$\hat{c}_R = \text{Unif}\{1, K\}. \tag{21}$$

where Unif$\{1, K\}$ is the discrete uniformly random function which assigns a categorical class c_i to an example $\mathbf{x}$ with probability $1/K$. All the performance scores contemplated in Table 1 assign the same value to the goodness of this classifier:

$$\mathcal{P}^i = \mathcal{R}^i = \tfrac{1}{K}, \forall i, \text{ and } \mathcal{A}cc = \mathcal{A} = \mathcal{G} = \mathcal{H} = max = min = \frac{1}{K}.$$

In general, for the Hölder means among the recalls[13] (independently of the values of weights, $\boldsymbol{\zeta}$, and the exponent p), it holds that

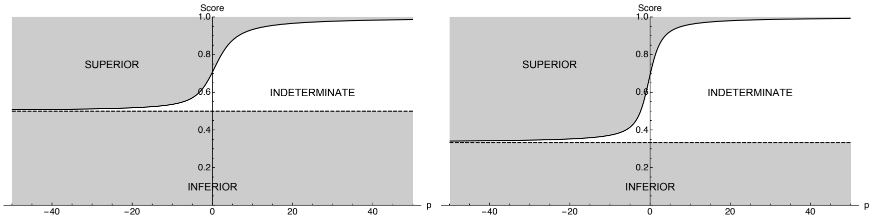
$$M_p(\mathbf{R}, \boldsymbol{\zeta}) = \frac{1}{K}, \text{ where } \mathbf{R} = (\mathcal{R}^1, \ldots, \mathcal{R}^K).$$

Now, by means of the previous instantiations, we can determine the *competitiveness* of Ψ by just checking which values for an unweighted Hölder mean ensure that a recall of at least $1/K$ is obtained for all classes. Opportunely, this calculation can be easily performed in this kind of functions. Then, let S^p_{sup} be defined as the lowest value for $\mathcal{S}^p_\Psi$ ensuring that the classifier Ψ is certainly superior to RAND. This extreme situation takes place when just one recall is equal to $1/K$ and the remaining are all equal to 1, i.e. $\exists! j(\mathcal{R}^j = 1/K \wedge \forall i \neq j, \mathcal{R}^i = 1)$. In this scenario, the slightest negative variation in the score could result in an incompetent classifier, i.e. a classifier Ψ reporting a score of $\mathcal{S}^p_\Psi = S^p_{\text{sup}} - \epsilon$, where $\epsilon \to 0^+$, could indicate that $\exists i, \mathcal{R}^i < 1/K$. On the contrary, a score of $S^p_{\text{sup}} + \epsilon$ will always indicate that Ψ is a competitive classifier with $\forall i, \mathcal{R}^i > 1/K$. Thus, by just substituting these recalls in the definition of a Hölder means – eq. (1) –, we obtain

$$S^p_{\text{sup}} = \quad \left(K^p + K^{(p-1)} + 1\right)^{\frac{1}{p}}. \tag{22}$$

Regarding the opposite scenario, the *incompetence* is determined by calculating which values of the Hölder means ensure that at least one recall is below the random guessing value, $1/K$. Here, let S^p_{inf} be the strictly upper value for the score $\mathcal{S}^p_\Psi$ indicating that Ψ is not competitive, i.e. it is inevitably inferior to RAND. For this case, we choose the scenario of obtaining a classifier behaving in the same manner as the RAND. Therefore, any negative variation in the score of Ψ will indicate the incompetence of it. Analogously, any positive variation might indicate

[13] This can be trivially proved by following a similar reasoning to Theorem 1 of (Ortigosa-Hernández et al., 2016) but using the Hölder mean equality instead.



(a) Binary classifier

(b) Ternary classifier

Fig. 6: Limiting values for the unweighted Hölder means ensuring that a given classifier is superior/inferior to the random guessing.

that the classifier is competent. In the latter assertion we cannot remove the modal verb 'might' due to the fact that a higher score, but less than S^p_{sup}, is not sufficient for safeguarding better recalls than the RAND for all classes. Hence, the value for this upper value is:

$$S^p_{\text{inf}} = \quad \frac{1}{K}\ [= \mathsf{RAND}] \tag{23}$$

As a summary of the previous paragraphs, we plot Figure 6. Here, the values for both eq. (22), S^p_{sup}, and eq. (23), S^p_{inf}, (y-axis) in the domain $p \in [-50, 50]$ (x-axis) are presented. Figure 6a represents these values for binary problems and Figure 6b for ternary cases. In the figures, three different areas can be seen. They correspond to the cases for $\mathcal{S}^p_{\Psi}$ argued in this section:

if $\mathcal{S}^p_{\Psi} \in [S^p_{\text{sup}}, 1]$: SUPERIOR.
Ψ is a competitive classifier for the problem
if $\mathcal{S}^p_{\Psi} \in [0, S^p_{\text{inf}})$: INFERIOR.
Ψ is incompetent to solve the problem
if $\mathcal{S}^p_{\Psi} \in [S^p_{\text{inf}}, S^p_{\text{sup}})$: INDETERMINATE.
The competitiveness of Ψ cannot be checked with just $\mathcal{S}^p_{\Psi}$

In conclusion, it can be seen that the thesis highlighted throughout the whole paper is supported once more; the adequateness of the unweighted Hölder mean among the recalls with $p \leq 1$ to assess the goodness of a classifier in multi-class class-imbalance domains. Not only do they focus on the behaviour of all recalls, independently of their class probability values, but also the resulting score value is acutely informative in terms of competitiveness of the classifier. Here again, *min* is the most restrictive and the most informative score: a value above $1/K$ indicates that the classifier is competitive. A value below denotes the opposite.

6 Limitations of the Studies and Suggestions for Future Research

One limitation with the research described in this article is that, because all the conclusions are extracted from the assumed manageable model used to draw the influence functions, our conclusions may hold only for the assumed model. However, the generality of the results holds true for almost all the contributions of the paper as we indicate throughout the sections. Only for the case of our claim saying that, within our assumed model, averaging precisions is inadequate to validate unbalanced domains, the general validity cannot be assured. This, along with the high number of papers using precision to assess classifiers in the class-imbalance literature, sustains a potential theoretical research line in order to determine whether the usage of precision is adequate when facing skewed class distributions. Other potential research lines regarding our assumed model for the class-imbalance problem could be the following: Firstly, several other classifiers rather than the BDR can be used as a representative classifier in the study. Secondly, analytical solutions can be found for either the influence function for each score or for the decision rules optimising Hölder means with $p < 1$. Thirdly, here, we exclusively deal with the supervised learning framework, however, other types of learning scenarios such as semi-supervised learning (Ortigosa-Hernández et al., 2016) can be studied.

Another limitation of our paper is that, in the second study, where the state-of-the-art methodological contributions to this intricate problem are reviewed, the scope is limited. There, we only investigate the data sampling and the cost-sensitive learning approaches. As pointed out in Section 4, since our aim is to study the multi-class setting, where the effectiveness of the non-dominant (other than data sampling and cost-sensitive learning) is still unknown, we narrow our scope to just the approaches which can be used in the multi-class setting and which never behave as blackboxes when studied using the Bayesian decision theory (Wang and Yao, 2012; Rodríguez et al., 2014). However, there are reasons to believe that our conclusions will hold for other binary approaches as long as they seek a maximisation of the $\mathcal{AUC}$ (Weiss and Provost, 2003). We believe that an important potential future line of research, before expanding our scope in this study, is to determine which approaches proposed for binary unbalanced problems are effective in the multi-class setting.

Finally, the fact that the number of performance scores studied is limited may also be seen as a restriction of this study; we deal only with the most utilised numerical performance scores. Since there is a huge number of performance scores (Gu et al., 2009), future research lines can make use of our framework to investigate their adequateness to unbalanced domains; similar studies can be proposed using other interesting kinds of scores[14]; graphical performance scores (Santafe et al., 2015), adjusted (to the class-imbalance domain) performance scores (López et al., 2013), or, even, the whole confusion matrix (Koço and Capponi, 2013).

7 Summary

Most of the existing learning algorithms are designed to asymptotically converge to the BDR by minimising the 0-1 loss or, what is equivalent, maximising the classification accuracy. Unfortunately, when dealing with unbalanced problems, the classification accuracy is not an adequate performance score due to the fact that the underrepresented classes have very little impact on the measure when they are compared to the overrepresented classes. Therefore, it is imperative to define more adequate learning systems which can effectively deal with skewed class distributions in the multi-class setting. Thus, in order to shorten the distance towards the

[14] Interested readers can find, in the appendix, a preliminary study on another two broadly used scores for binary unbalanced problems; the $\mathcal{F}_1$-score and the $\mathcal{AUC}$ for discrete classifiers.

previous ideal, in this paper, an exhaustive analysis of a set of numerical performance scores is carried out, not only to be able to determine their adequateness to assess the goodness of a classifier in this complex scenario, but also to be capable of studying whether they are suited to be used to produce more competitive learning systems for unbalanced problems. Specifically, we focus on performing an exhaustive analysis of the most-common numerical performance scores used in the class-imbalance domain which can also be represented as Hölder means (Bullen, 2003) among the recalls of all the classes. This set groups well-known performance scores such as accuracy, a-mean, g-mean etc. Since many interdependent factors may hinder the discerning skill of the classifier and, therefore, vary the value of the score, we develop a novel classification framework which allows us to marginalise out the class-imbalance component from the rest of the factors. As a result, the influence of the class-imbalance extent on the performance score using the long studied BDR as a classifier can be measured in isolation. With this study, we provide answers to the following questions:

Which performance scores are adequate to determine the competitiveness of a classifier in multi-class unbalanced domains? *The performance scores which are unweighted Hölder means with* $p \leq 1$ *(a-mean, g-mean, h-mean, etc.) among the recalls are the most appropriate to evaluate the competitiveness of classifiers in multi-class unbalanced problems. In these cases, misclassifying the least probable classes is highly penalised.*

Which performance scores are maximised in the most well-known learning solutions designed to deal with skewed classes? *Data sampling and cost-sensitive learning techniques are designed to maximise the a-mean (which is the unweighted Hölder mean with* $p = 1$*) due to the fact that they asymptotically converge to the BDR for equiprobable classes. Other solutions to the class-imbalance problem also seem to similarly behave owing to the fact that they are designed to maximise the* $\mathcal{AUC}$*.*

Can bounds guaranteeing the competitiveness of a classifier be provided for certain adequate performance scores? *Yes, we finalise the paper by providing two different practical bounds for the performance scores expressed as unweighted Hölder means among the recalls with* $p \in \mathbb{R} \cup \{+\infty, -\infty\}$*; a bound for the lowest value of the performance score ensuring a competitive solution for unbalanced problems and a bound for the highest value of the score indicating an incompetent solution.*

Acknowledgments

This work is partially supported by the Basque Government (IT609-13 and Elkartek BID3A) and the Spanish Ministry of Economy and Competitiveness (TIN2013-41272P). Moreover, Jonathan Ortigosa-Hernández is partly financed by the Ministry of Science and Innovation (MEC-FPU AP2008-00766) and Jose A. Lozano by both the Basque Government (BERC program 2014-2017) and the Spanish Ministry of Economy and Competitiveness (Severo Ochoa Program SEV-2013-0323).

References

Angluin, D. and Laird, P. (1988). Learning from noisy examples. *Machine Learning*, 2:343–370.

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. on Information and Systems Security*, 3(3):186–205.

Bartlett, P., Jordan, M., and McAuliffe, J. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *Journal of the American Statistical Association*, 46(6):2102–2117.

Basu, M. and Ho, T. (2006). *Data Complexity in Pattern Recognition*. New York City, NY: Springer-Verlag.

Batista, G., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter*, 6(1):20–29.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. New York City, NY: Springer.

Bradley, A. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):145–1159.

Branco, P., Torgo, L., and Ribeiro, R. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49(2):31:1–31:50.

Bullen, P. S. (2003). The power means. In *Handbook of Means and Their Inequalities*. Dordrecht, Netherlands: Springer.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Denil, M. and Trappenberg, T. (2010). Overlap versus imbalance. In *Proc. of the 23rd Canadian Conf. on Advances in Artificial Intelligence (CANADA AI 2010)*, pages 220–231.

Di Martino, M., Martínez, A., Iturralde, P., and Lecumberry, F. (2013). Novel classifier scheme for imbalanced problems. *Pattern Recognition Letters*, 34(10):1146–1151.

Drummond, C. and Holte, R. (2005). Severe class imbalance: Why better algorithms aren't the answer. In *Proc. of the 16th European Conf. on Machine Learning*, pages 539–546.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *In Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI2001)*, pages 973–978.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874.

Fernández, A., García, S., and Herrera, F. (2011). Addressing the classification with imbalanced data: Open problems and new challenges on class distribution. In *Proc. of the 6th Int. Conf. on Hybrid Artificial Intelligent Systems - Part I*, pages 1–10.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., and Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 42(4):463–484.

Gu, Q., Zhu, L., and Cai, Z. (2009). Evaluation measures of the classification performance of imbalanced data sets. In *Proc. of the 4th Int. Symp. on Advances in Computation and Intelligence (ISICA2009)*, pages 461–471.

Hand, D. and Till, R. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186.

Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE Trans. on Information Theory*, 14:515–516.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. on Knowledge and Data Engineering*, 21(9):1263–1284.

Ho, T. K. and Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transactions om Pattern Analysis and Machine Intelligence*, 24(3):289–300.

Holte, R. C., Acker, L. E., and Porter, B. W. (1989). Concept learning and the problem of small disjuncts. In *Proc. of the 11th Int. Joint Conf. on Artificial Intelligence (IJCAI'89)*, pages 813–818.

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

Kalayeh, H. and Landgrebe, D. (1983). Predicting the required number of training samples. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5:664–667.

Kanamori, T., Takeda, A., and Suzuki, T. (2013). Conjugate relation between loss functions and uncertainty sets in classification problems. *Journal of Machine Learning Research*, 14:1461–1504.

Koço, S. and Capponi, C. (2013). On multi-class classification through the minimization of the confusion matrix norm. In *Proc. of the 5th Asian Conf. on Machine Learning (ACML2013)*, pages 277–292.

Krishnapuram, B., Yu, S., and Rao, R. B. (2011). *Cost-Sensitive Machine Learning*. Boca Raton, FL: CRC Press Inc., 1st edition.

Kubat, M., Holte, R., and Matwin, S. (1997). Learning when negative examples abound. In *Proc. of the 9th European Conf. on Machine Learning (ECML1997)*, pages 146–153.

Liu, X.-Y. and Zhou, Z.-H. (2006). The influence of class imbalance on cost-sensitive learning: An empirical study. In *Proc. of the 6th Int. Conf. on Data Mining (ICDM'06)*, pages 970–974.

López, V., Fernández, A., García, S., Palade, V., and Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141.

Matthews, R. (1996). Base-rate errors and rain forecast. *Nature*, 382(6594):766.

Matthews, R. (1997). Decision-theoretic limits on earthquake prediction. *Geophysical Journal International*, 131(3):526–529.

Menon, A., Narasimhan, H., Agarwal, S., and Chawla, S. (2013). On the statistical consistency of algorithms for binary classification under class imbalance. In *Proc. of the 30th Int. Conf. on Machine Learning (ICML2013)*, pages 603–611.

Ortigosa-Hernández, J., Inza, I., and Lozano, J. A. (2016). Semi-supervised multi-class classification problems with scarcity of labelled data: A theoretical study. *IEEE Trans. on Neural Networks and Learning Systems*, 27(12):2602–2614.

Ortigosa-Hernández, J., Inza, I., and Lozano, J. A. (2017). Measuring the class-imbalance extent of multi-class problems. *Pattern Recognition Letters*, 98(15):32–38.

Prati, R., Batista, G., and Monard, M. (2004). Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Proc. of the 3rd Mexican Int. Conf. on Artificial Intelligence (MICAI2004)*, volume 2972, pages 312–321.

Prati, R. C., Batista, G. E., and Silva, D. F. (2015). Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1):247–270.

Provost, F. and Domingos, P. (2000). Well-trained pets: Improving probability estimation trees. Technical Report IS-00-04, Stern School of Business, New York University.

Provost, F., Fawcett, T., and Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *Proc. of the 15th Int. Conf. on Machine Learning (ICML1998)*, pages 445–453.

Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

Rodríguez, D., Herraiz, I., Harrison, R., Dolado, J., and Riquelme, J. C. (2014). Preliminary comparison of techniques for dealing with imbalance in software defect prediction. In *Proc. of the 18th Int. Conf. on Evaluation and Assessment in Software Engineering (EASE2014)*, pages 43:1–43:10.

Sáez, J., Krawczyk, B., and Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178.

Santafe, G., Inza, I., and Lozano, J. (2015). Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review*, 44(4):467–508.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., and Krasser, S. (2009). SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288.

Thai-Nghe, N., Gantner, Z., and Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. In *Proc. of IEEE Int. Joint Conf. on Neural Networks (IJCNN2010)*, pages 1–14.

Tomek, I. (1976). Two modifications of CNN. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-6:769–772.

Wang, S. and Yao, X. (2012). Multi-class imbalance problems: Analysis and potential solutions. *IEEE Trans. on Systems, Man and Cybernetics - Part B*, 42(4):1119–1130.

Weiss, G. (2004). Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19.

Weiss, G. and Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19(1):315–354.

Weiss, G. M. (2013). Foundations of imbalanced learning. In He, H. and Ma, Y., editors, *Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ: Wiley-IEEE Press.

Wolfram Research Inc. (2014). Mathematica 10.4.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37.

Young, T. Y. and Calvert, T. W. (1974). *Classification, Estimation, and Pattern Recognition*. Oxford, UK: Elsevier Science Ltd.

Zhou, Z.-H. and Lui, X. (2010). On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257.

A Influence Functions for Other Predominant Performance Scores Used in Binary Unbalanced Problems

Although the main aim of this manuscript is to propose a set of adequate performance scores for the scarcely studied unbalanced multi-class setting where, to the best of our knowledge, there is no convention on how to assess the learnt classifiers, the current practice in the literature for binary unbalanced problems, e.g. (Axelsson, 2000; Tang et al., 2009; Denil and Trappenberg, 2010; Di Martino et al., 2013; López et al., 2013; Prati et al., 2015), also uses the $\mathcal{AUC}$ (Bradley, 1997; Provost et al., 1998) and the $\mathcal{F}$-score (Rijsbergen, 1979) to assess the performance of the proposed solutions.

In this context, this appendix is devoted to performing a preliminary study on those broadly used performance scores by means of the framework presented in this manuscript. The objetives of this study are the following: (i) Discovering the existing relationships between our proposals and the state-of-the-art performance scores for binary problems, and, in the case where they are not as recommendable as the proposals of the paper, (ii) providing a justification on why these scores for binary problems are not as recommendable as our proposals for the multi-class scenario.

A.1 Area Under the $\mathcal{ROC}$ Curve ($\mathcal{AUC}$)

In the binary setting, the Receiver Operating Curve ($\mathcal{ROC}$) curve is a graphical score which allows the visualisation of the trade-off between the proportion of positives that are correctly identified and the number of negative events wrongly categorised as positive, evidencing that any classifier cannot increase the number of true positives without also increasing the false positives (Provost et al., 1998; López et al., 2013). The area under the $\mathcal{ROC}$ curve ($\mathcal{AUC}$) provides a single measure of a classifier's performance (Bradley, 1997), see Figure 7. Since there are classifiers for which only one point of the curve is available (Fawcett, 2006), the area under the $\mathcal{ROC}$ curve ($\mathcal{AUC}$) is frequently approximated in the class-imbalance literature, e.g. (Galar et al., 2012; López et al., 2013), by means of the following summary statistic, the arithmetic mean of the true positive rate and the complement of the false positive rate:

$$\mathcal{AUC} = \frac{1 + \mathcal{TP}_{rate} - \mathcal{FP}_{rate}}{2}. \tag{24}$$

There, $\mathcal{TP}_{rate}$ measures the proportion of positives that are correctly identified and coincides with the definition of the recall. $\mathcal{FP}_{rate}$, on the contrary, is the ratio between the number of negative events wrongly categorised as positive and the total number of actual negative events, which coincides with the complementary of the recall of the other class. This formula calculates the area between one selected operating point ($\mathcal{TP}_{rate}, \mathcal{FP}_{rate}$) and the triangular ROC curve constructed with only $(0, 0)$ and $(1, 1)$ as perceived in Figure 7. Formally and assuming that the positive class corresponds to first class, we have that

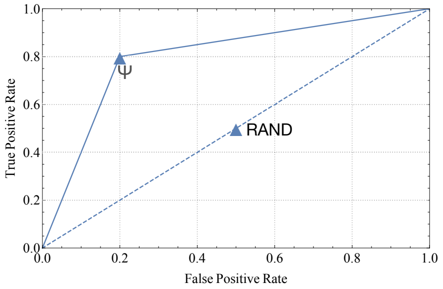


Fig. 7: Example of a ROC plot. Two curves for two classifiers are plotted: the dashed line represents the random classifier (RAND), whilst the solid line is a classifier Ψ behaving better than the random classifiers.

$$\mathcal{TP}_{rate} = \frac{a_{1,1}}{a_{1,1} + a_{2,1}} = \mathcal{R}^1, \text{ and } \mathcal{FP}_{rate} = \frac{a_{1,2}}{a_{1,2} + a_{2,2}} = 1 - \frac{a_{2,2}}{a_{1,2} + a_{2,2}} = 1 - \mathcal{R}^2. \quad (25)$$

Then, by substituting eq. (25) in eq. (24) and after simple algebra we reach the conclusion that using the $\mathcal{AUC}$ to assess classifiers producing just one point of the curve is, for binary classification problems, equivalent to using the arithmetic mean between the recalls, $\mathcal{A}$, as claimed in (Santafe et al., 2015):

$$\mathcal{AUC} = \frac{1}{2}(\mathcal{R}^1 + \mathcal{R}^2) = \mathcal{A} \quad (26)$$

These calculations manifest that all the conclusions made for $\mathcal{A}$ in binary problems holds for the $\mathcal{AUC}$ when only one point of the ROC curve is available, including the fact that the EDR is also an optimal decision rule.

Unfortunately, the $\mathcal{AUC}$, unlike the proposed Hölder means among the recalls, is a measure of the discriminability of a pair of classes (Fawcett, 2006): “In a binary problem, the $\mathcal{AUC}$ is a single scalar value, but a multi-class problem introduces the issue of combining multiple pairwise discriminability values. There are several approaches to perform this combination, e.g. (Provost and Domingos, 2000) or (Hand and Till, 2001)”. In that paper, both advantages and disadvantages of the usage of each approach are given. For that reason, we believe that averaging recalls is more recomendable than $\mathcal{AUC}$ in the multi-class setting due to the following: (i) Hölder means among the recalls possess a natural and trivial expansion to multiple classes, (ii) their computation is straightforward, and (iii) there exist bounds ensuring competitive classifiers for these scores. Yet, we believe that this brief preliminary study of the $\mathcal{AUC}$ justifies the opening of a potential theoretical research line to answer questions such as the following:

1. Is there a similarity to $\mathcal{A}$ when more than one curve point is available for a classifier? Is the EDR also an optimal decision rule for that case? Would the EDR remain so having any number K of class variables?
2. While $\mathcal{A}$ has a natural and trivial expansion to the multiple categorical classes, $\mathcal{AUC}$ introduces the issue of deciding which approach should be used to combine multiple pairwise discriminability values. Would $\mathcal{A}$ be a more adequate performance score than $\mathcal{AUC}$ for assessing multi-class unbalanced problems?
3. This similarity supports the empirical conclusions extracted by (Weiss and Provost, 2003) claiming that balanced class distributions in training datasets maximise the $\mathcal{AUC}$. Could their claim be theoretically proved by means of our framework?

A.2 The $\mathcal{F}$-score

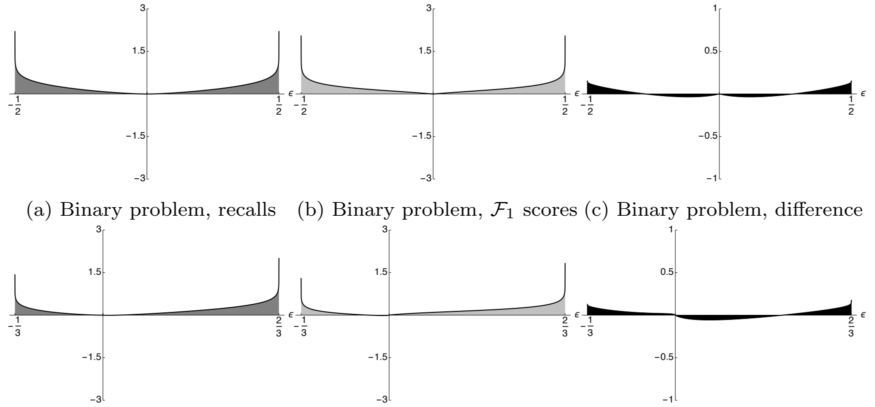
The $\mathcal{F}$-score was introduced in information retrieval applications domains but, since then it is also widely used in machine learning in general (Rijsbergen, 1979). The definition of this score depends on a parameter β provided by the practitioner in order to weight the terms of precision $\mathcal{P}$ and recall $\mathcal{R}$:

$$\mathcal{F}^i_\beta = \frac{(\beta^2 + 1)\mathcal{R}^i\mathcal{P}^i}{\beta^2\mathcal{R}^i + \mathcal{P}^i} \quad (27)$$

When $\beta = 1$, the obtained F-score is known as $\mathcal{F}_1$-score and corresponds to the harmonic mean (unweighted Hölder mean with $p = -1$) of precision and recall. This value is usually the one used in the class-imbalance literature (Tang et al., 2009; Denil and Trappenberg, 2010; Di Martino et al., 2013). Since the $\mathcal{F}_1$-score is a local performance score depending on just the precision and recall of a single class, its usage in the multi-class setting and, thus, in the proposed framework is straightforward: let the $\mathcal{F}^i_1$-score be defined as

$$\mathcal{F}^i_1 = \frac{2\mathcal{R}^i\mathcal{P}^i}{\mathcal{R}^i + \mathcal{P}^i}, \quad (28)$$

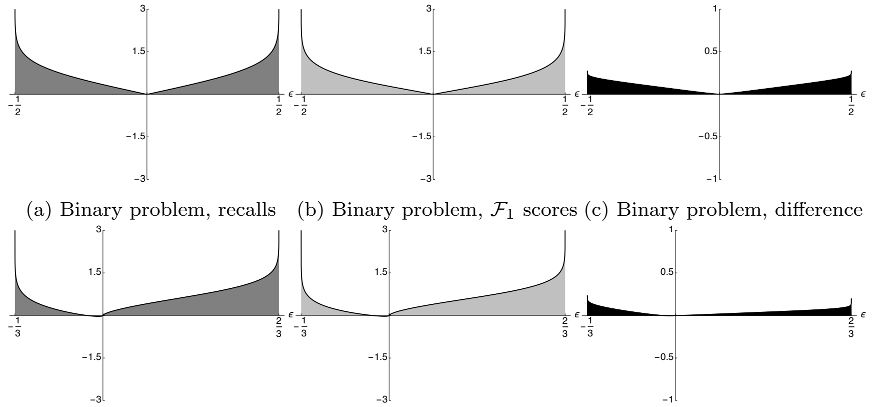
so that we can determine whether averaging $\mathcal{F}^i_1$-scores produces adequate global performance scores for the multi-class unbalanced scenario.



(a) Binary problem, recalls (b) Binary problem, $\mathcal{F}_1$ scores (c) Binary problem, difference

(d) Ternary problem, recalls (e) Ternary problem, $\mathcal{F}_1$ scores(f) Ternary problem, difference

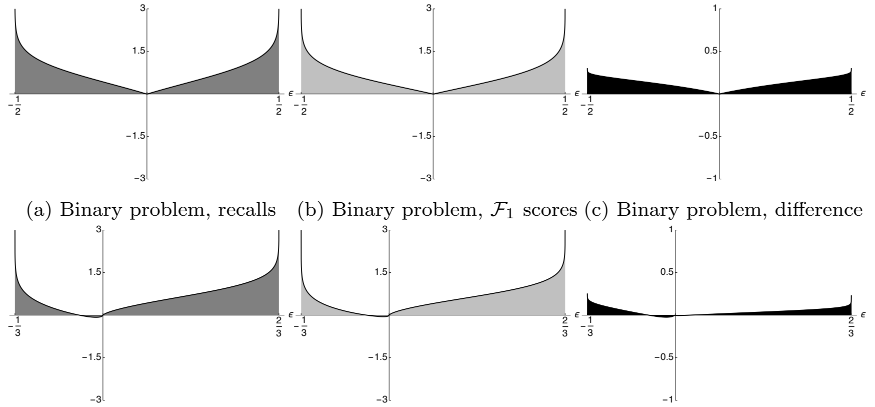
Fig. 8: The influence function in binary and ternary problems, $\mathbb{I}_K^{\mathcal{S}}(\epsilon)$, for both the **arithmetic mean** among the recalls and among the $\mathcal{F}_1$-scores throughout the range $-\frac{1}{K} \leq \epsilon \leq \frac{K-1}{K}$.



(a) Binary problem, recalls (b) Binary problem, $\mathcal{F}_1$ scores (c) Binary problem, difference

(d) Ternary problem, recalls (e) Ternary problem, $\mathcal{F}_1$ scores(f) Ternary problem, difference

Fig. 9: The influence function in binary and ternary problems, $\mathbb{I}_K^{\mathcal{S}}(\epsilon)$, for both the **geometric mean** among the recalls and among the $\mathcal{F}_1$-scores throughout the range $-\frac{1}{K} \leq \epsilon \leq \frac{K-1}{K}$.

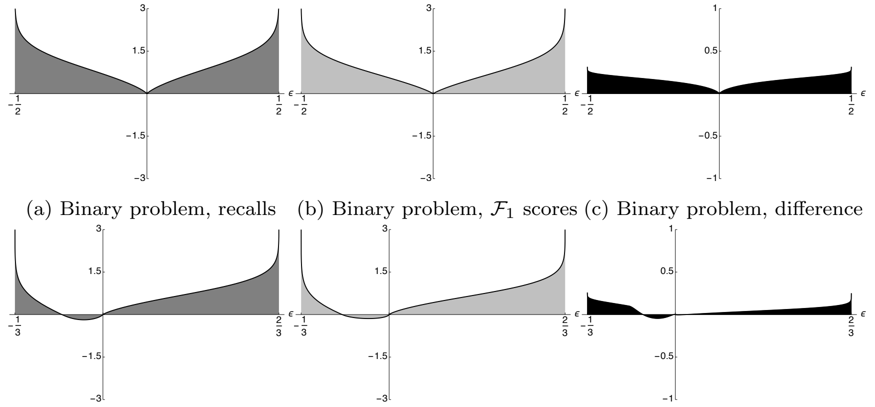


(a) Binary problem, recalls (b) Binary problem, $\mathcal{F}_1$ scores (c) Binary problem, difference

(d) Ternary problem, recalls (e) Ternary problem, $\mathcal{F}_1$ scores(f) Ternary problem, difference

Fig. 10: The influence function in binary and ternary problems, $\mathbb{I}^{\mathcal{S}}_{K}(\epsilon)$, for both the **harmonic mean** among the recalls and among the $\mathcal{F}_1$-scores throughout the range $-\frac{1}{K} \leq \epsilon \leq \frac{K-1}{K}$.

In order to do that, we plot the influence functions $\mathbb{I}^{\mathcal{S}}_{K}(\epsilon)$ for the averages among the recalls used in the manuscript, the influence functions for the averages among the $\mathcal{F}^i_1$-scores, and, afterwards, compare them (by computing the difference). Figures 8-11 share the same structure; the first column is the influence function of a Hölder mean among the recalls, the second column is the influence function of the same Hölder mean, but among the $\mathcal{F}_1$-scores, and third column corresponds to the subtraction of the influence function, using the average of recalls as minuend. Then, the first row corresponds the binary setting and the second row deals with a ternary problem. The Hölder means presented are the following: Figure 8 is the arithmetic mean, Figure 9 is the geometric mean, Figure 10 is the harmonic mean, and 11 is minimum value.



(a) Binary problem, recalls (b) Binary problem, $\mathcal{F}_1$ scores (c) Binary problem, difference

(d) Ternary problem, recalls (e) Ternary problem, $\mathcal{F}_1$ scores(f) Ternary problem, difference

Fig. 11: The influence function in binary and ternary problems, $\mathbb{I}^{\mathcal{S}}_{K}(\epsilon)$, for both the **minimum** recall and the **minimum** $\mathcal{F}_1$-score throughout the range $-\frac{1}{K} \leq \epsilon \leq \frac{K-1}{K}$.

Examining the results, we conclude that using averages among the $\mathcal{F}_1$-scores to assess the performance of multi-class unbalanced problems is also an adequate practise. In all the figures, both sets of performance scores behave similarly. Yet, from the subtraction, it can be inferred that averages among the $\mathcal{F}_1$-scores seems to be slightly less expressive than averages among the recalls.

B Case Study: How are the Decision Regions for the Classifiers Optimising the Studied Scores Placed in a Ternary Problem?

With the purpose of exposing the asymptotical behaviour of the classifiers maximising the numerical performance scores studied in the manuscript, here, we set up a controlled example showing how the optimal classifiers for those scores split the feature space into different decision regions. As we assume the generative model to be known, here, each optimal classifier will be referred to as the decision rule optimising a certain numerical performance score.

Deliberately, we define the example in terms of the framework of the manuscript: Let the example γ_3 be a ternary classification problem with a generative model composed of a univariate Gaussian mixture model represented by the following joint probability density function

$$f(\mathbf{x}, c|\boldsymbol{\theta}) = \sum_{i=1}^{K} \eta_i f(\mathbf{x}|c_i, \boldsymbol{\theta}_i) \mathbb{1}(c = c_i). \tag{29}$$

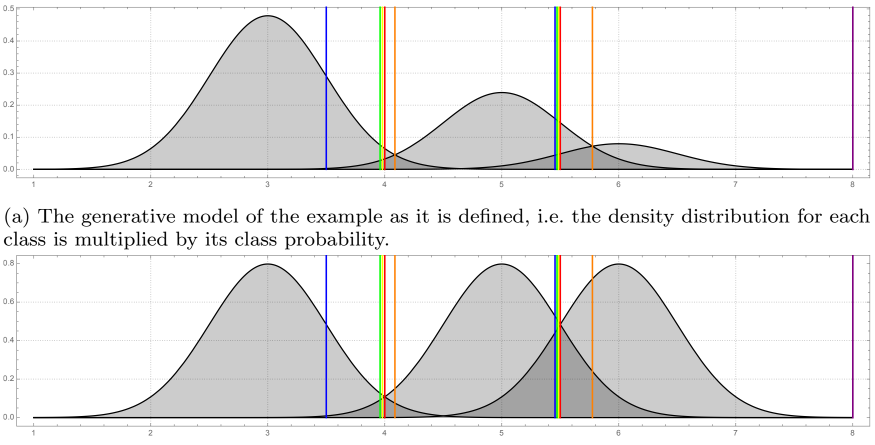
Note that the previous equation uses f instead of ρ (as in the manuscript) for the distribution of the feature space. This is due to the fact that, here, we assume the features to be continuous. Let $\mathbb{1}(c = c_i)$ stand for the indicator function, i.e. it is equal to 1 if $c = c_i$ and 0 otherwise. Then, let us instantiate each mixture component in this example as a univariate Gaussian distribution with parameters

$$f(x|c_1, \boldsymbol{\theta}_1) \sim N(3, 0.5) \qquad f(x|c_2, \boldsymbol{\theta}_2) \sim N(5, 0.5) \qquad f(x|c_3, \boldsymbol{\theta}_3) \sim N(6, 0.5)$$

and let the class distribution be set as $\boldsymbol{\eta} = (0.6, 0.3, 0.1)$. Figure 12 shows the generative model of the proposed example. Whilst Figure 12a plots the density distribution for each class multiplied by its class probability ($\eta_i f(\mathbf{x}|c_i, \boldsymbol{\theta}_i)$), Figure 12b presents only the mixture density distribution of the classes ($f(\mathbf{x}|c_i, \boldsymbol{\theta}_i)$). The main objective of presenting two views of the same model is two-fold; (i) the influence of the class distribution on the intricacy of the generative model can be perceived at a glance, and (ii) each view is a key factor in showing the behaviour of the two main decision rules studied in the manuscript; the BDR (which optimises $\mathcal{A}cc$) and the EDR (which is optimal for $\mathcal{A}$).

Moreover, the figures also display the limits of the decision rules optimising all the numerical performance scores studied in the paper. Here, the decision regions are represented by vertical lines dividing the real number line; the first line to the left divides Ω_1 and Ω_2, and the second line separates Ω_2 and Ω_3, i.e. $\Omega_1 = (-\infty, \text{left line}]$, $\Omega_2 = (\text{left line}, \text{right line}]$, and $\Omega_3 = (\text{right line}, \infty)$. In order to calculate these regions, we directly applied the BDR and the EDR to find their optimal decision regions for $\mathcal{A}cc$ and $\mathcal{A}$, respectively, and an exhaustive search to find the optimal classifiers and their decision regions for the performance scores which have an unknown (to us) decision rule; $\mathcal{G}$, $\mathcal{H}$, max, and min. Since these unknown decision rules have not been named in the manuscript, henceforth, we will refer to the unnamed decision rule optimising a score S as S-DR, e.g. $\mathcal{G}$-DR will stand for the decision rule optimising $\mathcal{G}$. In the figures, the limiting values are coloured as follows: the red lines represent the limits for the EDR, the orange is for the BDR, yellow is used for $\mathcal{G}$-DR, green for $\mathcal{H}$-DR, and the blue and purple colours are for the extreme rules; min-DR and max-DR, respectively.

By comparing both figures, it can be easily seen that the BDR uses both the class distribution and the feature distribution to split the real number line into regions, and that the EDR relies just on the feature distribution to accomplish the same task. While the BDR cuts the feature space in the intersection points of Figure 12a, the EDR relies on the intersections of Figure 12b. Next, note that the insensitive max-DR has only one limiting value (we set this value at $+\infty$, i.e. $\Omega_1 = (-\infty, \infty)$ and $\Omega_2 = \Omega_3 = \emptyset$). This is because the optimality can be easily achieved with the dummy classifier which classifies all instances as just one class. Therefore, an optimal classifier for that rule will be the one that assigns the whole real number line to a determined decision region and that leaves the rest of the decision region empty. Then,



(a) The generative model of the example as it is defined, i.e. the density distribution for each class is multiplied by its class probability.

(b) The density distribution of each categorical class.

Fig. 12: The decision region limits for the decision rules optimising the studied scores: EDR (red), BDR (orange), $\mathcal{G}$-DR (yellow), $\mathcal{H}$-DR (green), min-DR (blue) and max-DR (purple).

regarding the min-DR, it can be seen that it seeks that the area of each class in its decision region is equal to any area of any other class in its decision region so that the minimum recall will be the maximum, i.e.

$$\forall 1 \leq i, j \leq 3, \int\limits_{\Omega_i} f(\mathbf{x}|c_i, \boldsymbol{\theta}_i) d\mathbf{x} = \int\limits_{\Omega_j} f(\mathbf{x}|c_j, \boldsymbol{\theta}_j) d\mathbf{x}.$$

Finally, the decision regions optimising the other adequate numerical performance scores, i.e. $\mathcal{G}$-DR and $\mathcal{H}$-DR always seem to have limiting values bounded by the EDR and the min-DR. It may be an interesting potential research line to determine how the limiting values of the decision region vary on the class-overlapping for the unweighted Hölder means ($p \leq 1$). In the figures, it can be seen that while Ω_3 has a similar size for these scores, Ω_1 and Ω_2 fluctuate considerably with p.

In conclusion, with this example we reinforce one of the theses of the manuscript. Although, graphically, both the BDR and the EDR seem to be more compelling for splitting the x-axis in the intersection points, the use of classifiers maximising unweighted Hölder means with $p < 1$ may be of interest as we have proved that they are more informative on the class-imbalance problems. More competitive classifiers may be proposed by adding these numerical performance scores to the definition of the forthcoming learning algorithms for unbalanced problems. These adequate classifiers tend to split the x-axis so that the decision region of each class shares the same area in terms of just the density function. Therefore, they will be more reactive to the potential problems derived from the skewness of the class distribution.

C Closest Studies in the Literature about the Class-imbalance Problem

Feature	Manuscript	(Prati et al., 2004)	(Prati et al., 2015)	(Weiss and Provost, 2003)
Type of study	Theoretical	Empirical	Empirical	Empirical
Problem Faced	Multi-class	Binary	Binary	Binary
Classifiers	The BDR and the EDR over the generative model	C4.5 over a set of artificial datasets	Several learning paradigms over a set of real datasets	C4.5 over a set of real datasets
Description	By analysing the BDR, we propose a set of adequate performance scores to assess classifiers in the multi-class unbalanced domain. Moreover, we highlight that several approaches to class-imbalance problems inherently maximise the discovered adequate scores. Finally, we also propose bounds on these scores in order to discriminate competitive from uncompetitive classifiers.	By means of a set of experiments, the authors seek for answer the question of whether class-imbalance, by itself, can degrade the performance of a learning system. They affirmatively answer the question and conclude by claiming that class-overlapping plays an important role in the concept induction, even strong than class-imbalance.	They perform a study in order to relate the performance loss, the degree of class imbalance and the different learning paradigms. Specifically, they address these questions: i Which distribution bounds the unbalanced datasets? ii Are all learning paradigms equally affected by class-imbalance? iii How much of the losses can be recovered by the treatment methods?	The authors analyse, for a previously fixed training set size, the relationship between a changing class distribution in the training data and its resulting classifier performance. Moreover, they proposed an under sampling method to obtain the class distribution producing the best classifier.
Limitations	The **theoretical model is simplified** in order to be able to fully interpret the results. The **state-of-the-art algorithms reviewed are limited** to data sampling (RUS, ROS) and cost-sensitive learning. The **performance scores studies for multi-class unbalanced situation are restricted** to Hölder means among the recalls.	Restricted to **binary problems**. 12 artificial datasets generated with **Gaussian distributions with unit variance**. **Bias is introduced** in the study by the limitation of the training sample size, the use of using C4.5 as learning algorithm instead of the generative model as classifier and the estimation of the error.	Restricted to **binary problems**. The **recovery equation is insensitive** to small losses. Large recovery rates are avoided. Bias is introduced in the study: 22 real-world datasets with unknown generative models are used to feed 6 learning algorithms in order to produce several classifiers whose performance is estimated.	Restricted to **binary problems**. Restricted to just once learning algorithm C4.5. **Bias is introduced** in the study: 26 real-world datasets with unknown generative models and performance estimation.

Table 2: Differences among several similar studies of the class-imbalance problem.

7

Measuring the Class-imbalance Extent of Multi-class Problems

> *Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it.*
>
> - H. James Harrington, *CIO, Vol. 12, No. 23*

Pattern Recognition Letters 98 (2017) 32–38

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Measuring the class-imbalance extent of multi-class problems

Jonathan Ortigosa-Hernández[a,*], Iñaki Inza[a], Jose A. Lozano[a,b]

[a] Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, Computer Science Faculty, The University of the Basque Country UPV/EHU, P. Manuel Lardizabal 1, 20018, Donostia-San Sebastián, Spain
[b] Basque Center for Applied Mathematics BCAM, Alameda de Mazarredo 14, 48009, Bilbao, Spain

ARTICLE INFO

Article history:
Received 2 October 2016
Available online 8 August 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Class-imbalance problem
Skewed class distribution

ABSTRACT

Since many important real-world classification problems involve learning from unbalanced data, the challenging class-imbalance problem has lately received considerable attention in the community. Most of the methodological contributions proposed in the literature carry out a set of experiments over a battery of specific datasets. In these cases, in order to be able to draw meaningful conclusions from the experiments, authors often measure the class-imbalance extent of each tested dataset using imbalance-ratio, i.e. dividing the frequencies of the majority class by the minority class.

In this paper, we argue that, although imbalance-ratio is an informative measure for binary problems, it is not adequate for the multi-class scenario due to the fact that, in that scenario, it groups problems with disparate class-imbalance extents under the same numerical value. Thus, in order to overcome this drawback, in this paper, we propose *imbalance-degree* as a novel and normalised measure which is capable of properly measuring the class-imbalance extent of a multi-class problem. Experimental results show that imbalance-degree is more adequate than imbalance-ratio since it is more sensitive in reflecting the hindrance produced by skewed multi-class distributions to the learning processes.

1. Introduction

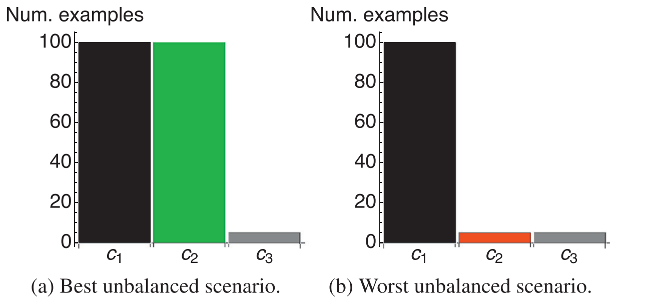
Most of the well-known traditional machine learning techniques are designed to solve classification problems showing reasonably balanced class distributions [24]. However, this assumption does not always hold in reality. Occasionally, real-world problems have skewed class distributions and, due to this, they present training datasets where several classes are represented by an extremely large number of examples, while some others are represented by only a few. This particular situation is known as the class-imbalance problem, a.k.a. learning from unbalanced data [17], and it is considered in the literature as a major obstacle to building precise classifiers: the solutions obtained for problems showing class-imbalance through the traditional learning techniques are usually biased towards the most probable classes showing a poor prediction power for the least probable classes [10]. Thus, in an attempt to overcome this obstacle, hundreds of methodological solutions have been proposed recently in order to balance the prediction powers for both the most and the least probable classes.

According to [28], the proposed solutions can be mainly categorised into the following three major groups: (i) the development of *inbuilt mechanisms* [11], which change the classification strategies to impose a bias toward the minority classes, (ii) the usage of *data sampling methods* [3], which modify the class distribution to change the balance between the classes, and (iii) the adoption of *cost-sensitive learning techniques* [22] which assume higher misclassification costs for examples of the minority classes.

Usually, every paper proposed within those categories shares the same experimental setup: the proposed method is compared against one or several competing methods over a dozen or so datasets. However, although this experimental setup is reasonable enough to support an argument that the new method is “as good as” or “better than” the state-of-the-art, it still leaves many unanswered questions [27]. Besides, it is costly in computing time [30]. Thus, in order to be able to perform more meaningful analyses, some authors complement this experimental schema with a study of the inherent properties of the checked datasets by extracting from them a set of informative measures [30,31]. By means of this data characterisation, more solid empirical conclusions may be efficiently extracted: on the one hand, a better understanding of the problem faced may be achieved since it is a structured manner of investigating and explaining which intrinsic features of the data are affecting the classifiers [2]. On the other hand, the measured data can be related to the classifier performance so that the appli-

[*] Corresponding author.
E-mail address: jonathan.ortigosa@ehu.es (J. Ortigosa-Hernández).

http://dx.doi.org/10.1016/j.patrec.2017.08.002
0167-8655/© 2017 Elsevier B.V. All rights reserved.



(a) Best unbalanced scenario. (b) Worst unbalanced scenario.

Fig. 1. Extreme cases of an unbalanced ternary toy example showing an imbalance-ratio of 20.

cability and performance of a classifier based upon the data can be predicted, avoiding a great amount of computing time [30].

In the literature, authors often measure the class-imbalance extent. In those works, *imbalance-ratio* is the most frequently used summary of the class-imbalance extent due to its simplicity [11]. It reflects the (expected) number of instances of the most probable class for each instance of the least probable class. However, in this paper, we state that whilst it is a very informative summary of the class-imbalance extent for binary problems, it is not capable of completely and honestly describing the disparity among the frequencies of more than two classes. In the multi-class scenario, there exists other classes rather than the most and least probable classes and they are not taken into account for the calculation of this summary. This may lead to the undesired situation of characterising multi-class problems with disparate class-imbalance extents using the same imbalance-ratio.

In order to clarify this drawback, let's consider the toy example presented in Fig. 1; Imagine that a 3-class problem with an imbalance-ratio of 20 (100: 5) is provided. This means that there are 20 examples of the most probable class (c_1) for each example of the least probable class (c_3). However, by means of just imbalance-ratio, little knowledge can be extracted regarding the remaining class c_2, i.e. the number of examples of c_2 can vary from 5 to 100, and all these 95 different possible scenarios share an imbalance-ratio equal to 20.

As can be easily noticed, the scenario with 100 examples for the second class – Fig. 1a –, is far less problematic than having only 5 examples of the second class – Fig. 1b –. While there is only one minority class in the former scenario, we find two minority classes in the latter. So, it can be straightforwardly concluded that imbalance-ratio is not a proper summary of the class-imbalance extent in the multi-class scenario as it groups diverse problems with different class-imbalance extents under the same numerical value.

Thus, in order to bridge this gap, in this paper, we propose a new summary which is capable of properly shortening the class distributions of both binary and multi-class classification problems into a single value. This measure, which we name *imbalance-degree*, represents the existing difference between a purely balanced distribution and the studied unbalanced problem, and it has the following three interesting properties:

1. By means of a single real value in the range [0, K), where K is the number of classes, it not only summarises the class distribution of a given problem but also inherently expresses the number of majority and minority classes.
2. Depending on the requirements of the experimental setup and the degree of sensitivity sought, this measure can be instantiated with any common distance between vectors or divergence between probability distributions.
3. A unique mapping between the class distributions and the numerical value of imbalance-degree is ensured for problems showing different numbers of majority and minority classes. Therefore, diverse problems cannot share a common numerical value as happens with imbalance-ratio.

Experimental results show that imbalance-degree is a more appropriate summary than imbalance-ratio. In the multi-class framework, the former is not only able of differentiating class distributions than the latter groups with the same value but it also achieves a greater correlation with the hindrance that skewed class distributions cause in the learning processes.

The rest of the paper is organised as follows: Section 2 introduces the framework, notation, and a review of the most-commonly used measures and summaries of the class distribution. In Section 3, we introduce imbalance-degree as a more informative measure for the multi-class scenario. After that, Section 4 presents an empirical study of the adequateness of the proposed measure. Finally, Section 5 sums up the paper.

2. Problem formulation and state-of-the-art measures for the class-imbalance extent

Let γ_K be a K-class classification problem with a generative model given by the generalised joint probability density function

$$\rho(\mathbf{x}, c) = p(c)\rho(\mathbf{x}|c), \tag{1}$$

where $p(c)$ is a multinomial distribution representing the class probabilities and $\rho(\mathbf{x}|c)$ is the conditional distribution of the feature space. For convenience, henceforth, we rewrite the former as $\boldsymbol{\eta} = (\eta_1, \eta_2, \dots, \eta_K)$, where each $\eta_i = p(c_i)$ stands for the probability of each categorical class c_i. Also, we denote the special case of equiprobability as $\mathbf{e} = (e_1, e_2, \dots, e_K)$, where $\forall i, \eta_i = 1/K = e_i$. Then, depending on the outline of its class distribution $\boldsymbol{\eta}$, every classification problem γ_K can be catalogued into one of the following groups: (i) γ_K may be a balanced problem, (ii) an unbalanced problem showing multi-majority, or (iii) a multi-minority unbalanced problem. The formal definitions for these groups, as expressed in [17] and [31], are the following:

Definition 1. A K-class classification problem, γ_K, is balanced if it exhibits a uniform distribution between its classes. Otherwise, it is considered to be unbalanced. Formally,

$$\gamma_K \text{ is balanced } \iff \boldsymbol{\eta} = \mathbf{e}. \tag{2}$$

Definition 2. A multi-class classification problem ($K > 2$), γ_K, shows a multi-majority class-imbalance if most of the classes have a higher or equal probability than equiprobability, i.e.

$$\gamma_K \text{ is multi-majority } \iff \sum_{i=1}^{K} \mathbb{1}\left(\eta_i \geq \frac{1}{K}\right) \geq \frac{K}{2}. \tag{3}$$

Definition 3. An unbalanced classification problem, γ_K with $K > 2$, shows a multi-minority class-imbalance when most of the class probabilities are below the equiprobability. Formally,

$$\gamma_K \text{ is multi-minority } \iff \sum_{i=1}^{K} \mathbb{1}\left(\eta_i < \frac{1}{K}\right) > \frac{K}{2}. \tag{4}$$

Here, $\mathbb{1}(\mathcal{E})$ is the indicator function, 1 if the event $\mathcal{E}$ is true, 0 otherwise. Note that Fig. 1a and Fig. 1b correspond to multi-majority and multi-minority problems respectively, and that only when facing multi-class problems do Definition 2 and 3 make sense.

Unfortunately, in most of the real-world cases, the generative model, along with the real class distribution, is unknown. Thus, authors must estimate $\boldsymbol{\eta}$ from a training dataset D in order to not only classify γ_K into one of the groups proposed in the definitions, but also to be capable of using a close approximation of the real class distribution to properly validate the conclusions exposed in their experimental schemas.

Then, let $D = \{(\mathbf{x}^{(1)}, c^{(1)}), \dots, (\mathbf{x}^{(l)}, c^{(l)})\} = \{(\mathbf{x}^{(n)}, c^{(n)})\}_{n=1}^{l}$ be defined as a supervised training dataset of size l drawn from the generative function[1]. There, let the class labels $\{c^{(n)}\}_{n=1}^{l}$ be i.i.d. random values drawn from $\boldsymbol{\eta}$ and let each observation $\{\mathbf{x}^{(n)}\}_{n=1}^{l} \in D$ be also an i.i.d. random value but drawn from $\rho(\mathbf{x}|c_i)$. In order to estimate the class distribution $\boldsymbol{\eta}$, we define the empirical distribution $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_K)$. $\boldsymbol{\zeta}$ is a multinomial distribution with K categories, which exhibits the information available in the dataset about the class distribution of the problem γ_K. There, each statistic ζ_i estimates each class probability η_i by just determining the frequency of the class c_i in the dataset. Formally, the statistic is defined as follows:

$$\zeta_i = \frac{1}{l} \sum_{n=1}^{l} \mathbb{1}(c^{(n)} = c_i). \tag{5}$$

Unless otherwise stated, henceforth, we only use the estimator $\boldsymbol{\zeta}$ of the class distribution since having an unknown generative model is the most common scenario. Anyhow, in the event of knowing the generative model, all the methodologies presented can be directly used with $\boldsymbol{\eta}$ by just substituting the empirical class distribution by the real class distribution in the formulae.

A few measures for the class-imbalance extent of the class distribution using the empirical class distribution $\boldsymbol{\zeta}$ have been already utilised in the experimental setups of the state-of-the-art literature: the most simple manner to measure the class-imbalance extent of a given problem is just to write down the **empirical class distribution** [3], $\boldsymbol{\zeta}$, or to directly transcribe **the occurrences of all the classes** [13,31] in the dataset, i.e. $\mathbf{l} = (l_1, l_2, \dots, l_K)$ s.t. $\forall i, l_i = l\zeta_i$.

These descriptions seem to be a good choice due to the fact that they contain all the information available in the dataset with regards to the class-imbalance extent of the generative class distribution $\boldsymbol{\eta}$. However, analysing them can be quite tedious in problems with a large number of class values, especially in highly multi-class problems ($K \geq 1,000$, [15]). In those cases, it is very common to find unbalanced distributions among the classes. Additionally, these solutions are also more difficult to read and/or compare than single value summaries. Therefore, functions $d(\cdot)$ which assign different single real numbers to disparate values of $\boldsymbol{\zeta}$, i.e. $d : \boldsymbol{\zeta} \mapsto \mathbb{R}$ and which are somehow correlated with the hindrance that skewed class distributions cause on learning algorithms mainly dominate the class-imbalance literature [11,27]. Regarding the summaries, **imbalance-ratio** (IR) between the majority and minority classes is, to the best of our knowledge, the only summary for $\boldsymbol{\zeta}$ used for multi-class problems. It is calculated by dividing the maximum statistic ζ_i by the minimum. Formally,

$$IR(\boldsymbol{\zeta}) = \frac{\max_i \zeta_i}{\min_j \zeta_j}. \tag{6}$$

It is trivial to prove that $IR : \boldsymbol{\zeta} \mapsto \mathbb{R}$ is an injective function for binary problems. This property makes this summary appropriate for such scenarios due to the fact that all possible unbalanced scenarios yield to different IR values and that any $\boldsymbol{\zeta}$ can be easily recovered from the $IR(\boldsymbol{\zeta})$. However, when the number of classes outnumbers 2, the injection is lost (as previously shown in the toy example of Fig. 1, where multi-majority and multi-minority problems share the same numerical value). This is an inappropriate characteristic for a summary of the class-imbalance extent since previous papers have shown that multi-minority problems are harder than multi-majority [31]. This may imply that IR is not correlated with the hindrance produced by skewed multi-class distributions.

Therefore, it can be concluded that neither of the presented measures (summarised in Table 1) for the class-imbalance extent are appropriate for multi-class unbalanced problems.

3. Imbalance-degree

In this section, our aim is to propose a new and more suitable summary for any empirical class distribution $\boldsymbol{\zeta}$ with $K \geq 2$ which, at least, fulfils the following properties: (i) it must be an easily readable finite single valued summary of a multinomial distribution and (ii) it needs to be correlated with the hindrance that highly unbalanced datasets cause in the learning processes.

Thus, since the class distribution does harm the learning processes as it extremely diverges from the balanced one [27], it is immediate to use a distance/similarity function, $d_{\Delta}(\boldsymbol{\zeta}, \mathbf{e})$, between both the empirical and balanced distributions, $\boldsymbol{\zeta}$ and $\mathbf{e}$, to summarise the degree of skewness of a classification problem γ_K. Here, Δ stands for any chosen distance between vectors or divergence between probability distributions which can be found in the literature.

However, just relying on the direct usage of a distance/similarity function has, for our purpose, two undesirable properties which may clash with our aim of having an informative easily readable or comparable summary function:

1. Similar to IR, *different values for different number of majority/minority classes cannot be assured.* For instance, imagine we use the Kullback-Leibler divergence [19] as a summary of two diverse class distributions $\boldsymbol{\zeta}^{(1)} = (0.027009, 0.486495, 0.486495)$ and $\boldsymbol{\zeta}^{2} = (0.712853, 0.143573, 0.143573)$. There, both calculi reach the same value: $d_{KL}(\boldsymbol{\zeta}^{(1)}, \mathbf{e}) = d_{KL}(\boldsymbol{\zeta}^{(2)}, \mathbf{e}) = 0.273$.
2. Although a measure is always a finite positive value, *it is not necessarily upper bounded*. For example, Kullback-Leibler divergence may be unbounded, and Manhattan and Euclidean distances [9], in this context, are upper bounded by the values 2 and 1, respectively.

In order to overcome these drawbacks, we purposely divide the space of class distributions so that we can operate on the distance/similarity function and obtain an adequate summary: let $\mathcal{Z}^K$ be defined as the set containing all the possible empirical distributions $\boldsymbol{\zeta}$ of a K-class problem and let $\mathcal{Z}_m^K \subset \mathcal{Z}^K$, $m \in \{0, 1, \dots, K-1\}$ be a subset containing all the empirical class distributions containing exactly m minority classes. Formally,

$$\mathcal{Z}_m^K \triangleq \left\{ \boldsymbol{\zeta} \in \mathcal{Z}^K : m = \sum_{i=1}^{K} \mathbb{1}\left(\zeta_i < \frac{1}{K}\right) \right\}. \tag{7}$$

Straightaway, this severance of $\mathcal{Z}^K$ into K different subsets $\mathcal{Z}_m^K$ allows us to tackle both problems:

1. On the one hand, *different values for different numbers of minority/majority classes can be directly provided* in the summary function by just forcing different ranges of values to different subsets. Here, the range $(m-1, m]$ is assigned to each subset $\mathcal{Z}_m^K$ in the summary (0 for $\mathcal{Z}_0^K$).
2. On the other hand, a *common upper bound* for each subset, and consequently to the summary, can also be assured by applying a $0-1$ normalisation to the distance of the empirical class distribution (a range of size 1 has been assigned to each subset). This is achieved through the division of $d_{\Delta}(\boldsymbol{\zeta}, \mathbf{e})$ by $d_{\Delta}(\boldsymbol{\iota}_m, \mathbf{e})$, being $\boldsymbol{\iota}_m$ the distribution in $\mathcal{Z}_m^K$ most distant to $\mathbf{e}$.

Then, through the application of these amendments on the distance/similarity function, we define our main proposal as:

[1] Note that we assume that D is i.i.d. from eq. (1). Therefore, in this work, we only focus on the case that the nature of the class-imbalance is in the probability distribution, not on the case of having a biased training dataset.

Table 1
Summary of measures for the class-imbalance extent of the class distribution $\boldsymbol{\eta}$ used in the literature and our proposal.

Measure	Formula	Strength	Weakness	References
Empirical distribution	$\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_K)$	It is the most informative measure.	Difficult to read and/or compare in highly multi-class problems.	[3]
Frequency of the classes	$\mathbf{l} = (l_1, l_2, \dots, l_K)$ s.t. $\forall i, l_i = l\zeta_i$	Very informative (equivalent to the empirical class distribution).	Difficult to read and/or compare in highly multi-class problems.	[31]
Imbalance-ratio	$IR(\boldsymbol{\zeta}) = \max_i \zeta_i / \min_j \zeta_j$	It is a single value and easily readable *summary*.	Inappropriate summary for multi-class problems since the injection is lost.	[27]
Imbalance-degree	$ID(\boldsymbol{\zeta}) = d_\Delta(\boldsymbol{\zeta}, \mathbf{e})/d_\Delta(\boldsymbol{\iota}_m, \mathbf{e}) + (m-1)$	It is a single easily readable *summary* appropriate for binary and multi-class problems.	A total injection can only be achieved by the proper choice of the metric/divergence Δ.	This paper

Definition 4. The **imbalance-degree** (ID) of a multi-class dataset showing an empirical class distribution $\boldsymbol{\zeta}$ is given by

$$ID(\boldsymbol{\zeta}) = \frac{d_\Delta(\boldsymbol{\zeta}, \mathbf{e})}{d_\Delta(\boldsymbol{\iota}_m, \mathbf{e})} + (m-1), \tag{8}$$

where m is the number of minority classes, d_Δ is the chosen distance/similarity function to instantiate ID, and $\boldsymbol{\iota}_m$ is the distribution showing exactly m minority classes with the highest distance to $\mathbf{e}$ ($\arg\max_{\boldsymbol{\zeta}\in\mathcal{Z}_m^K} d_\Delta(\boldsymbol{\zeta}, \mathbf{e})$).

In eq. (8), the term $m-1$ is intentionally added to the normalisation term to ensure different values for different values of m, i.e. $ID(\boldsymbol{\zeta}) \in (m-1, m]$ when $\boldsymbol{\zeta} \in \mathcal{Z}_m^K$. Moreover, in the purely balanced scenario $\boldsymbol{\zeta} = \mathbf{e}$, our proposal $ID(\mathbf{e}) = 0$ due to the fact that, conventionally, $d_\Delta(\mathbf{e}, \mathbf{e})/d_\Delta(\mathbf{e}, \mathbf{e}) = 1$.

4. Empirical study

In order to determine the appropriateness of ID (over IR) as a summary of the class-imbalance extent in the multi-class framework, we define two different sets of experiments to empirically corroborate the following hypotheses:

- H_1: *While IR has a deficient resolution to summarise the class-imbalance extent in the multi-class scenario, ID offers a wide variety of high resolution summaries.*
- H_2: *When used on real-world multi-class classification problems, ID is more sensitive to the class-imbalance extent than IR.* I.e. ID is more accurate than IR in informing about a poor performance of traditional learning systems.

Since ID can be instantiated with any chosen distance/similarity function, we first introduce the measures used in the experiments: from the metrics in the vector space [9], Manhattan[2], Euclidean and Chebyshev distances are chosen. Together, the f-divergences [7], the most utilised measures for probability distributions, are also included. Within the latter group, we introduce Kullback-Leibler divergence [20], Hellinger [18] (closely related to, although different from, the Bhattacharyya distance [4]) and total variation distances, and χ^2-divergence [26]. These measures are mathematically defined in Table 2.

Additionally, in order to use eq. (8), the furthest distribution $\boldsymbol{\iota}_m = (\iota_1, \iota_2, \dots, \iota_K)$ to $\mathbf{e}$ must be calculated for every instantiation and every subset $\mathcal{Z}_m^K$. Opportunely, this class distribution coincides for all the considered measures and for all values of m. It satisfies that

$$\sum_{i=1}^{K} \mathbb{1}(\iota_i = 0) = m \wedge \sum_{i=1}^{K} \mathbb{1}\left(\iota_i = \frac{1}{K}\right) = K - m - 1, \tag{9}$$

i.e. the furthest distribution is composed of (i) m minority classes with zero probability, (ii) $K-m-1$ (all but one) majority classes with probability $1/K$, and (iii) a majority class with the remaining probability $1-(K-m-1)/K$. This distribution always shows the lowest entropy [29] in the subset $\mathcal{Z}_m^K$, whilst the balanced setting $\mathbf{e}$ corresponds to the distribution with the highest entropy in $\mathcal{Z}$. Note that, by symmetry, there may be up to $K!$ different furthest distributions $\boldsymbol{\iota}_m$. Fortunately, $ID(\boldsymbol{\zeta})$ is not affected by an arbitrary choice of $\boldsymbol{\iota}$ since the entropy, $H(\boldsymbol{\iota}_m)$, and distance values, $d_\Delta(\boldsymbol{\iota}_m, \mathbf{e})$, remain equal for all furthest distributions.

Table 2
Mathematical formulae for the distance/similarity functions used to instantiate ID in the empirical studies.

Distance/Similarity function	Δ	$d_\Delta(\boldsymbol{\zeta}, \mathbf{e})$
Metrics in the vector space		
Euclidean distance	EU	$\sqrt{\sum_{i=1}^{K}(\zeta_i - e_i)^2}$
Chebyshev distance	CH	$\max_i \lvert\zeta_i - e_i\rvert$
f-divergences		
Kullback-Leibler divergence	KL	$\sum_{i=1}^{K} \zeta_i \log \frac{\zeta_i}{e_i}$
Hellinger distance	HE	$\frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{K}(\sqrt{\zeta_i} - \sqrt{e_i})^2}$
Total variation distance	TV	$\frac{1}{2}\sum_{i=1}^{K}\lvert\zeta_i - e_i\rvert$
Chi-square divergence	CS	$\sum_{i=1}^{K}\frac{(\zeta_i - e_i)^2}{e_i}$

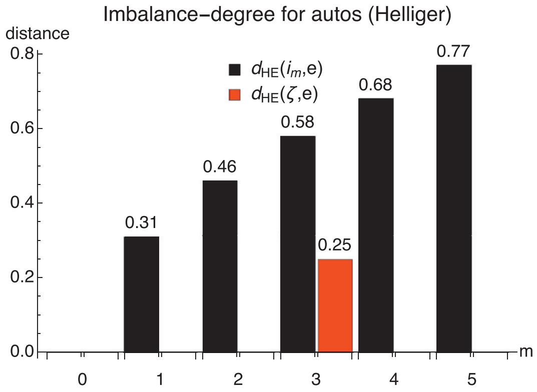


Fig. 2. Calculating ID using the Hellinger distance [18] for the dataset autos ($K = 6$, $IR = 16$, $ID_{HE} = 2.44$).

In order illustrate calculation of ID using the distance/similarity functions considered, Fig. 2 shows, in a bar chart, an example to instantiate ID using the Hellinger distance, $d_{HE}(\boldsymbol{\zeta}, \mathbf{e})$, on the UCI dataset called autos [21]. The numbers above the black bars represent the value of each normaliser $d_\Delta(\boldsymbol{\iota}_m, \mathbf{e})$ for all possible scenarios of m of minority classes in a 6-class problem. Since autos has 3 (out of 6) minority classes and $d_{HE}(\boldsymbol{\zeta}, \mathbf{e}) = 0.25$, the problem has a (normalised) ID of 2.44 ($0.25/0.58 = 0.44$ plus $3-1$).

[2] Manhattan distance has been left out of the experimentation due to the fact that, for our purposes, it is equivalent to total variation distance for any $K \geq 2$.

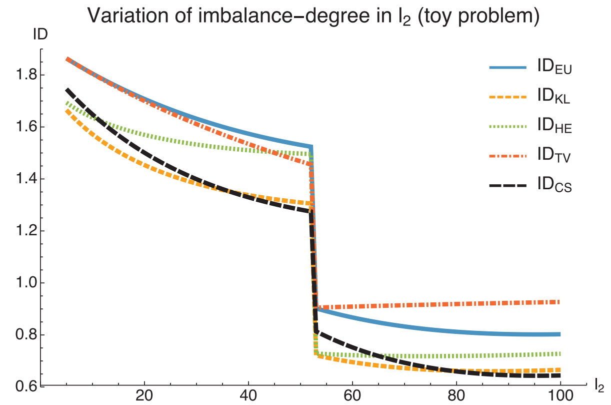


Fig. 3. The variation of ID_Δ in all the 95 possible scenarios ($l_2 = \{5, \ldots, 100\}$) of the toy problem of Fig. 1. For these scenarios, $IR = 20$.

4.1. Study 1: resolution and diverseness of imbalance-degree

The resolution of a measure is the smallest change which can be quantified. As previously put forward, IR cannot be considered as a measure which has a satisfactory resolution for multi-class problems; it only changes based on either the most or the least probable classes. In the toy example, for instance, it groups 95 different class distributions using the value $IR = 20$.

Thus, in order to corroborate the first hypothesis, those 95 scenarios are used to not only show that ID is capable of assigning diverse and reasonable values to them, but also to study the behaviours of the different instantiations of ID. Consequently, Fig. 3 plots the values of ID for the indicated 95 different frequency scenarios, i.e. $\mathbf{l} = (100, l_2, 5)$, where $l_2 = \{5, \ldots, 100\}$, from the toy problem. The abscissa shows the number l_2 of instances of the second class and the ordinate shows the value of ID. From Table 2, Euclidean distance (ID$_{EU}$), Kullback-Leibler divergence (ID$_{KL}$), Hellinger distance (ID$_{HE}$), total variation distance (ID$_{TV}$) and chi-square divergence (ID$_{CS}$) are plotted. Note that Chevbysev distance is not included[3].

Results show that ID is capable of differentiating each and every different scenario that IR groups with the value 20. Moreover, it can be seen that ID instantiations behave differently as result of the diversity of their distance/similarity functions: whilst all the instantiations share a similar monotonically decreasing shape up to the limiting point where the number the m changes from 2 to 1 ($l_2 = 53$), above that limit, two different groups of instantiations can be perceived. On the one hand, ID$_{EU}$, ID$_{KL}$ and ID$_{CS}$ show a convex shape since they descent down to a minimum and then slightly increase. On the other hand, ID$_{HE}$ and ID$_{TV}$ show a quasi-linear behaviour which starts increasing soon after reaching the limiting point. Thus, it can be straightforwardly concluded that there might be instantiations of ID which are more adequate to summarise the class-imbalance extent than others. Seemingly, the latter group of instantiations (ID$_{HE}$ and ID$_{TV}$) are more appropriate as they reflect the increased intricacy of the classification problem above the limiting point. When $l_2 > 53$, the probability of the minority class c_3 distance itself from the equiprobability causing an increase in the intricacy of the classification problem. In Section 4.2, we also deal with this issue by empirically determining which ID instantiations are more adequate summaries in real-world multi-class datasets. Finally, we believe that, in practise, the above mentioned diversity may also be potentially exploited to adapt ID to different requirements and constraints resulting from real-world unbalanced problems.

4.2. Study 2: sensitivity and validity of imbalance-degree

A measure is sensitive to recognise a given set of events if it is capable of valuing them differently. Specifically, we can consider a summary of the class-imbalance extent to be sensitive to recognise the hindrance that highly unbalanced data produce in the traditional learning systems if it is correlated with the performance of those learning systems. Thus, to determine which instantiation of ID is more sensitive to the exposed hindrance than IR, in this section, the following experiment is carried out:

A database containing the 15 unbalanced multi-class datasets recommended in the key work of [1] is assembled and the value of each summary presented in this paper (see Table 2) is calculated for each dataset. Their values, along with some main characteristics of the datasets, are presented in Table 3. There, each row corresponds to a dataset and each column stands for a characteristic (name, features and number of classes) or a summary (empirical class distribution, number of occurrences, IR and IDs). Afterwards, each dataset is used to feed a representative learning algorithm from the traditional major learning paradigms [27]. Specifically, for each problem, a different classifier is learnt using 5 different popular supervised algorithms[4]: C4.5 (Decision trees), RIPPER (Decision rules), Neural Networks (Connectionism), Naïve Bayes (Probabilistic), and SVM (Statistical learning). In order to assess the performance of each learnt classifier, three different performance scores which are highly recommended for multi-class unbalanced problems are used [24]: the arithmetic mean among the recall of the classes ($\mathcal{A}$), the geometric mean among the recalls ($\mathcal{G}$), and the minimum recall obtained (min). In order to obtain the values of these performance scores for each dataset, we estimate them using 10×10 fold cross-validation[5].

Then, the correlation between the estimated values for the performance scores and the summaries, IR and ID, are determined using the Pearson product-moment correlation coefficient [25] so that H$_2$ may be checked. Since a licit calculation of the correlation requires an ideal scenario with a fixed number of minority/majority classes, we emulate this requirement by subtracting $(m-1)$ from the ID value before the calculation so that all considered classification problems are normalised in the same range [0, 1]. The results are presented in Table 4; rows represent the summaries and columns represent the estimated values for each score in each learning paradigm. Since the utilised scores assign higher values to better performance, an adequate summary is expected to have a negative correlation; the lowest the correlation, the better the sensitivity. We conclude from the results that summaries are, in general, negatively correlated with the performance of the classifiers, and that instantiations of ID are more sensitive than IR as the former obtain a lower negative correlation. The best results (highlighted in Table 4) are obtained by ID$_{TV}$ and ID$_{HE}$.

Finally, to determine if there are summaries significantly more sensitive to the hindrance produced by skewed class distributions, a statistical hypothesis testing procedure is performed: Friedman test [8] with Shaffer's static post-hoc with $\alpha = 0.05$ [12]. The test results are represented by means of critical difference diagrams (CDD) [8], which show, in a numbered line, the arithmetic mean of the ranks of the correlation between each summary and the estimation of each score in the database. If there is no statistically significant difference between two summaries, they are connected in the diagram by a straight grey line. Figs. 4a, 4b, and 4 show the

[3] It holds that instantiations of ID using Chevbysev and total variation distances are equivalent for the case $K = 3$.

[4] In this experimentation, all learning and error estimation tasks have been performed using the software `Weka 3` [16].

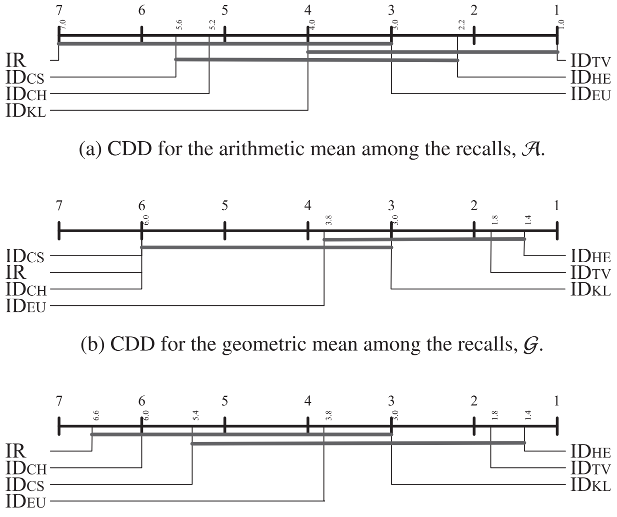
[5] These results can be downloaded, along with the source code.

Table 3
Characteristics of the studied unbalanced datasets [1] and the value of the summaries introduced in this paper for each dataset.

Dataset	$\|\mathbf{x}\|$	K	Empirical class distribution	Occurrences	IR	ID_{EU}	ID_{CH}	ID_{KL}	ID_{HE}	ID_{TV}	ID_{CS}
Autos	25	6	0.02/0.13/0.30/0.29/0.18/0.08	3/20/48/46/29/13	16.0	2.44	2.30	2.24	2.44	2.55	2.19
Balance	4	3	0.46/0.08/0.46	288/49/288	5.9	0.66	0.76	0.40	0.53	0.76	0.44
Contraceptive	9	3	0.43/0.23/0.35	629/333/511	1.9	0.30	0.32	0.07	0.20	0.32	0.09
Dermatology	34	6	0.31/0.17/0.20/0.13/0.14/0.05	112/61/72/49/52/20	5.6	2.32	2.28	2.11	2.29	2.34	2.10
Ecoli	7	8	0.43/0.23/0.15/0.10/ 0.06/0.01/0.01/0.01	143/77/52/35/ 20/5/2/2	71.5	4.56	4.48	4.42	4.61	4.70	4.31
Glass	9	7	0.33/0.36/0.08/0.00/0.06/0.04/0.14	70/76/17/0/13/9/29	8.4(∞)	4.44	4.30	4.28	4.53	4.56	4.20
Hayes-Roth	4	3	0.39/0.39/0.23	51/51/30	1.7	0.28	0.32	0.06	0.19	0.32	0.08
Lymphography	18	4	0.01/0.55/0.41/0.03	2/81/61/4	40.5	1.77	1.59	1.65	1.73	1.92	1.59
New-thyroid	5	3	0.70/0.16/0.14	150/35/30	5.0	1.55	1.55	1.25	1.40	1.55	1.30
Pageblocks	10	5	0.90/0.06/0.01/0.02/0.01	492/33/8/12/3	164.0	3.87	3.87	3.73	3.75	3.87	3.76
Penbased	16	10	0.10/0.10/0.10/0.10/0.10/ 0.10/0.10/0.10/0.10/0.10	115/114/114/106/114/ 106/105/115/105/106	1.1	4.02	4.01	4.00	4.02	4.04	4.00
Shuttle	9	7	0.78/0.00/0.00/0.16/0.06/0.00/0.00	1706/2/6/338/123/0/0	853.0(∞)	4.90	4.90	4.83	4.88	4.92	4.82
Thyroid	21	3	0.02/0.05/0.93	17/37/666	39.2	1.89	1.89	1.72	1.73	1.89	1.79
Wine	13	3	0.33/0.40/0.27	59/71/48	1.5	1.11	1.10	1.01	1.09	1.10	1.01
Yeast	8	10	0.16/0.29/0.31/0.03/0.03/ 0.11/0.02/0.02/0.01/0.00	244/429/463/44/51/ 163/35/30/20/5	92.6	5.54	5.35	5.42	5.60	5.79	5.29

Table 4
Pearson correlation coefficient (× 100) among the performance of the major learning paradigms on the datasets of Table 3 and the studied summaries.

Summary	Decision trees (C4.5)			Decision rules (RIPPER)			Connectionism (Neural Net.)			Probabilistic (Naïve Bayes)			Statistical learn. (SVM)		
	$\mathcal{A}$	$\mathcal{G}$	min	$\mathcal{A}$	$\mathcal{G}$	min	$\mathcal{A}$	$\mathcal{G}$	min	$\mathcal{A}$	$\mathcal{G}$	min	$\mathcal{A}$	$\mathcal{G}$	min
IR	−9	−42	−41	14	−41	−38	−11	−50	−43	−4	−42	−39	−14	−34	−33
$ID_{EU} - (m+1)$	−27	−46	−52	−15	−46	−45	−51	−65	−61	−40	−56	−57	−60	−63	−65
$ID_{CH} - (m+1)$	−19	−39	−42	−4	−37	−34	−35	−50	−44	−29	−50	−51	−55	−56	−59
$ID_{KL} - (m+1)$	−25	−50	−54	−13	−50	−49	−54	−75	−72	−39	−59	−63	−54	−60	−61
$ID_{HE} - (m+1)$	−34	−56	−63	−25	−57	−59	−59	−77	−75	−48	−67	−68	−59	−69	−71
$ID_{TV} - (m+1)$	−42	−60	−66	−32	−60	−59	−62	−73	−69	−52	−64	−66	−61	−68	−70
$ID_{CS} - (m+1)$	−14	−38	−42	−1	−38	−36	−44	−63	−60	−31	−50	−54	−53	−55	−56



(a) CDD for the arithmetic mean among the recalls, $\mathcal{A}$.

(b) CDD for the geometric mean among the recalls, $\mathcal{G}$.

(c) CDD for minimum recall obtained, min.

Fig. 4. Pearson correlation ranking between the performance of the supervised learning paradigms on the studied datasets and the summaries, $\alpha = 0.05$.

CDD for the Pearson correlation between the summaries and $\mathcal{A}$, $\mathcal{G}$ and min, respectively. Results confirm the second hypothesis, in all rankings, IR shows the worst behaviour and significant differences are found between IR and other instantiations of ID for the performance scores. Moreover, they also show that instantiating ID using either Hellinger or total variation (Manhattan) distances produces significant robust summaries of the class-imbalance extent.

5. Summary

Authors often measure the class-imbalance extent in their experimental schemas when there is a reasonable suspicion of having unbalanced problems in the checked database. Up to now, the most utilised summary of the class-imbalance extent of a dataset was the imbalance-ratio, i.e. the (expected) number of instances of the most probable class for each instance of the least probable class. Although it is a powerful measure for binary problems, in this paper, we prove that it is a suboptimal summary for the multi-class scenario. For that reason, we propose a new more adequate and robust summary of the class-imbalance extent to deal with multiple classes, named *imbalance-degree*. It has three interesting properties: (i) it is a single easy-readable real value in the range [0, K), where K is the number of classes. (ii) Depending on the requirements of the sensitivity sought in the tackled problem, it can be instantiated by any chosen metric or divergence. (iii) It is an injective function for different class distributions showing different numbers of majority/minority classes. Empirical results show that imbalance-degree has a higher resolution and is more sensitive to express the hindrance that skewed class distributions cause in the traditional supervised algorithms than imbalance-ratio. Additionally, it can also be concluded that either Hellinger, total variation or Manhattan distances are recommended distance/similarity functions to instantiate our proposal, imbalance-degree.

This work can be extended in several ways. For example, only 8 different distance/similarity functions over 15 datasets are used in this paper. A more exhaustive analysis can be carried out using a larger number of distance/similarity functions [5,14] over a larger set of unbalanced problems in order to statistically determine which functions behave differently and are recommended for highly different class-imbalanced scenarios.

Another straightforward future path to this research can be a study on the variation of the correlation between ID and the performance of the classifiers when class-imbalance techniques, such as SMOTE [6], are used. This could be a step forward in determining which intrinsic features of the data are affecting the classifiers [2], and whether the performance of a classifier can be predicted based upon the available data [30]. However, note that, although the negative correlation between ID and the performance is expected to decrease as long as the class-imbalance techniques alleviate the hindering effect of the class distribution, there might exist other hindering aspects [23] which may harm the performance of the classifiers.

Acknowledgements

This work is partially supported by the Basque Government (IT609-13 and Elkartek BID3A) and the Spanish Ministry of Economy and Competitiveness (TIN2013-41272P). Moreover, Jonathan Ortigosa-Hernández is partly financed by the Ministry of Science and Innovation (MEC-FPU AP2008-00766) and Jose A. Lozano by both the Basque Government (BERC program 2014–2017) and the Spanish Ministry of Economy and Competitiveness (Severo Ochoa Program SEV-2013-0323).

References

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrara, Bkeel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2011) 255–287.

[2] N. Anwar, G. Jones, S. Ganesh, Measurement of data complexity for classification problems with unbalanced data., Stat. Anal. Data Min. (2014) 194–211.

[3] G. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explor. Newsl. 6 (2004) 20–29.

[4] A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions, Bull. Calcutta Math. Soc. 35 (1943) 99–109.

[5] S. Cha, Comprehensive survey on distance/similarity measures between probability density functions, Int. J. Math. Models Methods Appl. Sci. 1 (2007) 300–307.

[6] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[7] I. Csiszár, Eine informations theoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von Markoffschen Ketten, Magyar. Tud. Akad. Mat. Kutató Int. Közl 8 (1963) 85–108.

[8] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[9] M.M. Deza, E. Deza, Encyclopedia of distances, Springer-Verlag Berlin Heidelberg, 2009.

[10] C. Drummond, R. Holte, Severe class imbalance: why better algorithms aren't the answer, in: Proc. of the 16th European Conf. on Machine Learning, 2005, pp. 539–546.

[11] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man, Cybern. C: Appl. Rev. 42 (2012) 463–484.

[12] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets", J. Mach. Learn. Res. 9 (2008) 2677–2694.

[13] A.S. Ghamen, S. Venkatesh, G. West, Multi-class pattern classification in imbalanced data, in: Proc. of the Int. Conf. on Pattern Recognition 2010, 2010, pp. 2881–2884.

[14] A.L. Gibbs, F.E. Su, On choosing and bounding probability metrics, Int. Stat. Rev. 70 (2002) 419–435.

[15] M. Gupta, S. Bengio, J. Weston, Training highly multiclass classifiers, J. Mach. Learn. Res. 15 (2014) 1461–1492.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, R. Reutemann, I. Witter, The WEKA data mining software: an update, SIGKDD Explor. 11 (2009) 10–18.

[17] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. on Knowledge and Data Engineering 21 (2009) 1263–1284.

[18] E. Hellinger, Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen, Journal für die reine und angewandte Mathematik 136 (1909) 210–271.

[19] S. Kullback, Information Theory and Statistics, Wiley, 1959.

[20] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 9–86.

[21] M. Lichman, UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml.

[22] X.Y. Liu, Z.H. Zhou, The influence of class imbalance on cost-sensitive learning: an empirical study, in: Proc. of the 6th Int. Conf. on Data Mining (ICDM'06), 2006, pp. 970–974.

[23] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. (Ny) 250 (2013) 113–141.

[24] J. Ortigosa-Hernández, I. Inza, J.A. Lozano, Towards competitive classifiers for the class-imbalance problem, arXiv:1608.08984 (2016) 1–21.

[25] K. Pearson, Notes on regression and inheritance in the case of two parents, in: Proc. of the Royal Society of London 58, 1895, pp. 240–242.

[26] K. Pearson, On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling, 1900.

[27] R.C. Prati, G.E. Batista, D.F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, Knowl. Inf. Syst. 45 (2015) 247–270.

[28] J. Sáez, B. Krawczyk, M. Woźniak, Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets, Pattern Recognit. 57 (2016) 164–178.

[29] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423.

[30] J. Sotoca, J. Sánchez, R. Mollineda, A review of data complexity measures and their applicability to pattern classification problems, in: Proceedings of the III Simposio de Teoria y Aplicaciones de Mineria de Datos (TAMIDA 2005), 2005, pp. 77–83.

[31] S. Wang, X. Yao, Multi-class imbalance problems: analysis and potential solutions, IEEE Trans. Syst. Man Cybern. B 42 (2012) 1119–1130.

References

[1] E. Brynjolfsson and T. Mitchell, “What can machine learning do? Workforce implications,” *Science*, vol. 358, pp. 1530–1534, Dec. 2017.

[2] R. Kohavi and F. Provost, “Glossary of terms,” *Machine Learning*, vol. 30, pp. 271–274, Feb-Mar. 1998.

[3] P. Simon, *Too big to ignore: the business case for big data.* New Delhi, India: Wiley, 1st ed., 2013.

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* Upper Saddle River, NJ: Prentice Hall, 2nd ed., 2002.

[5] A. N. Kolmogorov, *Foundations of the Theory of Probability.* White River Junction, VT: Chelsea Publishing Company, 1st ed., 1950.

[6] V. Dhar, “Data science and prediction,” *Communications of the ACM*, vol. 56, pp. 64–73, Dec. 2013.

[7] K. Schwab, *The Fourth Industrial Revolution.* Danvers, MA: Crown Publishing Group, 1st ed., 2017.

[8] T. H. Davenport and P. D. J., “Data scientist: The sexiest job of the 21st century,” *Harvard Business Review*, vol. 90, pp. 70–76, 2012.

[9] R. Duda, P. Hart, and D. Stork, *Pattern classification.* New York City, NY: Wiley Inter-Science, 1st ed., 2001.

[10] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised Learning.* Cambridge, MA: The MIT Press, 1st ed., 2006.

[11] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, Sep. 2009.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* New York City, NY: Springer, 1st ed., 2006.

[13] C. Bielza, G. Li, and P. Larrañaga, “Multi-dimensional classification witn Bayesian networks,” *International Journal of Approximate Reasoning*, vol. 52, pp. 705–727, Sep. 2011.

[14] S. Axelsson, “The base-rate fallacy and the difficulty of intrusion detection,” *ACM Transactions on Information and Systems Security*, vol. 3, pp. 186–205, Aug. 2000.

[15] O. Tsur, D. Davidiv, and A. Rappoport, “A great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews,” in *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, Washington, DC, May 2010.

[16] B. Calvo, N. López-Bigas, S. J. Furney, P. Larrañaga, and J. A. Lozano, “A partially supervised classification approach to dominant and recessive human disease gene prediction,” *Computer Methods and Programs in Biomedicine*, vol. 85, pp. 229–237, Mar. 2007.

[17] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, pp. 179–188, Sep. 1936.

[18] A. Paton, N. Brummitt, R. Govaerts, K. Harman, S. Hinchcliffe, B. Allkin, and E. Lughadha, “Target 1 of the global strategy for plant conservation: a working list of all known plant speciesprogress and prospects,” *Taxon*, vol. 57, pp. 602–611, May 2008.

[19] M. Gupta, S. Bengio, and J. Weston, “Training highly multiclass classifiers,” *Journal of Machine Learning Research*, vol. 15, pp. 1461–1492, Apr. 2014.

[20] M. L. Zhang and Z. H. Zhou, “A review on multi-label learning algorithms,” *IEEE Transactions on Knowdelge and Data Engineering*, vol. 26, pp. 1819–1837, Aug. 2014.

[21] S. Shuaib Ahmed, B. Puma Chandra Rao, and T. Jayakumar, “Radial basis functions for multidimensional learning with an application to nondestructive sizing of defects,” in *Proceedings of the 2013 IEEE Symposium on Foundations of Computational Intelligence*, pp. 38–43, Singapore, Apr. 2013.

[22] J. A. Fernandes, J. A. Lozano, I. Inza, X. Irigoien, A. Pérez, and J. D. Rodríguez, “Supervised pre-processing approaches in multiple class variables classification for fish recruitment forecasting,” *Environmental Modelling and Software*, vol. 40, pp. 245–254, Feb. 2013.

[23] J. Ortigosa-Hernández, J. Rodríguez, L. Alzate, M. Lucania, I. Inza, and J. A. Lozano, “Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers,” *Neurocomputing*, vol. 92, pp. 98–115, Sep. 2012.

[24] B. Krishnapuram, S. Yu, and R. B. Rao, *Cost-Sensitive Machine Learning.* Boca Raton, FL: CRC Press Inc., 1st ed., 2011.

[25] E. McClennen, “Pascal’s wager and finite decision theory,” in *Gambling on God: Essays on Pascal’s Wager* (J. Jordan, ed.), pp. 115–137, Lanham, MD: Rowman & Littlefield, 1st ed., 1994.

[26] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis.* New York City, NY: Springer, 1st ed., 1985.

[27] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, “Convexity, classification, and risk bounds,” *Journal of the American Statistical Association*, vol. 101, pp. 138–156, Nov. 2006.

[28] T. Y. Young and T. W. Calvert, *Classification, Estimation, and Pattern Recognition.* Oxford, UK: Elsevier Science Ltd, 1st ed., 1974.

[29] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proceedings of the 15th International Conference on Machine Learning*, pp. 445–453, Madison, WI, Jul 1998.

[30] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, "Towards competitive classifiers for unbalanced classification problems: A study on the performance scores," *arXiv:1608.08984*, pp. 1–21, 2016.

[31] V. Castelli and T. M. Cover, "On the exponential value of labeled samples," *Pattern Recognition Letters*, vol. 16, pp. 105–111, Jan. 1995.

[32] M. DeGroot, *Optimal Statistical Decisions.* New York City, NY: John Wiley & Sons, 1st ed., 1970.

[33] T. M. Mitchell, "The need for biases in learning generalizations," Tech. Rep. CBM-TR-117, Rutgers University, New Brunswick, NJ, 1980.

[34] J. Hernández-González, I. Inza, and J. A. Lozano, "Weak supervision and other non-standard classification problems: A taxonomy," *Pattern Recognition Letters*, vol. 69, pp. 49–55, Jan. 2016.

[35] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Series in Data Management Systems, Burlington, MA: Morgan Kaufmann Publishers Inc., 2nd ed., 2005.

[36] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, "Semisupervised multiclass classification problems with scarcity of labelled data: A theoretical study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, pp. 2602–2614, Dec. 2016.

[37] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, University of Wisconsin, Madison, WI, 2005.

[38] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: Theory, algorithms and their application to human-computer interaction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1553–1567, Dec. 2004.

[39] J. H. Krijthe, "RSSL: Semi-supervised learning in R," *arXiv:1612.07993*, pp. 1–12, 2016.

[40] T. Kanamori, A. Takeda, and T. Suzuki, "Conjugate relation between loss functions and uncertainty sets in classification problems," *Journal of Machine Learning Research*, vol. 14, pp. 1461–1504, Jun. 2013.

[41] R. C. Prati, G. E. A. P. A. Batista, and D. F. Silva, "Class imbalance revisited: a new experimental setup to assess the performance of treatment methods," *Knowledge and Information Systems*, vol. 45, pp. 247–270, Oct. 2015.

[42] S. L. Salzberg, "C4.5: Programs for machine learning," *Machine Learning*, vol. 16, pp. 235–240, Sep. 1994.

[43] R. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning. An Artificial Intelligence Approach.* Heidelberg, Germany: Springer, 1st ed., 1984.

[44] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 115–123, Tahoe City, CA, Jul. 1995.

[45] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* New York City, NY: Springer, 1st ed., 2009.

[46] N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines: And Other Kernel-based Learning Methods.* Cambridge, MA: Cambridge University Press, 1st ed., 2000.

[47] N. Gruzling, *Linear separability of the vertices of an n-dimensional hypercube.* PhD thesis, University of Northern British Columbia, Prince George, Canada, 2006.

[48] L. Pei, J. Liu, R. Guinness, Y. Chen, H. Kuusniemi, and R. Chen, "Using LS-SVM based motion recognition for smartphone indoor wireless positioning," *Sensors*, vol. 12, pp. 6155–6175, May 2012.

[49] J. Xu, X. Liu, Z. Huo, C. Deng, F. Nie, and H. Huang, "Multi-class support vector machine via maximizing multi-class margins," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3154–3160, Melbourne, Australia, Aug. 2017.

[50] D. Medler, "A brief history of connectionism," *Neural Computing Surveys*, vol. 1, pp. 61–101, Jan. 1998.

[51] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 65–386, Jun. 1958.

[52] J. L. McClelland and D. E. Rumelhart, "A distributed model of human learning and memory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2* (D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, eds.), pp. 170–215, Cambridge, MA: The MIT Press, 1st ed., 1986.

[53] U. López-Novoa, A. Mendiburu, and J. Miguel-Alonso, "A survey of performance modeling and simulation techniques for accelerator-based computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, pp. 272–281, Feb. 2015.

[54] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, pp. 1097–1105, Lake Tahoe, Nevada, Dec. 2012.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, pp. 1–14, 2014.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, Jun. 2016.

[57] L. C. van der Gaag and P. R. de Waal, "Multi-dimensional Bayesian network classifiers," in *Proceedings of the 3rd European Workshop in*

Probabilistic Graphical Models, pp. 107–114, Prague, Czech Republic, Sep. 2006.

[58] J. D. Rodríguez and J. A. Lozano, "Learning Bayesian network classifiers for multi-dimensional supervised classification problems by means of a multi-objective approach," Tech. Rep. EHU-KZAA-TR-3-2010, University of the Basque Country UPV/EHU, San Sebastián, Spain, 2010.

[59] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Burlington, MA: Morgan Kaufmann Publishers Inc., 1st ed., 1988.

[60] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, Oct. 1992.

[61] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 223–228, San Jose, CA, Jul. 1992.

[62] C. Bielza and P. Larrañaga, "Discrete bayesian network classifiers: A survey," *ACM Computing Surveys*, vol. 47, pp. 1–43, Jul. 2014.

[63] P. R. de Waal and L. C. van der Gaag, "Inference and learning multidimensional Bayesian network classifiers," *Lecture Notes in Artificial Intelligence*, vol. 4724, pp. 501–511, 2007.

[64] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, Nov. 1997.

[65] M. Sahami, "Learning limited dependence Bayesian classifiers," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 335–338, Portland, OR, Aug. 1996.

[66] V. Castelli and T. M. Cover, "The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter," *IEEE Transactions on Information Theory*, vol. 42, pp. 2102–2117, Nov. 1996.

[67] J. Ratsaby and S. Venkatesh, "Learning from a mixture of labeled and unlabeled examples with parametric side information," in *Proceedings of the 8th Annual Conference on Computational Learning Theory*, pp. 412–417, Santa Cruz, CA, Jul. 1995.

[68] T. Zhang and F. J. Oles, "A probability analysis on the value of unlabeled data for classification problems," in *Proceedings of the International Conference on Machine Learning*, pp. 1191–1198, Jul. 2000.

[69] K. Sinha and M. Belkin, "The value of labeled and unlabeled examples when the model is imperfect," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, pp. 1361–1368, Vancouver, Canada, Dec. 2008.

[70] H. Chen and L. Li, "Semisupervised multicategory classification with imperfect model," *IEEE Transactions on Neural Networks*, vol. 20, pp. 1594–1603, Oct. 2009.

[71] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning, Burlington, MA: Morgan Kaufmann Publishers Inc., 1st ed., 2009.

[72] I. Cohen, *Semisupervised Learning of Classifiers with Application to Human-Computer Interaction.* PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, 2003.

[73] J. H. Krijthe and M. Loog, "Robust semi-supervised least squares classification by implicit constraints," *Pattern Recognition*, vol. 63, pp. 115–126, Mar. 2017.

[74] J. H. Krijthe and M. Loog, "Projected estimators for robust semi-supervised classification," *Machine Learning*, vol. 106, pp. 993–1008, Jul. 2017.

[75] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, pp. 103–134, May 2000.

[76] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms.* Cambridge, MA: Cambridge University Press, 1st ed., 2002.

[77] J. Wang, X. Shen, and W. Pan, "On transductive support vector machines," *Contemporary Mathematics*, vol. 443, pp. 7–20, Jan. 2007.

[78] S. Ben-David, T. Lu, and D. Pal, "Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning," in *Proceedings of the 21st Annual Conference on Learning Theory*, pp. 33–44, Helsinki, Finland, Jul. 2008.

[79] S. Koço and C. Capponi, "On multi-class classification through the minimization of the confusion matrix norm," in *Proceedings of the 5th Asian Conference on Machine Learning*, pp. 277–292, Canberra, Australia, Nov. 2013.

[80] G. Santafe, I. Inza, and J. A. Lozano, "Dealing with the evaluation of supervised classification algorithms," *Artificial Intelligence Review*, vol. 44, pp. 467–508, Dec. 2015.

[81] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," in *Proceedings of the 4th International Symposium on Advances in Computation and Intelligence*, pp. 461–471, Huangshi, China, Oct. 2009.

[82] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, Jun. 2006.

[83] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval.* Cambridge, MA: Cambridge University Press, 1st ed., 2008.

[84] C. J. V. Rijsbergen, *Information Retrieval.* Newton, MA: Butterworth-Heinemann, 2nd ed., 1979.

[85] Y. Tang, Y. Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 281–288, Dec. 2009.

[86] M. Di Martino, A. Martínez, P. Iturralde, and F. Lecumberry, "Novel classifier scheme for imbalanced problems," *Pattern Recognition Letters*, vol. 34, pp. 1146–1151, Jul. 2013.

[87] A. Menon, H. Narasimhan, S. Agarwal, and S. Chawla, "On the statistical consistency of algorithms for binary classification under class imbalance," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 603–611, Atlanta, GA, Jun. 2013.

[88] M. Kubat, R. Holte, and S. Matwin, "Learning when negative examples abound," in *Proceedings of the 9th European Conference on Machine Learning*, pp. 146–153, London, UK, Apr. 1997.

[89] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, Nov. 2013.

[90] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145–1159, Jul. 1997.

[91] F. Provost and P. Domingos, "Well-trained pets: Improving probability estimation trees," Tech. Rep. IS-00-04, New York University, New York City, NY, 2000.

[92] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine Learning*, vol. 45, pp. 171–186, Nov. 2001.

[93] J. D. Rodríguez and J. A. Lozano, "Multi-objective learning of multidimensional Bayesian classifiers," in *Proceedings of the 8th Conference on Hybrid Intelligent Systems*, pp. 501–506, Barcelona, Spain, Sep. 2008.

[94] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 289–300, Mar. 2002.

[95] M. Denil and T. Trappenberg, "Overlap versus imbalance," in *Proceedings of the 23rd Canadian Conference on Advances in Artificial Intelligence*, pp. 220–231, Ottawa, Canada, May 2010.

[96] H. M. Kalayeh and D. A. Landgrebe, "Predicting the required number of training samples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 664–667, Jun. 1983.

[97] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, pp. 343–370, Apr. 1988.

[98] R. C. Holte, L. E. Acker, and B. W. Porter, "Concept learning and the problem of small disjuncts," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 813–818, Detroit, MI, Nov. 1989.

[99] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "Class imbalances versus class overlapping: An analysis of a learning system behavior," in *Proceedings of the 3rd Mexican International Conference on*

Artificial Intelligence, vol. 2972, pp. 312–321, Mexico City, Mexico, Apr. 2004.

[100] M. Kubat and S. Matwin, "Addressing the course of imbalanced training sets: One-side selection," in *Proceedings of the 14th International Conference on Machine Learning*, pp. 179–186, Nashville, TN, Jul. 1997.

[101] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, Oct. 2003.

[102] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, pp. 177–210, Nov. 2003.

[103] J. A. Sáez, B. Krawczyk, and M. Woźniak, "Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets," *Pattern Recognition*, vol. 57, pp. 164–178, Sep. 2016.

[104] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Folleco, "An empirical study of the classification performance of learners on imbalanced and noisy software quality data," *Information Sciences*, vol. 259, pp. 571–595, Feb. 2014.

[105] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, pp. 7–19, Jun. 2004.

[106] H. Xiong, J. Wu, and L. Liu, "Classification with class overlapping: A systematic study," in *Proceedings of the 2010 International Conference on E-Business Intelligence*, pp. 491–497, Kunming, Chuna, Dec. 2010.

[107] C. Drummond and R. Holte, "Severe class imbalance: Why better algorithms aren't the answer," in *Proceedings of the 16th European Conference on Machine Learning*, pp. 539–546, Porto, Portugal, Oct. 2005.

[108] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 42, pp. 463–484, Jul. 2012.

[109] G. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations Newsletter*, vol. 6, pp. 20–29, Jun. 2004.

[110] X.-Y. Liu and Z.-H. Zhou, "The influence of class imbalance on cost-sensitive learning: An empirical study," in *Proceedings of the 6th International Conference on Data Mining*, pp. 970–974, Hong Kong, China, Dec. 2006.

[111] S. Wang and X. Yao, "Multi-class imbalance problems: Analysis and potential solutions," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 42, pp. 1119–1130, Aug. 2012.

[112] B. Liu, "Sentiment analysis and subjectivity," in *Handbook of Natural Language Processing* (N. Indurkhya and F. J. Damerau, eds.), pp. 627–666, London, UK: Chapmand & Hall, 2nd ed., 2010.

[113] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proceedings of the*

2002 Conference on Empirical Methods in Natural Language Processing, pp. 79–86, Philadelphia, PA, Jul. 2002.

[114] V. Ng, S. Dasgupta, and S. M. N. Arifin, "Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews," in *Proceedings of the 2006 International Committee on Computational Linguistics and the Association for Computational Linguistics Conference*, pp. 611–618, Sydney, Australia, Jul. 2006.

[115] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 129–136, Sapporo, Japan, Jul. 2003.

[116] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 105–112, Sapporo, Japan, Jul. 2003.

[117] S. Argamon, C. Whitelaw, P. Chase, S. Raj Hota, N. Garg, and S. Levitan, "Stylistic text classification using functional lexical features," *Journal of the American Society for Information Science and Technology*, vol. 58, pp. 802–822, Feb. 2007.

[118] A. Abbasi, H. Chen, S. Thoms, and T. Fu, "Affect analysis of web forums and blogs using correlation ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 1168–1180, Sep. 2008.

[119] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[120] N. Friedman, "The Bayesian structural EM algorithm," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 129–138, Morgan Kaufmann Publishers, Madison, WI, Jul. 1998.

[121] R. Kohavi, *Wrappers for Performance Enhancement and Oblivious Decision Graphs.* PhD thesis, Stanford University, 1995.

[122] R. Plutchik, *Emotion: A Psychoevolutionary Synthesis.* New York City, NY: Harper and Row, 1st ed., 1980.

[123] B. Everitt and D. J. Hand, *An Introduction to Finite Mixture Distributions.* London, UK: Chapman and Hall, 1st ed., 1981.

[124] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, pp. 429–449, Oct. 2002.

[125] G. M. Weiss, "Foundations of imbalanced learning," in *Imbalanced Learning: Foundations, Algorithms, and Applications* (H. He and Y. Ma, eds.), pp. 13–42, Hoboken, NJ: Wiley-IEEE Press, 1st ed., 2013.

[126] P. S. Bullen, *Handbook of Means and Their Inequalities.* Dordrecht, Netherlands: Springer, 1st ed., 2003.

[127] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 973–978, Seattle, WA, Aug. 2001.

[128] W. E. Deming, *The New Economics for Industry, Government, Education.* Cambridge, MA: The MIT Press, 2nd ed., 2000.

[129] A. S. Ghamen, S. Venkatesh, and G. West, "Multi-class pattern classification in imbalanced data," in *Proceedings of the 20th International Conference on Pattern Recognition*, pp. 2881–2884, Istanbul, Turkey, Aug. 2010.

[130] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, "Measuring the class-imbalance extent of multi-class problems," *Pattern Recognition Letters*, vol. 98, pp. 32–38, Oct. 2017.

[131] M. M. Deza and E. Deza, *Encyclopedia of Distances.* Heidelberg, Germany: Springer, 1st ed., 2009.

[132] I. Csiszár, "Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten," *Magyar Tudományos Akadémia Matematikai Kutató Intézet Közlemény'*, vol. 8, pp. 85–108, Jan. 1963.

[133] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, pp. 9–86, Mar. 1951.

[134] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen," *Journal für die Reine und Angewandte Mathematik*, vol. 136, pp. 210–271, Jan. 1909.

[135] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," in *Breakthroughs in Statistics: Methodology and Distribution* (S. Kotz, ed.), pp. 11–28, New York City, NY: Springer, 1st ed., 1992.

[136] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, pp. 255–287, Jan. 2011.

[137] F. Charte, A. Rivas, M. J. del Jesús, and F. Herrera, "A first approach to deal with imbalance in multi-label datasets," in *Proceedings of the 2013 International Conference on Hybrid Artificial Intelligence Systems*, pp. 150–160, Salamanca, Spain, Sep. 2013.

[138] J. Ortigosa-Hernández, J. Rodríguez, L. Alzate, I. Inza, and J. A. Lozano, "A semi-supervised approach to multi-dimensional classification with application to sentiment analysis," in *Proceedings of the V Simposio de Teoria y Aplicaciones de Mineria de Datos (TAMIDA 2010)*, pp. 129–138, Valencia, Spain, Sep. 2010.

[139] J. Ortigosa-Hernández, J. Rodríguez, L. Alzate, I. Inza, and J. A. Lozano, "Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers," Tech. Rep. EHU-KZAA-TR-4-2011, University of the Basque Country UPV/EHU, San Sebastián, Spain, 2011.

[140] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, "On the optimal usage of labelled examples in semi-supervised multi-class classification problems," Tech. Rep. EHU-KZAA-TR;2015-01, University of the Basque Country UPV/EHU, San Sebastián, Spain, 2015.

[141] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, "Measuring the class-imbalance extent of multi-class problems," Tech. Rep. 0167-8655, BCAM: Basque Centre for Applied Mathematics, Bilbao, Spain, 2017.