UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e Computação

Marcos Medeiros Raimundo

Multi-objective optimization in machine learning

Otimização multiobjetivo em aprendizado de máquina

Campinas

2018

Marcos Medeiros Raimundo

Multi-objective optimization in machine learning

Otimização multiobjetivo em aprendizado de máquina

Thesis presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering, in the area of Computer Engineering.

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Prof. Dr. Fernando José Von Zuben

Este exemplar corresponde à versão final da tese defendida pelo aluno Marcos Medeiros Raimundo, e orientada pelo Prof. Dr. Fernando José Von Zuben

---

Campinas

2018

Agência(s) de fomento e nº(s) de processo(s): FAPESP, 2014/13533-0
ORCID: https://orcid.org/0000-0003-0499-2564

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

R133m

Raimundo, Marcos Medeiros, 1988-
Multi-objective optimization in machine learning / Marcos Medeiros Raimundo. – Campinas, SP : [s.n.], 2018.

Orientador: Fernando José Von Zuben.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado de máquina. 2. Otimização multi-objetivo. 3. Reconhecimento de padrões. I. Von Zuben, Fernando José, 1968-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Otimização multi-objetivo em aprendizado de máquina
Palavras-chave em inglês:
Machine learning
Multi-objective optimization
Pattern classification systems
Área de concentração: Engenharia de Computação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Fernando José Von Zuben [Orientador]
Ricardo Hiroshi Caldeira Takahashi
Márcio Porto Basgalupp
Guilherme Palermo Coelho
Levy Boccato
Data de defesa: 12-12-2018
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA – TESE DE DOUTORADO

Candidato: Marcos Medeiros Raimundo RA: 071739

Data da defesa: 12/12/2018

Título da tese em inglês: Multi-objective optimization in machine learning
Título da tese: Otimização Multiobjetivo em Aprendizado de Máquina

Prof. Dr. Fernando José Von Zuben (Presidente, FEEC/UNICAMP)
Prof. Dr. Guilherme Palermo Coelho (FT/UNICAMP)
Prof. Dr. Levy Boccato (FEEC/UNICAMP)
Prof. Dr. Márcio Porto Basgalupp (UNIFESP)
Prof. Dr. Ricardo Hiroshi Caldeira Takahashi (DM/UFMG)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

To my family, my friends, and my mentors.

Acknowledgements

The doctoral research was composed of grateful times along which I fought side by side with awesome people, struggling every day to cross every personal limitation, hoping someday to achieve and (why not?) cross the knowledge frontier. Get involved in advanced research topics has never made part of my former dreams, when I was young, but I always received full support from my family to meet challenges that could fill my dreams, give me a purpose, and put me in a position in which I could devote the best of me to other people. I am really thankful for everything they taught me and gave me.

At the beginning of this journey, Prof. Fernando J. Von Zuben understood all of this and offered me wise words about strategic planning I should conceive to be successful in the academia. Those wise words gave me a direction every day during these last seven years (including the period as a Master student) and helped me to surpass the challenges. Fernando always pointed me out which battles I should fight, and joined me when I needed support. The advice, scolding and care to extract the best of me always directed me to find my best and inspired me to do the same for other people. Thank you, Fernando, for everything.

After walking this path, I learned that the academia can be a place of generosity, kindness, and carefulness, and this feeling revealed to me the importance of empathy and of a holistic view of my role in a research team. I thanks Prof. Christiano Lyra for welcoming me, and Profs. Romis Attux, Celso Cavellucci, Fabio Usberti, Paulo A. V. Ferreira and Fernando J. Von Zuben for reinforcing these values.

Even with so much support from my mentors, the doctoral experience would be much heavier without my friends from LBiC and Labore. We shared meals, laughs, memes, advises and troubles; oh, and works too; and I almost forgot about the technical, political, philosophical and foolish talks. I am glad to have participated and helped to build such a healthy, vivacious and cooperative environment.

I acknowledge and thank the Brazilian funding agencies FAPESP and CAPES for the doctoral scholarship, grant #2014/13533-0, São Paulo Research Foundation (FAPESP), that allowed me to develop my research under the best conditions. The research benefited from a

FAPESP/CAPES Ph.D. scholarship from September 2014 to August 2018.

"It was the best of times, it was the ~~worst~~ hardest of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of light, it was the season of darkness, it was the spring of hope, it was the winter of ~~despair~~ struggle."

Charles Dickens

Abstract

Regularized multinomial logistic regression, multi-label classification, and multi-task learning are examples of machine learning problems in which conflicting objectives, such as losses and regularization penalties, should be simultaneously minimized. Therefore, the narrow perspective of looking for the learning model with the best performance should be replaced by the proposition and further exploration of multiple efficient learning models, each one characterized by a distinct trade-off among the conflicting objectives. Committee machines and a posteriori preferences of the decision-maker may be implemented to properly explore this diverse set of efficient learning models toward performance improvement. The whole multi-objective framework for machine learning is supported by three stages: (1) The multi-objective modelling of each learning problem, explicitly highlighting the conflicting objectives involved; (2) Given the multi-objective formulation of the learning problem, for instance, considering loss functions and penalty terms as conflicting objective functions, efficient solutions well-distributed along the Pareto front are obtained by a deterministic and exact solver named NISE (Non-Inferior Set Estimation); (3) Those efficient learning models are then subject to a posteriori model selection, or to ensemble filtering and aggregation. Given that NISE is restricted to two objective functions, an extension for many objectives, named MONISE (Many Objective NISE), is also proposed here, being an additional contribution and expanding the applicability of the proposed framework. To properly access the merit of our multi-objective approach, more specific investigations were conducted, restricted to generalized linear learning models with regularization: (1) What is the relative merit of the a posteriori selection of a single learning model, among the ones produced by our proposal, when compared with other single-model approaches in the literature? (2) Is the diversity level of the learning models produced by our proposal higher than the diversity level achieved by alternative approaches devoted to generating multiple learning models? (3) What about the prediction quality of ensemble filtering and aggregation of the learning models produced by our proposal on: (*i*) multi-class classification, (*ii*) imbalanced classification, (*iii*) multi-label classification, (*iv*) multi-task learning, (*v*) multi-view learning? The deterministic nature of NISE and MONISE, their ability to properly deal with the shape of the Pareto front in each learning problem, and the guarantee of always obtaining efficient learning models are advocated here as being responsible for the promising results achieved in all those three specific investigations.

Keywords: Machine Learning; Multi-objective optimization; Ensemble learning, Pattern classification.

Resumo

Regressão logística multinomial regularizada, classificação multi-rótulo e aprendizado multi-tarefa são exemplos de problemas de aprendizado de máquina em que objetivos conflitantes, como funções de perda e penalidades que promovem regularização, devem ser simultaneamente minimizadas. Portanto, a perspectiva simplista de procurar o modelo de aprendizado com o melhor desempenho deve ser substituída pela proposição e subsequente exploração de múltiplos modelos de aprendizado eficientes, cada um caracterizado por um compromisso (*trade-off*) distinto entre os objetivos conflitantes. Comitês de máquinas e preferências a posteriori do tomador de decisão podem ser implementadas visando explorar adequadamente este conjunto diverso de modelos de aprendizado eficientes, em busca de melhoria de desempenho. A estrutura conceitual multi-objetivo para aprendizado de máquina é suportada por três etapas: (1) Modelagem multi-objetivo de cada problema de aprendizado, destacando explicitamente os objetivos conflitantes envolvidos; (2) Dada a formulação multi-objetivo do problema de aprendizado, por exemplo, considerando funções de perda e termos de penalização como objetivos conflitantes, soluções eficientes e bem distribuídas ao longo da fronteira de Pareto são obtidas por um *solver* determinístico e exato denominado NISE (do inglês *Non-Inferior Set Estimation*); (3) Esses modelos de aprendizado eficientes são então submetidos a um processo de seleção de modelos que opera com preferências a posteriori, ou a filtragem e agregação para a síntese de ensembles. Como o NISE é restrito a problemas de dois objetivos, uma extensão do NISE capaz de lidar com mais de dois objetivos, denominada MONISE (do inglês *Many-Objective* NISE), também é proposta aqui, sendo uma contribuição adicional que expande a aplicabilidade da estrutura conceitual proposta. Para atestar adequadamente o mérito da nossa abordagem multi-objetivo, foram realizadas investigações mais específicas, restritas à aprendizagem de modelos lineares generalizados com regularização: (1) Qual é o mérito relativo da seleção a posteriori de um único modelo de aprendizado, entre os produzidos pela nossa proposta, quando comparado com outras abordagens de modelo único na literatura? (2) O nível de diversidade dos modelos de aprendizado produzidos pela nossa proposta é superior àquele alcançado por abordagens alternativas dedicadas à geração de múltiplos modelos de aprendizado? (3) E quanto à qualidade de predição da filtragem e agregação dos modelos de aprendizado produzidos pela nossa proposta quando aplicados a: (*i*) classificação multi-classe, (*ii*) classificação desbalanceada, (*iii*) classificação multi-rótulo, (*iv*) aprendizado multi-tarefa, (*v*) aprendizado com multiplos conjuntos de atributos? A natureza determinística de NISE e MONISE, sua capacidade de lidar adequadamente com a forma da fronteira de Pareto em cada problema de aprendizado, e a garantia de sempre obter modelos de aprendizado eficientes são aqui pleiteados como responsáveis pelos resultados promissores alcançados em todas essas três frentes de investigação específicas.

Keywords: Aprendizado de Máquina; Otimização Multi-objetivo; Aprendizado por ensembles, Classificação de Padrões.

List of Figures

List of Tables

List of Symbols

Symbol	Meaning
Multi-objective optimization symbols	
m	objective space dimension
n	decision space dimension
$\mathbf{w} \in [0,1]^m$	weighted sum method's weighting vector
$\mathbf{x} \in \mathbb{R}^n$	decision space variable
$\mathbf{r} \in \mathbb{R}^m$	objective space variable
$\underline{\mathbf{r}}, \overline{\mathbf{r}} \in \mathbb{R}^m$	outer and inner approximation of the Pareto front for NISE and MONISE methods
$f(\cdot) \in \mathbb{R}$, $\mathbf{f}(\cdot) \in \mathbb{R}^m$	objective function and vector of objective functions
Machine learning variables	
N	number of samples
d	number of features
K	number of classes
L	number of labels
T	number of tasks
V	number of views or groups
$\mathbf{x} \in \mathbb{R}^d$	sample input
$\mathbf{y} \in \{0,1\}^K$	sample output
$\boldsymbol{\theta} \in \mathbb{R}^d$	vector of parameters
$f(\cdot)$	generic objective function
$l(\cdot)$	generic learning loss function
$r(\cdot)$	generic regularization function

Contents

Chapter 1

Introduction

The design of high-performance learning machines usually involves decision making under conflicting objectives. Let us take the example of medical diagnosis systems: due to the scarcity of information, if a learning machine focus on the patterns of healthy patients it can incorrectly diagnose sick patients (false negative), and vice-versa (false positive). When we design medical diagnosis systems, we rarely produce learning machines with no incorrect diagnosis, but we can usually tune the machine to promote lower false positive or false negative rates.

Given that, a system that creates distinct classifiers capable of capturing the patterns of healthy and sick patients, with distinct false positive and false negative rates, can be very beneficial, because the diversity of behavior of the obtained classifiers may promote better generalization. A posteriori multi-objective methods are capable of finding a set of efficient classifiers exhibiting a wide range of trade-offs between false positive and false negative rates.

The main purpose of this thesis is to further explore the multi-objective nature of the learning problem, by properly: (1) specifying the conflicting objectives in an effective mathematical formulation; (2) solving the resulting multi-objective optimization problem with an existing deterministic solver characterized by consistently spreading the candidate solutions (learning models) along the Pareto front; (3) extending that multi-objective solver to deal with more than two conflicting objectives, thus expanding the applicability of the methodology; (4) exploring the obtained efficient learning models using a posteriori selection of a single learning model or even aggregating multiple efficient learning models in an ensemble, thus taking full advantage of the distinct trade-offs among the learning models.

Theoretical aspects of machine learning and multi-objective optimization are introduced in what follows, evidencing the mutual relationship of both areas and thus creating

favourable conditions to support the main proposal.

1.1 Basics of machine learning

In psychology, learning is defined as a permanent change in the behaviour of the decision maker based on past experience (GROSS, 2010). Embedding the learning procedure on computers, without human interference, is the main objective of machine learning (MITCHELL, 1997). Machine learning is gaining more and more attention due to the increasing availability of data in the information age (HASTIE *et al.*, 2009), and learning from data usually occurs exploring one of two approaches (HASTIE *et al.*, 2009): (*i*) supervised learning, that uses the gathered data and some expert knowledge (used to label the data) to build a system capable of predicting new outcomes; this also allows human insights on the patterns that explain the expert knowledge; (*ii*) unsupervised learning, that aims to find the patterns that describe the statistical distribution of the data, searching for a model that explains its generation.

More formally, given an input $\mathbf{x}$ and an output y, supervised learning consists in finding a map function $f(\cdot)$ that provides the relation between input and output ($f(\mathbf{x}) \approx y$). In many methodologies, such as linear methods, support vector machines (SVM) and neural networks, it is defined a loss function $l(\cdot)$ that measures the dissimilarity between the function $f(\mathbf{x}, \boldsymbol{\theta})$ and the output y, given a parameter vector $\boldsymbol{\theta}$. In supervised learning, the two main problems are: (*i*) regression: in which we want to predict a real value $y \in \mathbb{R}$, and (*ii*) classification: in which we want to predict the membership $y^k \in \{0, 1\}$ to a given class k.

Taking the example of binary classification (where we want to find the membership $y \in \{0, 1\}$ to a single class), a suitable mapping function consists in measuring the probability of membership $f(\mathbf{x}, \boldsymbol{\theta}) \in [0, 1]$ to that class. A good loss function would measure how far the probability is from the correct assignment (0 or 1). However, it is still necessary to adjust the parameter vector $\boldsymbol{\theta}$ to reduce the learning loss $l(\cdot)$, which can be accomplished by optimization procedures.

It is important to notice the necessity of designing a robust procedure that is capable of finding the parameter vector $\boldsymbol{\theta}$ which guides to the minimization of the learning loss. To illustrate this challenge, Figure 1 shows the loss $l(\cdot)$ for different parameter vectors $\boldsymbol{\theta}$ (hyperplanes that separate samples belonging to one class from samples belonging to another class), represented here as lines of different colors, and the goal is represented by the orange line.

However, the selection of the learning model with minimal loss might be misguiding.

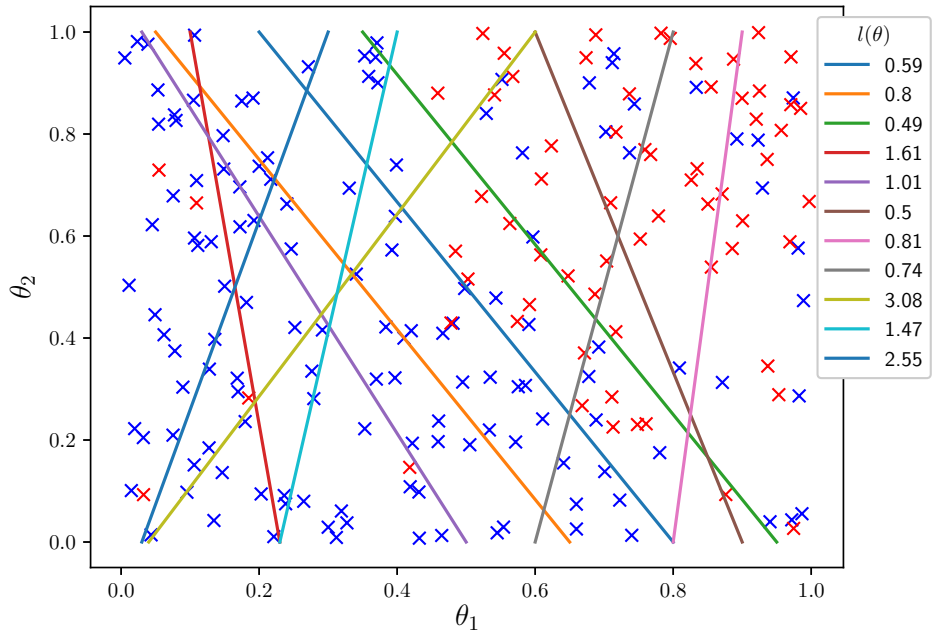


Figure 1 – Representation of multiple binary classifiers with their correspondent logistic loss. Scenario with more miss-labelled samples from the red class than from the blue class.

Figure 1 represents a scenario where there are more miss-labelled samples from the red class than from the blue class. A more realistic representation is shown in Figure 2, that represents the same distribution but with balanced miss-labelled samples. In this case, the classifier represented by the blue line is the most suitable classifier.

This reasoning highlights the necessity of dealing with two challenges: (1) how to direct the optimization procedure, and (2) how to exploit distinct models. The first challenge is called model selection or hyper-parameter tuning, which consists in selecting the (hyper-)parameters that guide the optimization (or learning) procedure; the second challenge may be faced by committee machines, in the form of an ensemble of learning models, and consists in properly combining the prediction of multiple classifiers, that was generated by exploring distinct perspectives of the problem.

The central concept for model selection relies on cross-validation. This procedure can be done by splitting the data sampling into distinct sets: training and validation. The first one is used to adjust the model (for example by optimization), represented in Figure 1; and the second one is used to select the model, fine-tune the parameters, or help another meta-learner (such as an ensemble) to build a learning machine, represented in Figure 2. We can see that, even if the classifiers were designed in Figure 1, a model selection was also able to mitigate

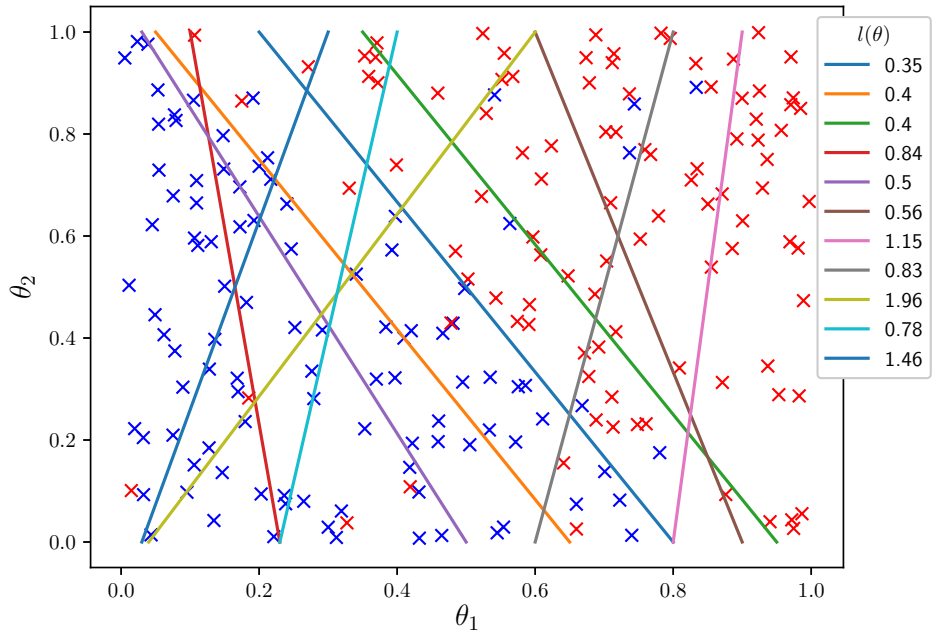


Figure 2 – Representation of multiple binary classifiers with their correspondent logistic loss. Scenario with balanced miss-labelled samples.

the miss-classification rate if we select the best model based on the scores of Figure 2. Even if we are not confident of selecting a single classifier, it is also possible to build an ensemble by using, for instance, the three best classifiers (blue, orange, and red), also achieving a good classification performance.

Besides training and validation, it is also important to separate a test set. This set is used to report the expected performance in practical applications. Training, validation and test sets should be independent among each other, and each one should be representative of the expected input-output behavior of the system being learned.

1.2 Basics of multi-objective optimization

Suppose you want to buy a CPU unit that has to be simultaneously cheap and powerful. We know the impossibility of achieving both criteria at the same time, and normally we end up with searching for a good **cost-benefit** option. When this selection is well executed, our choice is such that there is no other CPU, at the same time, cheaper and more powerful than the chosen one. This desired situation is called **Pareto-efficiency**, a concept developed by Vilfredo Pareto in his book “Manual of Political Economy”, 1906.

The problem with **cost-benefit** approaches involves some explicit or implicit preferences, for example, expense constraints. When you fix a priori a maximum spending limit, you might refuse a significantly more powerful CPU even if it is marginally more expensive than the limit. To help finding a suitable CPU, a posteriori decision making will benefit from the most representative trade-offs, also called Pareto-optimal candidate solutions. Indeed, dominated solutions (solutions that are not part of the Pareto-optimal set) will not interfere in the decision making process, and a posteriori preferences will be applied in a more consolidated scenario. A real-world example is presented in Figure 3 in which all dominated CPUs (with at least one option less expensive and with better benchmark) are displayed in grey and the Pareto-optimal CPU options are displayed in black.

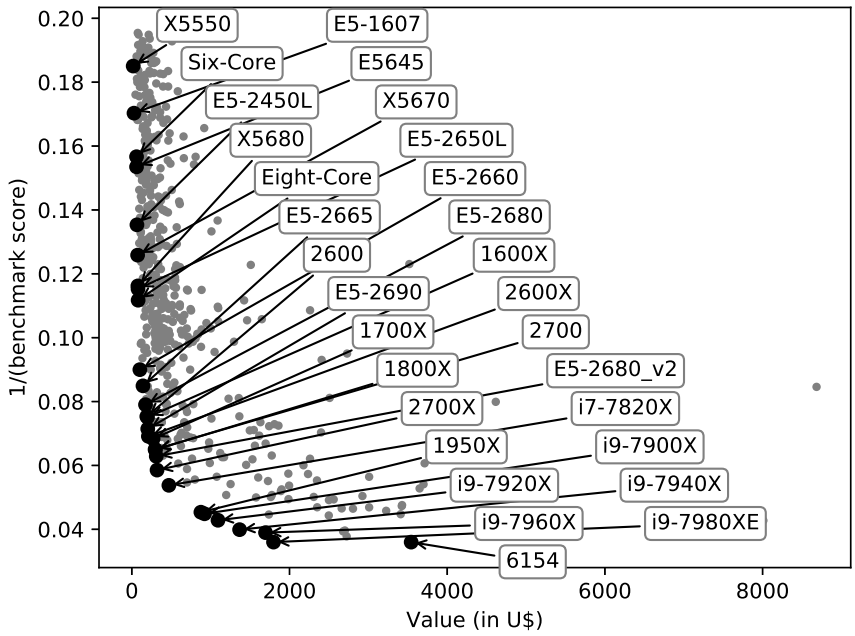


Figure 3 – Multi-objective representation of the High-End CPUs with only 29 Pareto-optimal options in 485 CPUs. Values and benchmark scores available at <cpubenchmark.net>.

In **machine learning**, the use of **multi-objective optimization** can surely be considered a comprehensive approach, being used to simultaneously find interpretable and accurate models, to generate multiple efficient models endowed with complementary properties, and also to create ensembles using conflicting objectives (JIN; SENDHOFF, 2008). Despite that, generally a single Pareto-optimal solution is achieved, by imposing a priori a relative relevance among the multiple objectives, thus guiding to a single-objective criterion.

An example of this negligence can be seen in a multinomial regression (further explored and explained in the text) expressed in Equation (1.1), where the loss of all classes are

considered of the same level of relevance:

$$\min_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K}\left[\sum_{i=1}^{N} -y_i^k \ln\left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}\right)\right] \equiv \sum_{k=1}^{K} l_k(\boldsymbol{\theta}). \tag{1.1}$$

where $\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}$ is the probability that the model assigns class k to sample i, $y_i^k = 1$ if sample i is originally assigned to class k and $y_i^k = 0$ otherwise, thus $-y_i^k \ln\left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}\right)$ indicates the learning error of sample i associated with class k

In this case, it is possible to suppose that the learning error in each class, expressed by $l_k(\boldsymbol{\theta})$, might be conflicting. This occurs because, to correctly classify more samples from one class, it may induce an increase in the miss-classification rate of other classes. Given that, Equation (1.1) implicitly takes a "flat" a priori preference among classes, and the optimization method will search the model with the lowest aggregate loss. As a consequence: (i) this aggregate loss will indirectly improve the classification accuracy of the majority classes because they have more samples in cases of data imbalance; and (ii) it may also not fulfill the expectation of the user in some sensible scenarios such as in medical cases, where wrongly classifying a case as a disease may not be as crucial as sending a sick patient home.

Given that, we can see that even with a good optimization method, this optimization model might guide to a misleading classifier in more complex cases, reinforcing relevance of multi-objective approaches. Multi-objective optimization methods can help in this task, and this thesis wants to show improvements when dealing with a variety of machine learning problems.

1.3 Using conflicts to deal with machine learning problems

The proposal of this thesis is a unified framework that adapts and constructs generalized linear learning models to solve an assortment of classification problems. This framework relies on 3 steps:

1. **Multi-objective modelling**: conflicting objectives are explicitly formalized and aggregated in a multi-objective formulation;

2. **Multi-objective training**: using a model with highlighted conflicting objectives, it is possible to use multi-objective optimization methods to find Pareto-optimal solutions.

In this work we rely on a posteriori methods, such as NISE (COHON *et al.*, 1979) and MONISE (RAIMUNDO; VON ZUBEN, 2017), to find a set of solutions acting as a good representation of all possible trade-offs between those objectives;

3. **A posteriori decision making**: since we build a set of solutions, we can select the best possible solution, or we can use these efficient candidate learning models as ensemble components, exploring the literature of ensembles to aggregate these models toward better performance.

The core of the proposal consists in exploring the Pareto-optimal solutions. To do this, it is necessary to model the problem in such a way that, when optimized, it generates trade-off solutions that can be useful to the problem. For example, in an L_1 regularized logistic regression, we do not know a priori which is the correct regularization constraint to find the best model in a real-world scenario. Given this, a diverse set of trade-offs between the classification loss and the L_1 strength can be obtained. Multi-class imbalanced classification is another scenario where it is not known how much importance to give to each class. When the learning machine gives too much importance to a class with scarce samples, it can result in a poor classifier if a good portion of those samples are noisy; and when the machine gives too little importance, it can result in a biased classifier which is focused only on the majority classes.

Our approach attacks those problems on two fronts: (1) giving more flexibility to the models, allowing training with distinct priorities for the losses and penalties; (2) multi-objective training, which generates a set of solutions that is a parsimonious representation of all possible trade-offs between the objectives. Considering this set of solutions, the primary objective of this work is to study the benefits of these Pareto-optimal solutions resorting to a posteriori preferences.

One primary aspect of the methods NISE and MONISE involves their necessity of using the weighted sum method, a scalarization procedure, as an auxiliary step of the method. Methods founded on scalarization are only capable of finding any member of the Pareto front for convex problems. Therefore, with convex Pareto fronts, NISE and MONISE are very good at properly sampling the Pareto front, in a deterministic and systematic way. With nonconvex Pareto fronts, there is no a priori control of the quality level of the sampling process. That is why we have concentrated our research in convex machine learning problems, involving learning models which are linear in the adjustable parameters. Moreover, this kind of learning models are known to be scalable and usually quite competent in solving machine learning problems. Additionally, ensemble methods, responsible for aggregating multiples learning

models in a single solution, mitigate the potential limitation of single learning models which are linear in the adjustable parameters.

Using a set of experiments, we deeply investigate many contexts of machine learning, thus raising characteristics and relevance of multi-objective optimization in every scenario, allowing us to make assumptions to the general field. In multi-class classification, we not only investigate the quality of the final classifier, but also investigate the merit of the multi-objective optimization sampling regarding model selection, and how much diversity is generated by this methodology. Exploring a model where every class loss is conflicting, we also investigate the relevance of allowing distinct weights for every class loss with weights being automatically determined to promote diverse sampling of the Pareto front. Another important field of machine learning is the knowledge transferring field (mainly transfer learning and multi-task learning). In this field, we investigate the importance of multi-objective trained models in multi-label classification, transfer learning in the detection of epileptic seizures and multi-task learning. In this last problem, we also investigate how crucial is the insertion of new ensemble candidates increase the quality of the prediction as well as the knowledge transfer relations that occur in our framework.

1.4 Organization of the manuscript

The content of this manuscript is organized as follows:

Chapter 2 introduces the main concepts of multi-objective optimization, further explaining the weighted sum method, the NISE multi-objective solver (COHON *et al.*, 1979) and also introducing MONISE (RAIMUNDO; VON ZUBEN, 2017) algorithm, which is an original contribution of the research. These algorithms are supported by the weighted sum method, which turns to be a very convenient formulation for multi-objective problems in machine learning.

Chapter 3 introduces the machine learning problems explored in this thesis. This chapter further explains the statistical background of the learning models, resulting in a weighted sum of the conflicting objectives. In other words, the machine learning problems are explicitly formulated as multi-objective optimization problems which admit a direct manipulation by solvers founded on scalarization.

Chapter 4 presents the whole framework of our proposal. This is achieved by explaining the following three steps: how the models are adapted to take advantage of the multi-

objective approach, how to optimize these models using NISE and MONISE, and how to use the set of models to construct a coherent and robust learning machine.

Chapter 5 summarizes the most important methodologies which exhibits a close relationship with the multi-objective approach in this work.

Chapter 6 presents a set of experiments to assess the merit of out proposal. These experiments comprehend 4 types of learning problems: multi-class classification with 121 datasets; detection of epileptic seizures with 17 patients; multi-label classification with 6 datasets; and multi-task learning with 15 synthetic and three real-world datasets. The experiments are independent of each other, comparing our methodology with the best models of every specific field, and resorting to consolidated metrics to evaluate the methodologies.

Chapter 7 outlines the concluding remarks and directions for further investigations.

1.5 Publications and already submitted manuscripts

During the development of this PhD research at the Laboratory of Bioinformatics and Bio-inspired Computing (LBiC), DCA/FEEC/UNICAMP the papers listed in what follows were conceived. Many of them are directly related to the contributions of this research.

- Raimundo, M.M.; Marques, A.C.R.; Drumond, T.; Rocha, A.; Lyra, C.; and Von Zuben, F.J.; "Exploring multiobjective training in multiclass classification". IEEE Transactions on Neural Networks and Learning Systems (submitted).
- Raimundo, M.M.; Ferreira, P.A.V.; and Von Zuben, F.J.; "An Extension of the Non-Inferior Set Estimation Algorithm for Many Objectives". European Journal of Operations Research (submitted).
- Raimundo, M.M.; and Von Zuben, F.J.; "Investigating multiobjective methods in multitask classification". 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018, pp. 1-9.
- Beserra, F.S.; Raimundo, M.M.; and Von Zuben, F.J.; (2018) "Ensembles of Multi-objective-Based Classifiers for Detection of Epileptic Seizures". In: Mendoza M., Velastín S. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017. Lecture Notes in Computer Science, vol 10657. Springer.

- Raimundo, M.M.; and Von Zuben, F.J.; (2018) "Many-Objective Ensemble-Based Multilabel Classification". In: Mendoza M., Velastín S. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017. Lecture Notes in Computer Science, vol 10657. Springer.

- Marques, A.C.R.; Raimundo, M.M.; Cavalheiro, E.M.B.; Salles, L.F.P.; Lyra, C.; and Von Zuben, F.J.; (2018) "Ant genera identification using an ensemble of convolutional neural networks". PLoS ONE 13(1): e0192011.

- Beserra, F.S.; Raimundo, M.M.; and Von Zuben, F.J.; "Multi-objective transfer learning for epileptic seizure detection" Journal of Epilepsy and Clinical Neurophysiology v. 23(2): 37-72, p. 67 (2017)

Chapter 2

Multi-objective optimization[1]

Multi-objective optimization is a class of problems in mathematical programming whose main characteristic is the existence of multiple, potentially conflicting, objective functions. The main challenge of multi-objective optimization is to simultaneously deal with conflicting objectives and be able to express the user's preference concerning these objectives.

Definition 2.1. *A multi-objective problem is defined as follows (MIETTINEN, 1999; MARLER; ARORA, 2004):*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) \equiv \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\} \\ \textit{subject to} \quad & \mathbf{x} \in \Omega, \Omega \subset \mathbb{R}^n \\ & \mathbf{f}(\cdot) : \Omega \rightarrow \Psi, \Psi \subset \mathbb{R}^m \end{aligned}$$

In Definition 2.1, the set $\Omega \subset \mathbb{R}^n$ is known as the decision space and $\Psi \subset \mathbb{R}^m$ is known as the objective space. Figure 4 represents the established relation between those two spaces (restricted to two dimensions for visualization purposes). Each point at the decision space has a correspondent point at the objective space, obtained by evaluating each objective function. On the objective space, the two bold lines correspond to the Pareto front, which is the set of all efficient or non-inferior solutions.

The symbol "min" in Definition 2.1 means searching for minimal solutions in a partial ordering (WIECEK *et al.*, 2016). Since the objective space is multidimensional, two solutions

[1]This chapter is based on both MONISE - Many Objective Non-Inferior Set Estimation (RAIMUNDO; VON ZUBEN, 2017) and Exploring multi-objective training in multiclass classification (©2018 IEEE) (RAIMUNDO *et al.*, 2018)

only have a relation of order (dominance relation) when the worse solution has, with respect to a better solution, all objectives of equal or lower quality and at least one objective strictly of lower quality (WIECEK *et al.*, 2016). The solutions not dominated by any other feasible solution are called Pareto-optimal solutions (also called efficient solution, further defined in Definition 2.2) (MIETTINEN, 1999; WIECEK *et al.*, 2016). In the absence of preferences, those solutions correspond to distinct trade-offs of the objectives.

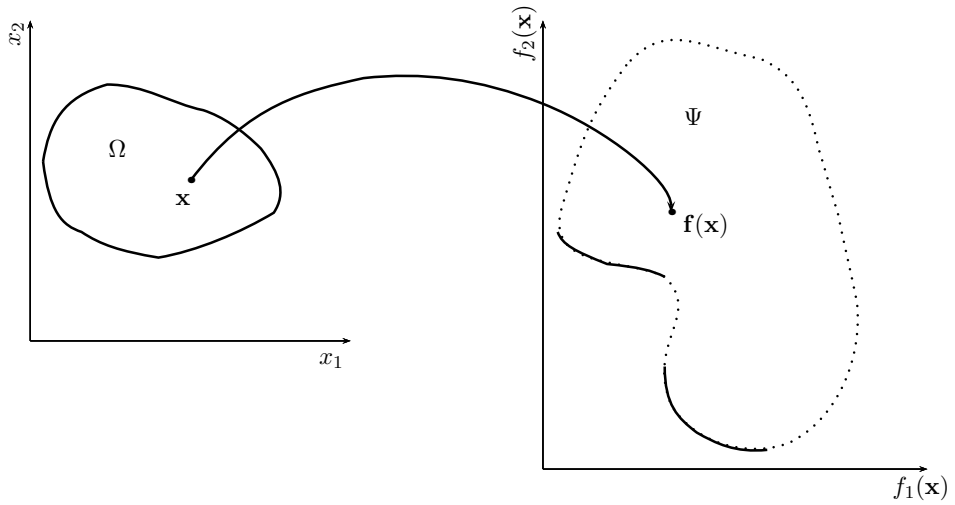


Figure 4 – Representation of the decision space (on the left) and the objective space (on the right) taking two decision variables and two objectives.

In the sequence, based on the formalism provide by Miettinen (1999), Marler & Arora (2004), Raimundo & Von Zuben (2017), we present some basic definitions to contextualize the multi-objective optimization problem. Without loss of generality, the objectives are associated with minimization problems.

Definition 2.2. ***Efficiency/Pareto-optimality****: A solution* $\mathbf{x}^* \in \Omega$ *is efficient (Pareto-optimal) if there is no other solution* $\mathbf{x} \in \Omega$ *such that* $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$, $\forall i \in \{1, 2, \ldots, m\}$ *and* $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ *for some* $i \in \{1, 2, \ldots, m\}$.

Definition 2.3. ***Efficient front/Pareto front****: An efficient front* Ψ^* *(Pareto front) is the set of all efficient solutions. When considered the problem on Definition 2.1, the efficient front* Ψ^* *is formed by efficient objective vectors* $\mathbf{f}(\mathbf{x}^*) \in \Psi^*$ *which has a corresponding feasible solution* $\mathbf{x}^* \in \Omega$. *Also,* Ω^* *is the Pareto-optimal set whose objective vectors are into the efficient front:* $\mathbf{x}^* \in \Omega^* \Leftrightarrow \mathbf{f}(\mathbf{x}^*) \in \Psi^*$.

The following definitions are necessary to support the proposition of some of the adaptive and scalarization methods. The "k-th definitions" are intended to refer to single objective solutions.

Definition 2.4. ***k-th individual minimum value****: When only the k-th component of the objective function vector is optimized, the solution* $\mathbf{x}^{*(k)}$ *is obtained. The k-th individual minimum value* $l^{(k)}$ *corresponds to the minimum value of the optimization* ($l^{(k)} = f_k(\mathbf{x}^{*(k)})$).

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_k(\mathbf{x}) \\ \textit{subject to} \quad & \mathbf{x} \in \Omega,\ \Omega \subset \mathbb{R}^n \\ & \mathbf{f}(\cdot) : \Omega \rightarrow \Psi, \Psi \subset \mathbb{R}^m \end{aligned}$$

Definition 2.5. ***k-th individual minimum solution****: An individual minimum solution* $\mathbf{l}^{*(k)}$ *is an efficient solution characterized by having its k-th component equal to the k-th individual minimum value* $l^{(k)}$.

Definition 2.6. ***Utopian solution****: A utopian solution* $\mathbf{r}^{utopian}$ *is a vector on the objective space characterized by having its k-th component* $r_k^{utopian}$ *given by the k-th individual minimum value* $l^{(k)}$ *(see Definition 2.4), and this is valid for all* $k \in \{1, 2, \ldots, m\}$*:*

$$\mathbf{r}^{utopian} = \{l^{(1)}; \ldots; l^{(m)}\}$$

It is important to notice that, if utopian solution is attainable it would dominate any other solutions, being possible to assume that objectives are not conflicting.

2.1 Weighted sum method

The weighted sum method consists in optimizing a convex combination of the objectives, with each component of the weight vector representing a relative importance of the corresponding objectives. With this scalarization, the designer expresses his/her preference (COHON, 1978). Additionally, as will be done in this work, the weighted vector may be automatically determined by a recursive process, aiming at exploring particular regions of the Pareto front.

Definition 2.7. *The definition of the weighted sum method is given by:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{w}^\top \mathbf{f}(\mathbf{x}) \\ \textit{subject to} \quad & \mathbf{x} \in \Omega,\ \Omega \subset \mathbb{R}^n, \\ & \mathbf{f}(\cdot) : \Omega \rightarrow \Psi, \Psi \subset \mathbb{R}^m. \end{aligned}$$

where $\sum_{i=1}^{m} w_i = 1,\ \mathbf{w} \in \mathbb{R}^m$ *and* $w_i \geq 0\ \forall i \in \{1, 2, \ldots, m\}$ *is the parameter of the scalarization.*

In Figure 5, the weight vector $\mathbf{w}$ defines the slope of the line that guides the optimization process, reaching a tangent point in the objective space.

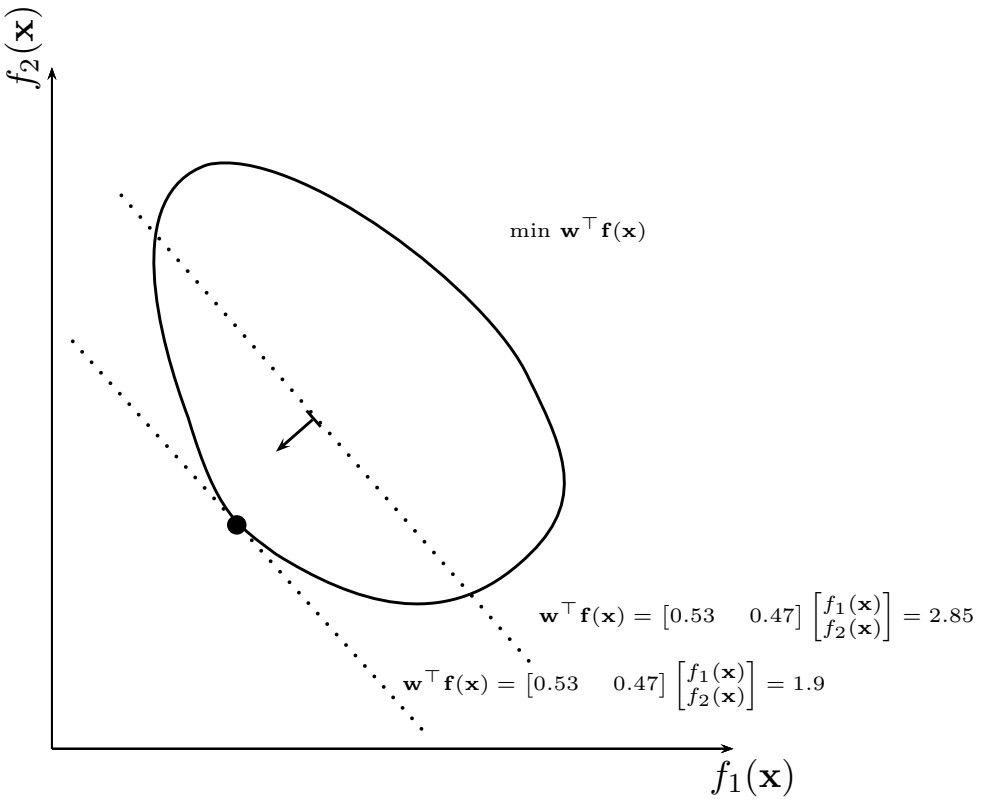


Figure 5 – Representation of the solution produced by the weighted sum method.

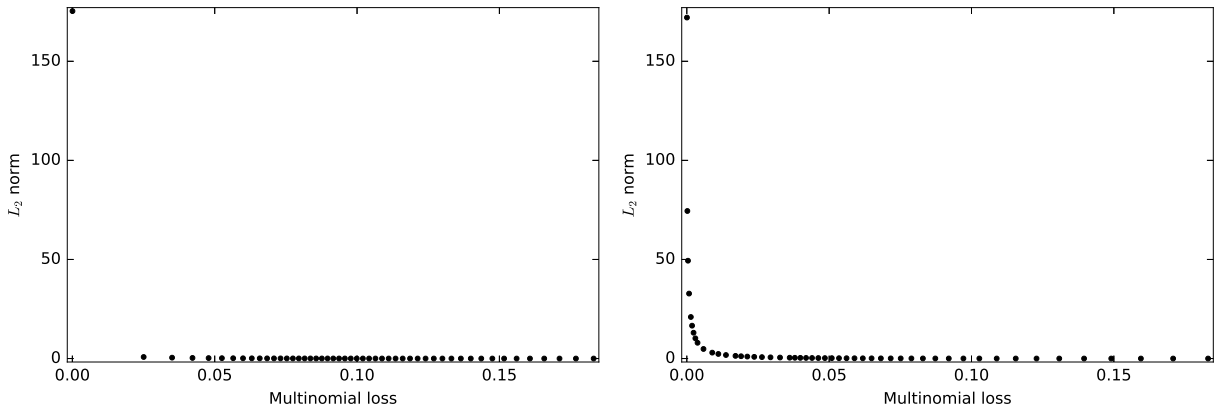
Some properties of this scalarization are relevant. In the general case, without assuming any particularity of the objective space $\mathbf{\Psi}$, an optimal solution for weighted sum method results in a efficient solution (this sufficient condition is proved in Geoffrion (1968) and Miettinen (1999)). Now, all efficient solutions are only attained by the weighted sum method if the problem is convex (this necessary condition is proved in Miettinen (1999)). Despite that, all efficient solutions which are dominated by a convex combination of other efficient solutions are not attained by the weighted sum method (non-necessary condition is proved in Koski (1985) and Das & Dennis (1997)), which means that there is no weight vector $\mathbf{w}$ capable of conducting the weighted sum method to find an efficient solution $\overline{\mathbf{x}}$ whose objective vector $\mathbf{f}(\overline{\mathbf{x}})$ is dominated by a convex combination of other efficient objective vectors, making the weighted sum method incapable of finding solutions in the so-called "concave" parts of Pareto fronts.

Taking the example of regularized multinomial logistic regression in machine learning

(further explained in in Sections 3.3 and 3.4):

$$\min_{\theta} \quad l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}) \equiv \sum_{i=1}^{N} \sum_{k=1}^{K} - \left[y_i^k \ln \left(\frac{e^{\theta_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\theta_k^\top \phi(\mathbf{x}_i)}} \right) \right] + \lambda ||\boldsymbol{\theta}||. \tag{2.1}$$

A naïve procedure to create a sampling of the Pareto front for regularized multinomial regression using the weighted method consists in creating a grid sampling $w_2 \equiv \frac{\lambda}{1+\lambda}$ from 0 to 1 in a fixed step (thus sampling $w_1 \equiv \frac{\lambda}{1+\lambda}$ as $w_1 = 1 - w_2$), such as using a step of 0.05 to generate samples of the Pareto front. However this procedure does not take into account the topology of the front, and also ignores that the mapping $\mathbf{w} \Rightarrow \mathbf{f}(\mathbf{x}^*)$ may not keep the uniformity of the Pareto front sampling. The application of this method is exemplified in Figure 6-a; when compared to Figure 6-b (produced by NISE), it clearly leads to a lower-quality exploration of the front. The main difficulty with this approach is related to the slope sensitivity (JUBRIL, 2012). Defining the slope $\tan(\phi) = \frac{w_2}{w_1} = \lambda$ of the parameter vector $\mathbf{w}$, it can be seen that changing w_2 from 0.95 to 1 leads to a corresponding change in the slope from 19 to ∞, missing all possible values of slope in this large interval.



(a) Grid Search (b) NISE

Figure 6 – Pareto front of logistic error vs L_2 norm of the parameter vector for the well-known Iris dataset. (©2018 IEEE)

Since there is no prior knowledge on the shape of the Pareto front, a priori methods (e.g. grid search) are not a reasonable choice. To overcome this issue, we leverage an adaptive method known as Non-Inferior Set Estimation (NISE) and its generalization known as Many objective NISE (MONISE).

2.2 NISE

The NISE (Noninferior Set Estimation) method (COHON, 1978) is an iterative approach that uses the weighted sum method to automatically create, at the same time, a representation and a relaxation of the Pareto front using a linear approximation. At every iteration, based on the already calculated efficient solutions, a line is traced connecting each neighboring pair of solutions, determining new weight vectors. This procedure finds an accurate and fast approximation for problems with two objectives (ROMERO; REHMAN, 2003).

Two neighboring efficient solutions (called neighborhood) are used to determine a new efficient solution employing the weighted sum method. More deeply explained: the initialization should generate the first two solutions (Section 2.2.1); at each iteration: the next neighborhood to be explored should be determined (Section 2.2.2), thus obtaining the parameters for the weighted sum method (Section 2.2.3), and a new solution, along with the next neighborhood (Section 2.2.4); the stopping criterion is defined to ensure a quality threshold of the approximation (Section 2.2.5).

2.2.1 Initialization

The initialization consists of finding the first two solutions $\mathbf{f}(\mathbf{x}^1)$ and $\mathbf{f}(\mathbf{x}^2)$ which are individual minimum solutions (Definition 2.5) for objectives 1 and 2, respectively. The weight vectors $\mathbf{w}^1$ and $\mathbf{w}^2$ have null elements except for the element corresponding to the objective being optimized, assumed to be equal to one. Finally, it is possible to define the first neighborhood $\mathcal{N}^1 = \{(1,2)\}$, containing the indexes of the first solutions 1 and 2, which will be used to find the subsequent solutions.

2.2.2 Neighborhood choice

Considering a set of neighborhoods $\mathcal{N}^k$, the neighborhood to be explored at the k-th iteration is the neighborhood that has the maximum error $\mu = \max\ \mu^{i,j}, \forall (i,j) \in \mathcal{N}$. To find that error, for every neighborhood $(i,j) \in \mathcal{N}$ it is possible to calculate the larger distance $\mu^{i,j}$ (defined in Equation (2.2)) between the normal vector $\mathbf{w}$ (calculated as described in Section 2.2.3) of the line that contains the solutions $\mathbf{f}(\mathbf{x}^i)$ and $\mathbf{f}(\mathbf{x}^j)$ and the intersection point between the solution hyperplanes ($\mathbf{w}^{i\top}\underline{\mathbf{r}} = \mathbf{w}^{i\top}\mathbf{f}(\mathbf{x}^i)$ and $\mathbf{w}^{j\top}\underline{\mathbf{r}} = \mathbf{w}^{j\top}\mathbf{f}(\mathbf{x}^i)$) of the neighborhood, thus

producing:

$$\mu^{i,j} = \sqrt{\frac{(\mathbf{w}^\top \mathbf{f}(\mathbf{x}^i) - \mathbf{w}^\top \underline{\mathbf{r}})^2}{||\mathbf{w}||^2}}. \tag{2.2}$$

In Figure 7, a geometrical view is depicted to help the comprehension of the steps involved. Vectors $\mathbf{w}^i$, $\mathbf{w}^j$ indicate the weight vectors used to find the solutions $\mathbf{f}(\mathbf{x}^i)$ and $\mathbf{f}(\mathbf{x}^j)$, respectively. Then, in the intersection of ${\mathbf{w}^i}^\top \mathbf{r} = {\mathbf{w}^i}^\top \mathbf{f}(\mathbf{x}^i)$ and ${\mathbf{w}^j}^\top \mathbf{r} = {\mathbf{w}^i}^\top \mathbf{f}(\mathbf{x}^j)$, it is obtained $\underline{\mathbf{r}}$, leading to the distance μ between $\underline{\mathbf{r}}$ and $\mathbf{w}^\top \mathbf{r} = \mathbf{w}^\top \mathbf{f}(\mathbf{x}^j)$ produced by Equation (2.2).

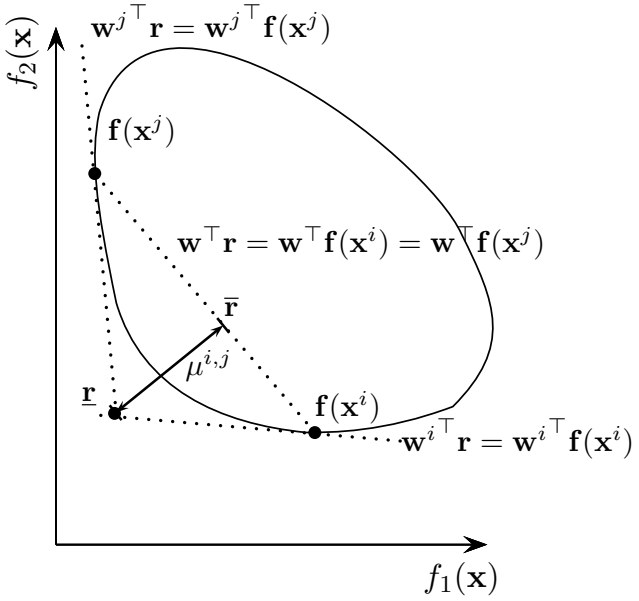


Figure 7 – Geometrical view of the current representation and relaxation of the Pareto front

2.2.3 Calculation of the scalarization weight vector

Given the neighborhood (i, j) composed of two efficient solutions $\{\mathbf{f}(\mathbf{x}^i), \mathbf{f}(\mathbf{x}^j)\}$, it is possible to calculate the unitary normal vector $\mathbf{w}$ of the line containing these points, using the following linear system:

$$\begin{cases} \mathbf{w}^\top \mathbf{f}(\mathbf{x}^i) = b \\ \mathbf{w}^\top \mathbf{f}(\mathbf{x}^j) = b \\ \mathbf{w}^\top \mathbf{1} = 1 \end{cases} \tag{2.3}$$

Considering the weight vector $\mathbf{w}$ with maximum error μ at the k-th iteration, it is possible to solve the weighted sum method (see Definition 2.7) and find $\mathbf{f}(\mathbf{x}^k)$.

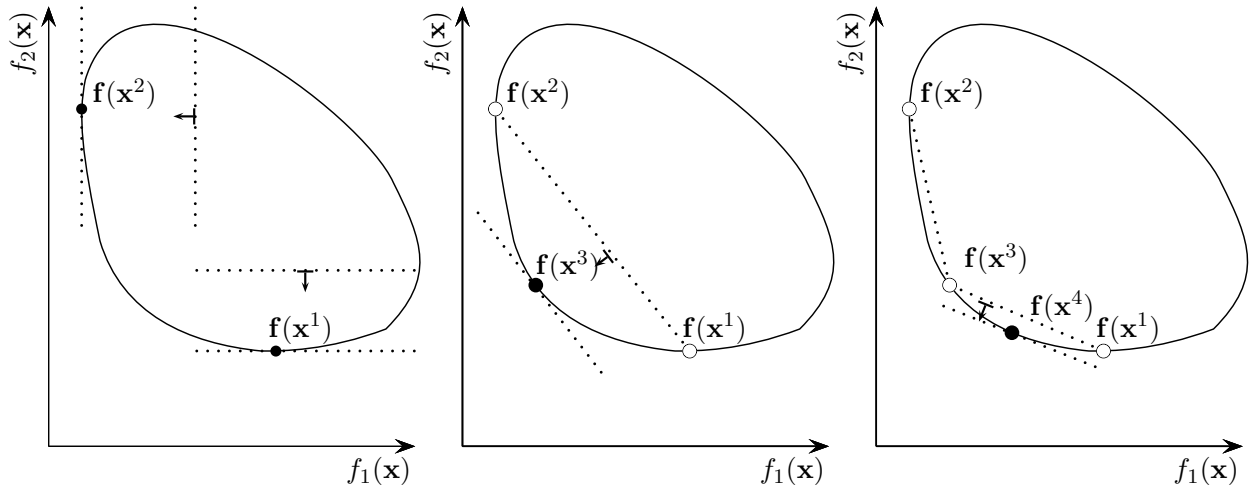
2.2.4 Updating new neighborhoods

Given that it was found a solution $\mathbf{f}(\mathbf{x}^k)$ associated with the current neighborhood, the new neighborhoods (i,k) and (k,j) are added to $\mathcal{N}^k$ and the previous neighbohood (i,j) is deleted, resulting in $\mathcal{N}^{k+1} = \mathcal{N}^k \cup \{(i,k),(k,j)\} \setminus (i,j)$.

2.2.5 Stopping criterion

The stopping criterion is fulfilled when the largest estimation error μ, defined in Section 2.2.2, is smaller than the threshold error μ^{stop}, or the desired number of efficient solutions is achieved.

2.2.6 Discussion



(a) Initialization (b) First iteration (c) Second iteration

Figure 8 – Illustrative sequence of steps of the NISE method.

A brief illustrative example of the execution of the method is shown in Figure 8. The initialization is represented in Figure 8-a, with the determination of the extreme solutions of the problem ($\mathbf{f}(\mathbf{x}^1)$ and $\mathbf{f}(\mathbf{x}^2)$). In Figure 8-b, the unitary normal vector of the segment containing solutions $\mathbf{f}(\mathbf{x}^1)$ and $\mathbf{f}(\mathbf{x}^2)$ is determined, and then Definition 2.7 is used to find solution $\mathbf{f}(\mathbf{x}^3)$. Finally, in Figure 8-c, we have two neighborhoods ($\mathbf{f}(\mathbf{x}^1)$,$\mathbf{f}(\mathbf{x}^3)$) and ($\mathbf{f}(\mathbf{x}^3)$,$\mathbf{f}(\mathbf{x}^2)$), where the first neighborhood is selected (given the larger margin error μ), thus finding the solution $\mathbf{f}(\mathbf{x}^4)$ using Definition 2.7 again. This procedure is repeated until convergence, when the larger margin error considering all the existing neighborhoods is smaller than μ^{stop}.

2.3 MONISE

In this section we present a novel adaptive multi-objective optimization algorithm acting as a generalization of NISE (COHON *et al.*, 1979) to deal with more than two objectives. The main distinct aspect of the proposed methodology is a new optimization model described in Definition 2.8, responsible for recursively finding the next weight vector $\mathbf{w}$ and the current estimation error μ. This generalization will be called Many Objective NISE (MONISE) and it reduces to NISE when only two objectives are considered.

2.3.1 Relaxation-approximation interpretation of the weighted sum method

Considering the utopian solution $\mathbf{r}^{utopian}$, as well as $L \geq 1$ efficient solutions $\mathbf{f}(\mathbf{x}^i) : i \in \{1,\ldots,L\}$ obtained by the weighted sum method (see Definition 2.7) using the weight vectors $\mathbf{w}^i : i \in \{1,\ldots,L\}$. For any weight vector $\mathbf{w}$, it is possible to determine the **outer approximation** $\overline{\mathbf{r}}$ and the **inner approximation** $\underline{\mathbf{r}}$ of the Pareto front guiding to a distance μ.

The **outer approximation** is a theoretical limitation for any efficient solution $\mathbf{x}^*$ attainable by the weighted sum method. So, it is possible to conclude that a relaxed objective vector $\underline{\mathbf{r}} \in \mathbb{R}^m$ will be limited by the inequalities $\mathbf{w}^{i\top}\mathbf{f}(\mathbf{x}^*) \geq \mathbf{w}^{i\top}\underline{\mathbf{r}} \geq \mathbf{w}^{i\top}\mathbf{f}(\mathbf{x}^i)\ \forall i \in \{1,\ldots,L\}$, since $\mathbf{x}^i$ is the optimal solution of the problem in Definition 2.7 considering the weight vector $\mathbf{w}^i$.

The **inner approximation** is a theoretical limitation for any efficient solution $\mathbf{x}^*$ attainable by the weighted sum method. Thus there is a weight vector $\mathbf{w}$ whose correspondent efficient solution is $\mathbf{x}^*$, and the approximate objective vector is $\overline{\mathbf{r}} \in \mathbb{R}^M$. Following the premises it is possible to demonstrate that $\mathbf{w}^\top\mathbf{f}(\mathbf{x}^*) \leq \mathbf{w}^\top\overline{\mathbf{r}} \leq \mathbf{w}^\top\mathbf{f}(\mathbf{x}^i)\ \forall i \in \{1,\ldots,L\}$, since $\mathbf{x}^*$ is the optimal solution of the problem in Definition 2.7 considering the weight vector $\mathbf{w}$.

Hence, there are two estimations, a inferior estimation $\underline{\mathbf{r}}$ associated with the front relaxation ($\mathbf{w}^{i\top}\underline{\mathbf{r}} \geq \mathbf{w}^{i\top}\mathbf{f}(\mathbf{x}^i)$), represented by the dashed line in Figure 9; and an superior estimation $\overline{\mathbf{r}}$ associated with the front approximation ($\mathbf{w}^\top\overline{\mathbf{r}} \leq \mathbf{w}^\top\mathbf{f}(\mathbf{x}^i)$), represented by the solid line in Figure 9. The space between these approximations define all solutions attainable by the weighted sum method, considering the information provided by L already found solutions.

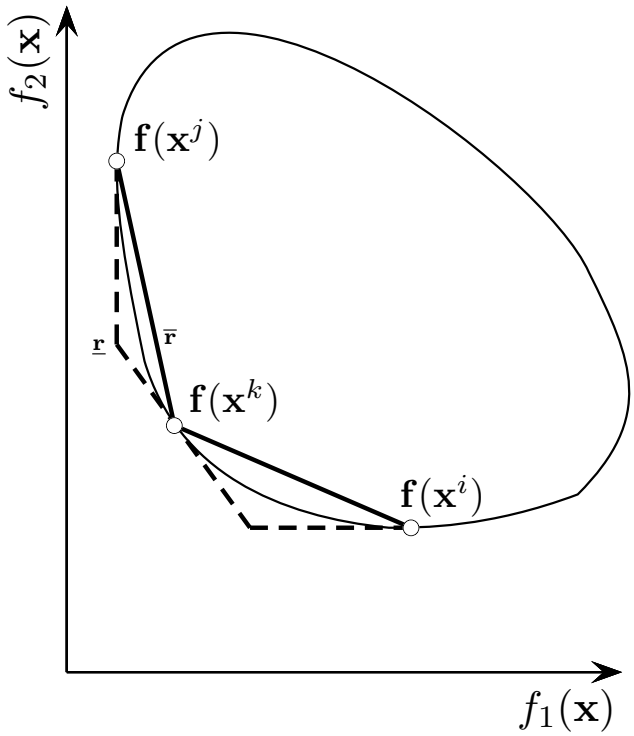


Figure 9 – Representation of inner and outer approximation derived from solutions of the weighted sum method.

2.3.2 Calculating the weight vector for the weighted sum method

The calculation of the weighted vector $\mathbf{w}$ at each iteration is done by finding the largest distance between the hyperplanes $\mathbf{w}^\top\overline{\mathbf{r}}$ and $\mathbf{w}^\top\underline{\mathbf{r}}$, provided by the solution of the following optimization problem:

Definition 2.8.

$$
\begin{aligned}
\underset{\mathbf{w},\overline{\mathbf{r}},\underline{\mathbf{r}}}{min} \quad & -\mu = \mathbf{w}^\top\underline{\mathbf{r}} - \mathbf{w}^\top\overline{\mathbf{r}} \\
subject\ to \quad & \mathbf{w}^{i\top}\underline{\mathbf{r}} \geq \mathbf{w}^{i\top}\mathbf{f}(\mathbf{x}^i)\ \forall i \in \{1,\ldots,L\} \\
& \mathbf{w}^\top\overline{\mathbf{r}} \leq \mathbf{w}^\top\mathbf{f}(\mathbf{x}^i)\ \forall i \in \{1,\ldots,L\} \\
& \underline{\mathbf{r}} \geq \mathbf{r}^{utopian} \\
& \mathbf{w} \geq \mathbf{0} \\
& \mathbf{w}^\top\mathbf{1} = 1.
\end{aligned}
$$

The problem formalized in Definition 2.8 has the role of determining the weight vector $\mathbf{w}$ and its inner and outer approximations $\overline{\mathbf{r}}$ and $\underline{\mathbf{r}}$ that leads to a maximal margin between the inner and outer approximations. From Figure 10-a to 10-d it is shown a representation of the optimization progress of Definition 2.8 for a suboptimal solution $\mathbf{w}'$ and its inner and

outer approximations $\overline{\mathbf{r}}$ and $\underline{\mathbf{r}}$. It is important to notice that, due to the non-convexity of this problem, the optimization process is capable of automatically progressing to a region with more quality (Figure 10-b onwards) until it finally achieves the best solution (Figure 10-d).

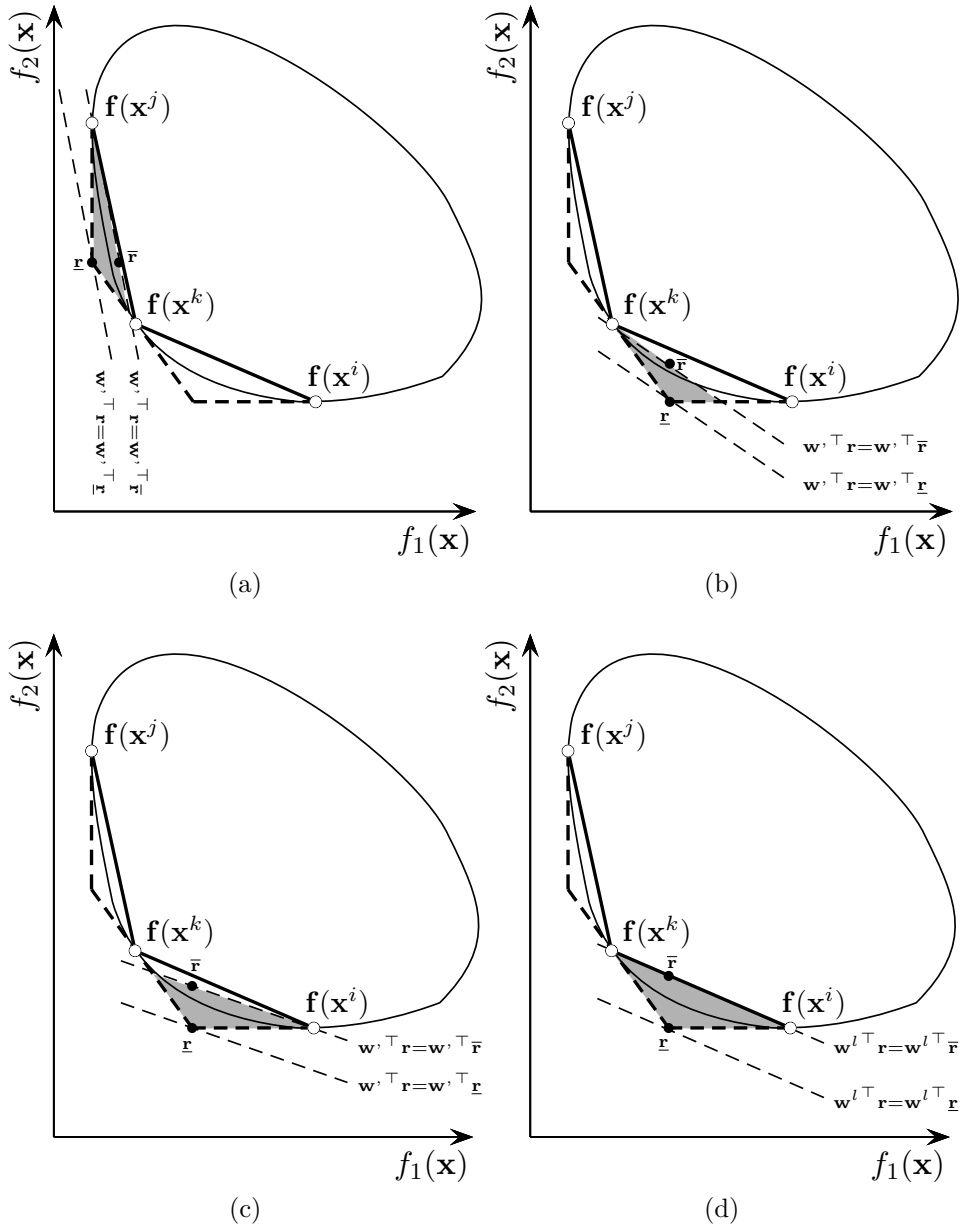


Figure 10 – Suboptimal solutions of the weight vector calculation (described in Definition 2.8) of the MONISE method.

Given that, the optimization procedure of Definition 2.8 can be seen as a search of $\mathbf{w}$ that leads to the maximum margin $\mathbf{w}^\top\overline{\mathbf{r}} - \mathbf{w}^\top\underline{\mathbf{r}}$ considering $\overline{\mathbf{r}}$ and $\underline{\mathbf{r}}$ constrained by the optimality premises of the weighted sum method.

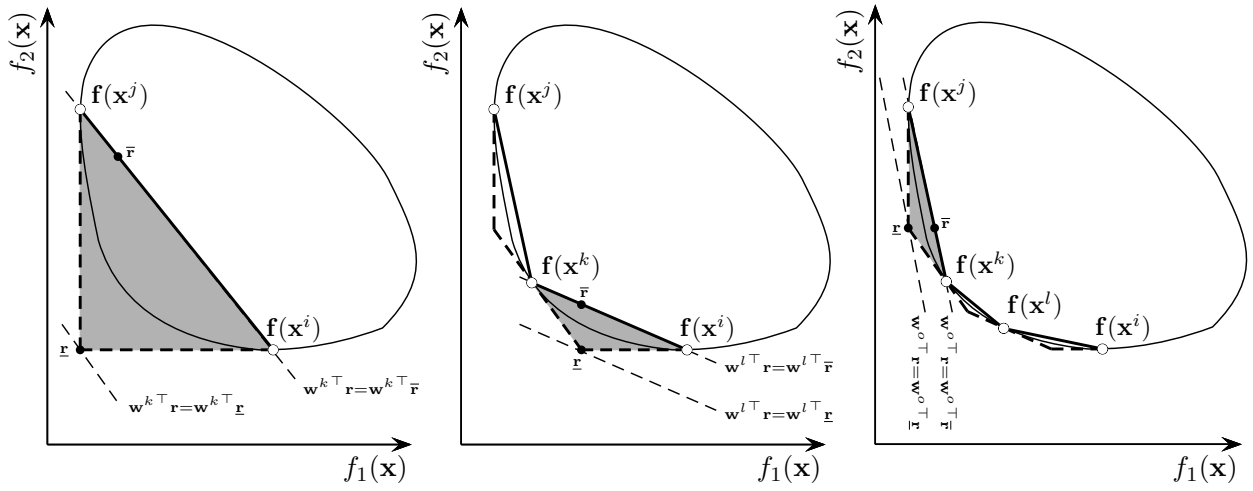
2.3.3 Outline of the methodology

The Many-Objective NISE method, called here MONISE, jointly estimates the weight vector and the estimation error. This is done without any additional structure (such as the neighborhood in NISE method), simply resorting to the previous solutions $\{\mathbf{x}^1, \ldots, \mathbf{x}^L\}$ and weight vectors $\{\mathbf{w}^1, \ldots, \mathbf{w}^L\}$. Therefore, the procedure adopted by MONISE turns to be much simpler than the one required by NISE and may be summarized in three phases:

Initialization - Consists on finding: (1) the utopian solution $\mathbf{r}^{utopian}$, and (2) at least one weight vector ($\mathbf{w}^i$) and its respective solution ($\mathbf{x}^i$).

Iterative Process - This phase is responsible for finding the weight vector $\mathbf{w}^{L+1}$ (by solving problem in Definition 2.8) and using it to find the solution $\mathbf{x}^{L+1}$ according to Definition 2.7. Furthermore, the negative of the obtained optimal value refers to the approximation error (μ) of the iteration.

Stopping Criterion - The execution of MONISE stops when the estimation error μ is lower than a threshold μ^{stop} or when the number of the already obtained efficient solutions achieves a pre-specified value R.



(a) First iteration (b) Second iteration (c) Third iteration

Figure 11 – Illustrative sequence of steps of the MONISE method.

Considering Figure 11, and given that $\mathbf{f}(\mathbf{x}^i)$ and $\mathbf{f}(\mathbf{x}^j)$ were found by in the initialization, Figure 11 depicts the iterative process of finding $\mathbf{w}^{L+1}$ using Definition 2.8 and finding the solution $\mathbf{x}^{L+1}$ according to Definition 2.7. These pictures start with the first iteration after initialization, and shows the evolution along the iterations.

2.3.4 Discussion

To exemplify the convergence of the method in terms of μ and the resultant coverage of the Pareto front, let us consider a case study with three objectives.

Definition 2.9.

$$\begin{aligned} \underset{\mathbf{x}}{\textit{minimize}} \quad & \mathbf{f}(\mathbf{x}) = \left[(x_1-1)^2, (x_2-1)^2, (x_3-1)^2\right] \\ \textit{subject to} \quad & \mathbf{x}^\top \mathbf{1} = 1, \; x_1, x_2, x_3 \geq 0 \end{aligned}$$

The simple problem presented in Definition 2.9 will be used to further investigate the behavior of MONISE. In Figure 12 it is shown the evolution of the optimized margin μ along the iterations, which monotonically decreases to zero in few iterations. Figure 13 shows a well distributed sample of the Pareto front after 300 iterations in two perspectives.

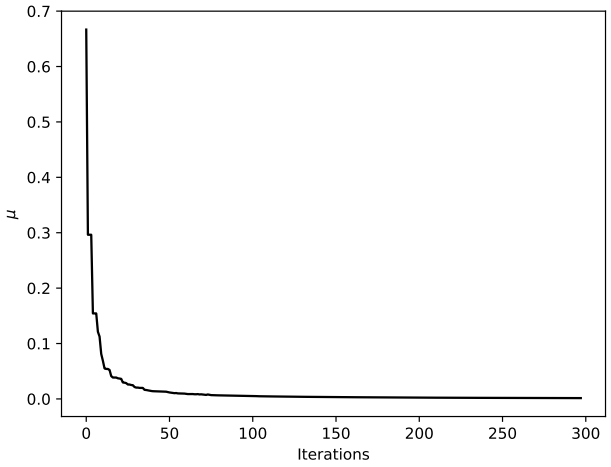


Figure 12 – Evolution of margin μ along iterations for the problem in Definition 2.9.

The quality of the Pareto front representation of this proposal is also reliable for machine learning problems. Its good performance can be verified in the reference paper (RAIMUNDO; VON ZUBEN, 2017) and it can be seen that it is more robust than other deterministic algorithms as well as evolutionary algorithms, being more suitable especially for a high number of objectives.

2.4 Summarizing comments

The focus of this thesis is on learning machines that can be represented by convex optimization problems, exploring the well known literature associated with these optimization problems.

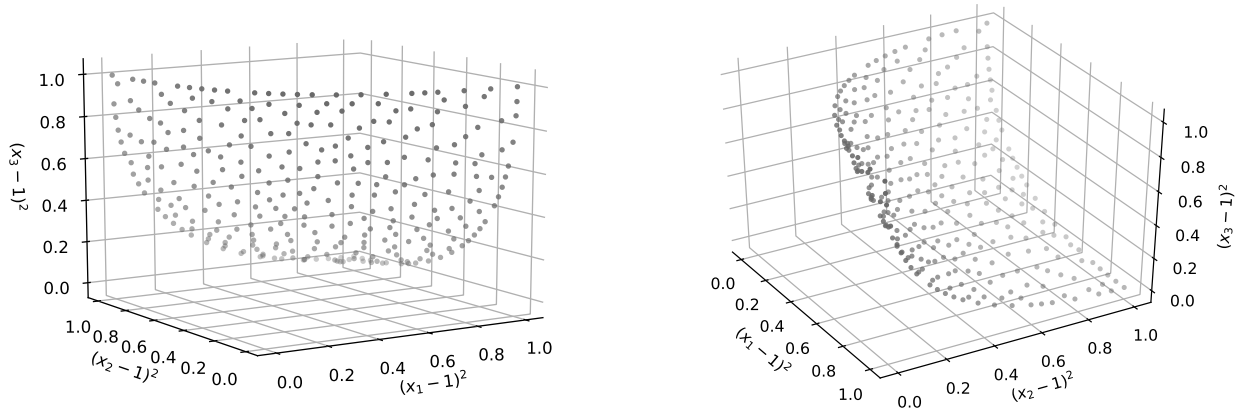


Figure 13 – Two perspectives of the non-inferior set automatically obtained at the Pareto front for the problem in Definition 2.9 using MONISE.

Since the models are convex, we decided to explore the NISE (COHON *et al.*, 1979) method because it is based on a convenient approach for multi-objective optimization: the weighted sum method. This scalarization method has a direct connection with the learning problem and can be applied in a vast scenario since it does not add constraints, thus not increasing the complexity of the model. Additionally, the sampling of the Pareto front is deterministic and diversity of trade-offs is achieved by conducting the sampling process toward less populated areas of the Pareto front. It also enables interpretability, since the weights indicate the relative importance of each objective to the solution.

However, the NISE method is not extensible to more than two objectives, which motivates the conception of the MONISE proposal, generalizing the NISE for any dimension in the objective space and improving the performance of multi-objective optimization when compared to traditional methods (RAIMUNDO; VON ZUBEN, 2017).

Chapter 3

Learning with generalized linear models

Aiming at constructing a better understanding of the statistical models used in this work, a statistical formalism is outlined here, starting with models for regression and classification and ending up with more complex models such as group lasso and multi-task learning models.

This material is based on the content of Bishop (2006) sequentially presenting the statistical development of linear regression in Section 3.1, logistic regression in Section 3.2, multinomial logistic regression in Section 3.3, regularizations in Section 3.4, a multi-label classification formulation in Section 3.5, a generic formulation of multi-task learning in Section 3.6, and ending up with Group LASSO formulation in Section 3.7.

3.1 Linear regression

The regression problem consists in finding a good mapping function $f(\cdot)$ for any sample $\mathbf{x}$ aiming at approximating a target $y \in \mathbb{R}$ leading to $f(\mathbf{x}) \approx y$. To solve this problem, it is used a set of N samples, where $\mathbf{x}_i \in \mathbb{R}^d : i \in \{1, \ldots, N\}$ represents the vector of input features and $y_i \in \mathbb{R} : i \in \{1, \ldots, N\}$ is the target value to predict. Using statistics framework, it is necessary to choose a predictive distribution and a model $f(\mathbf{x}, \boldsymbol{\theta}) : \mathbb{R}^d \rightarrow \mathbb{R}$, being $\boldsymbol{\theta}$ the feature vector of the function $f(\cdot)$.

The name linear regression came from the choice of a linear model $f(\mathbf{x}, \boldsymbol{\theta}) = \sum_{i=1}^{d} \theta_i x_i + \theta_0$ being $\boldsymbol{\theta} \in \mathbb{R}^{d+1}$. For convenience, we use a function $\boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x}), \ldots, \phi_d(\mathbf{x})]^\top$, where

$\phi_0(\mathbf{x}) = 1, \phi_i(\mathbf{x}) = \mathbf{x}_i \ \forall i \neq 0, \ i \in \{1, \ldots, d\}$, thus guiding to $f(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x})$.

Considering that the output y is determined by the function $f(\mathbf{x}, \boldsymbol{\theta})$ plus a Gaussian noise, we have:

$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \epsilon, \tag{3.1}$$

where ϵ is a Gaussian noise with zero mean and standard deviation σ. It is then possible to determine the predictive distribution $p(y|\mathbf{x})$ as a Gaussian distribution $\mathcal{N}(y|\mu, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}\right)$ with mean $\mu = f(\mathbf{x}, \boldsymbol{\theta})$ and variance $\sigma^2 = \beta^{-1}$, finding:

$$p(y|\mathbf{x}, \boldsymbol{\theta}, \beta^{-1}) = \mathcal{N}(y|f(\mathbf{x}, \boldsymbol{\theta}), \beta^{-1}) = \frac{1}{\sqrt{\beta^{-1}}\sqrt{2\pi}} e^{-\frac{(y-f(\mathbf{x},\boldsymbol{\theta}))^2}{2\beta^{-1}}}. \tag{3.2}$$

After choosing the model and the predictive distribution, it is necessary to determine the parameter $\boldsymbol{\theta}$ that makes the error ϵ as small as possible. One way to determine $\boldsymbol{\theta}$ is by means of likelihood maximization. The likelihood function for this problem is given by:

$$p(y|\mathbf{x}, \boldsymbol{\theta}, \beta^{-1}) = \prod_{i=1}^{N} \mathcal{N}(y_i|f(\mathbf{x}_i, \boldsymbol{\theta}), \beta^{-1}). \tag{3.3}$$

Taking the logarithm of the likelihood, we have that:

$$\begin{aligned} \ln p(y|\mathbf{x}, \boldsymbol{\theta}, \beta^{-1}) &= \sum_{i=1}^{N} \ln \mathcal{N}(y_i|f(\mathbf{x}_i, \boldsymbol{\theta}), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln 2\pi - \frac{1}{2}\beta \sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}_i)\right)^2. \end{aligned} \tag{3.4}$$

Discarding the constant terms, we have that:

$$\ln p(y|\mathbf{x}, \boldsymbol{\theta}, \beta^{-1}) \propto -\frac{\beta}{2} \sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}_i)\right)^2 \tag{3.5}$$

To reach a minimization problem, we define the loss function as $l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) = \sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}_i)\right)^2$, thus guiding to the usual optimization problem for linear regression:

$$\min_{\boldsymbol{\theta}} \quad l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) \equiv \sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}_i)\right)^2. \tag{3.6}$$

3.2 Logistic regression

In binary classification the target $y \in \{0,1\}$ reduces to a presence (class 1) or absence (class 0) of annotated characteristic, charging the problem to find a mapping function $f(\cdot)$ that correctly assigns any sample $\mathbf{x}$. To solve this problem, it is used a set of N samples, where $\mathbf{x}_i \in \mathbb{R}^d : i \in \{1,\ldots,N\}$ represents the vector of input features and $\mathbf{y}_i \in \{0,1\} : i \in \{1,\ldots,N\}$ is the target value to predict.Furthermore, to construct a logistic regression, it is chosen the Bernoulli distribution $p(y|z) = z^y(1-z)^{1-y}$ and a sigmoid as the classification model $f(\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}} \in [0,1]$, which describes the probability of a sample $\mathbf{x}$ belonging to class 1. Thus:

$$p(y|\mathbf{x},\boldsymbol{\theta}) = \mathcal{B}(y|f(\mathbf{x},\boldsymbol{\theta})) = \left(\frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}\right)^y \left(1 - \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}\right)^{1-y}. \tag{3.7}$$

After choosing the model and predictive distribution, it is necessary to find the parameter $\boldsymbol{\theta}$ which results in the minimal prediction error using the principle of maximal likelihood. The likelihood function to this problem is given by:

$$p(y|\mathbf{x},\boldsymbol{\theta}) = \prod_{i=1}^{N} \mathcal{B}(y_i|f(\mathbf{x}_i,\boldsymbol{\theta})). \tag{3.8}$$

Taking the logarithm of the likelihood, we have that:

$$\begin{aligned} \ln p(y|\mathbf{x},\boldsymbol{\theta}) &= \sum_{i=1}^{N} \ln \mathcal{B}(y_i|f(\mathbf{x}_i,\boldsymbol{\theta})) \\ &= \sum_{i=1}^{N} \left[y_i \ln\left(\frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}\right) + (1-y_i)\ln\left(1 - \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}\right)\right]. \end{aligned} \tag{3.9}$$

To reach a minimization problem, we define the loss function as follows:

$$l(\mathbf{x},\mathbf{y},\boldsymbol{\theta}) = \sum_{i=1}^{N} - \left[y_i \ln\left(\frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}\right) + (1-y_i)\ln\left(1 - \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}\right)\right]. \tag{3.10}$$

thus guiding to the usual optimization problem for logistic regression:

$$\min_{\boldsymbol{\theta}} \quad l(\mathbf{x},\mathbf{y},\boldsymbol{\theta}) \equiv \sum_{i=1}^{N} - \left[y_i \ln\left(\frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}\right) + (1-y_i)\ln\left(1 - \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}\right)\right]. \tag{3.11}$$

3.3 Multinomial regression

In multi-class classification the target y indicates the presence of a single label among k possible classes, charging the problem to find a mapping function $f(\cdot)$ that correctly assigns any sample $\mathbf{x}$. To solve this problem, it is used a set of N samples, where $\mathbf{x}_i \in \mathbb{R}^d : i \in \{1,\dots,N\}$ represents the vector of input features and $y_i^k \in \{0,1\} : i \in \{1,\dots,N\}, k \in \{1,\dots,K\}$ is the target value to predict, where $y_i^k = 1$ indicates that sample i belongs to class k and $y_i^k = 0$ that sample i does not belong to class k. Furthermore, to construct a multinomial regression, it is chosen the multinomial distribution $p(y|z) = \prod_{i=1}^{K} z^{y^i}$ and a softmax as the classification model $f(\mathbf{x},\boldsymbol{\theta}_k) = \frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x})}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x})}} \in [0,1]$, which describes the probability of a new sample $\mathbf{x}$ belonging to class k. Thus:

$$p(y|\mathbf{x},\boldsymbol{\theta}) = C(y|f(\mathbf{x},\boldsymbol{\theta})) = \prod_{k=1}^{K} \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x})}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x})}} \right)^{\mathbf{y}_k}. \tag{3.12}$$

After choosing the model and predictive distribution, it is necessary to find the parameter $\boldsymbol{\theta}$ which results in the minimal prediction error according to the maximal likelihood estimator. The likelihood function to this problem is given by:

$$p(y|\mathbf{x},\boldsymbol{\theta}) = \prod_{i=1}^{N} C(y_i|f(\mathbf{x}_i,\boldsymbol{\theta})). \tag{3.13}$$

Taking the logarithm of the likelihood, we have that:

$$\ln p(y|\mathbf{x},\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln C(y_i|f(\mathbf{x}_i,\boldsymbol{\theta})) = \sum_{i=1}^{N}\sum_{k=1}^{K} \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\mathbf{w}_k^\top \phi(\mathbf{x}_i)}} \right) \right]. \tag{3.14}$$

To reach a minimization problem, we define the loss function as follows:

$$l(\mathbf{x},\mathbf{y},\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{k=1}^{K} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right]. \tag{3.15}$$

thus guiding to the usual optimization problem for multinomial regression:

$$\min_{\boldsymbol{\theta}} \quad l(\mathbf{x},\mathbf{y},\boldsymbol{\theta}) \equiv \sum_{i=1}^{N}\sum_{k=1}^{K} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right]. \tag{3.16}$$

3.4 Parameter prior and regularization

Regularization is another relevant concept in machine learning. Through regularization, it is possible to deal with the bias × variance dilemma by adjusting the complexity of the model. The traditional regularizations, l_1 and l_2 norms, emerge from the prior distributions of the problem.

Assuming a generic predictive distribution $\mathcal{D}(\mathbf{y}|f(\mathbf{x},\boldsymbol{\theta}))$, when it is considered that $\boldsymbol{\theta}$ is a sample coming from a multivariate Gaussian with 0 mean and $\alpha^{-1}I$ covariance, the likelihood function is given by:

$$p(y|\mathbf{x},\boldsymbol{\theta},\sigma) = \prod_{i=1}^{N} \mathcal{D}(\mathbf{y}_i|f(\mathbf{x}_i,\boldsymbol{\theta}))\mathcal{N}(\boldsymbol{\theta}|0,\alpha^{-1}I). \tag{3.17}$$

Taking the logarithm of the likelihood, where $l(\mathbf{x}_i,\mathbf{y}_i,\boldsymbol{\theta})$ is the loss function coming from $\mathcal{D}$, we have that:

$$\ln p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) \propto \sum_{i=1}^{N} l(\mathbf{x}_i,\mathbf{y}_i,\boldsymbol{\theta}) + \frac{\alpha}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta}. \tag{3.18}$$

Setting $\lambda = \frac{\alpha}{2}$, we find the traditional learning model with l_2 regularization as follows:

$$\min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{N} l(\mathbf{x}_i,\mathbf{y}_i,\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}^\top\boldsymbol{\theta}. \tag{3.19}$$

Another approach is to choose a Laplacian distribution $\mathcal{L}(\mathbf{y}|\mathbf{x},b,\mu) = \left(\frac{1}{2b}e^{-\frac{|x-\mu|}{b}}\right)$ with $\mu = 0$ as a prior of parameter $\boldsymbol{\theta}$, finding the traditional learning model with l_1 regularization as follows:

$$\min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{N} l(\mathbf{x}_i,\mathbf{y}_i,\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_1, \tag{3.20}$$

where $||\boldsymbol{\theta}||_1 = \sum_{i=1}^{d} |\boldsymbol{\theta}_i|$.

In a more general formulation:

$$\min_{\boldsymbol{\theta}} \quad l(\mathbf{x},\mathbf{y},\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}), \tag{3.21}$$

with $l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) \equiv \sum_{i=1}^{N} l(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta})$ being the mean loss function in the samples and $r(\boldsymbol{\theta})$ being the regularization function of the model.

The l_1 norm has as its most interesting property the capability of controlling the complexity of the model by defining the sparsity of the parameter $\boldsymbol{\theta}$. In other words, when λ is correctly adjusted, a subset of the parameters $\boldsymbol{\theta}_i, i \in \{0, \ldots, d\}$ will be set to zero, resulting in a feature selection. This may not only lead to an improvement in performance, but also gives a qualitative analysis of the importance of each feature to the learning process (null parameters correspond to less important features).

The hyper-parameter λ should be properly defined for both l_1 and l_2 norms. Since we want to find the model with the best generalization, the choice of this parameter can be arbitrarily made by the specialist, or it can be done by strategies in which some values of λ are tested, so the resulting model with the best performance in a validation dataset is chosen. A representation of this methodology is presented in Figure 14.

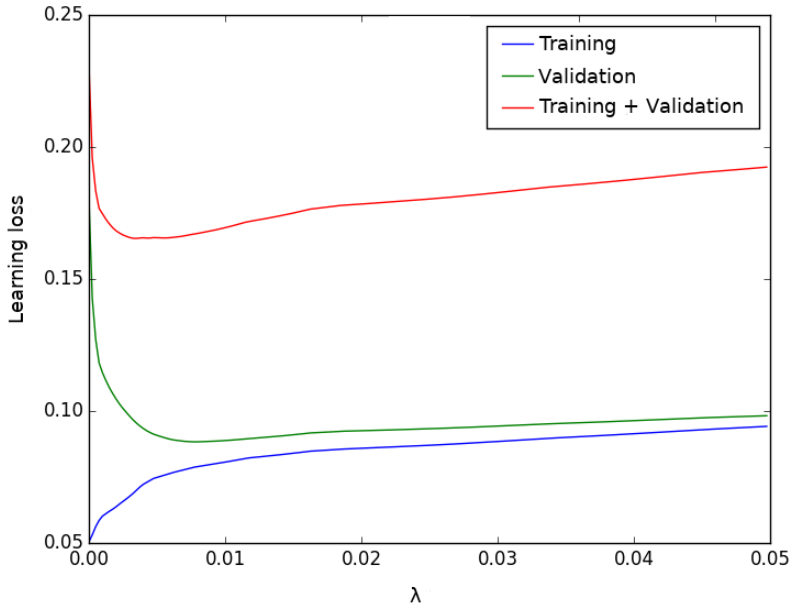


Figure 14 – Behavior of the prediction error in training and validation dataset when λ is increased.

We show in Figures 15 e 16 the typical behavior of the parameters that compose vector $\boldsymbol{\theta}$ as a function of the magnitude of λ when solving problems defined in Equations (3.19) and (3.20).

In Figure 15, associated with the regularization considering the l_2 norm, it is possible to see that the parameters tend to converge to zero but never meet this value. However, in

Figure 16 it is possible to notice the effective convergence of the parameters to zero when λ is increased, providing sparsity to the model.

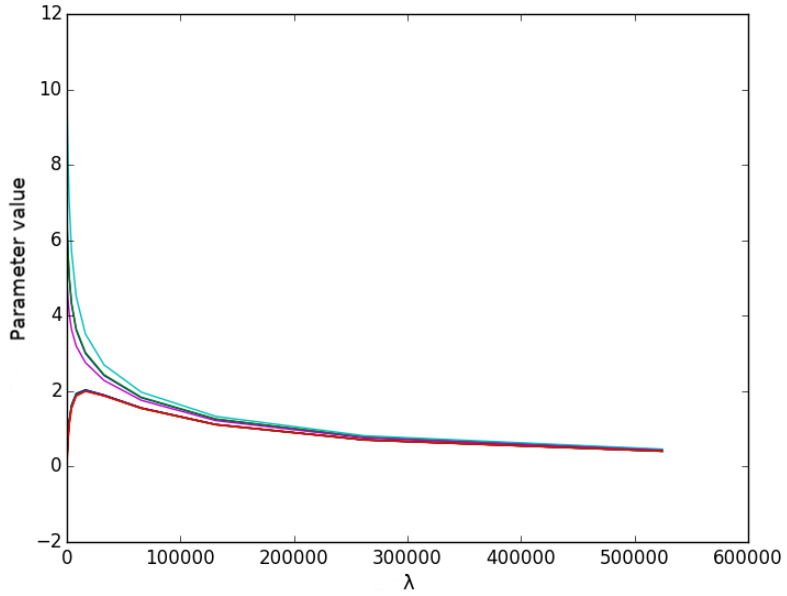


Figure 15 – Behavior of the parameters when λ is increased with l_2 norm regularization.

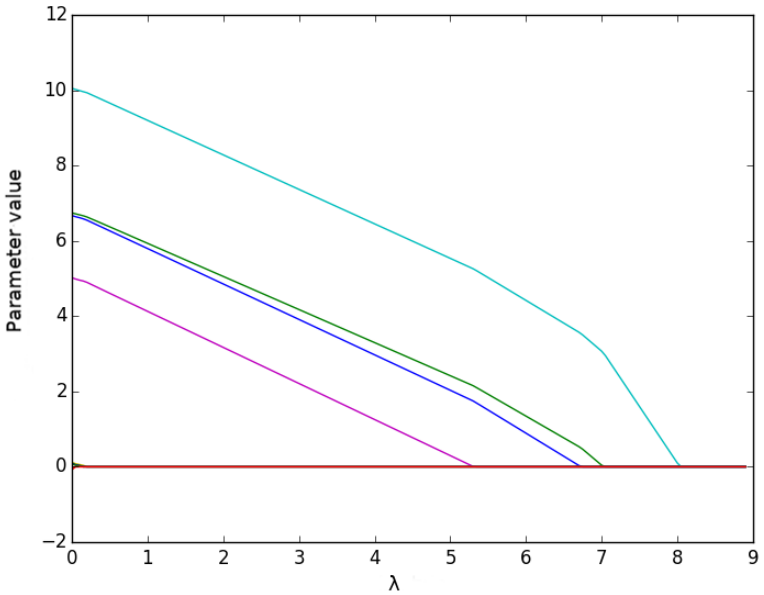


Figure 16 – Behavior of the parameters when λ is increased with l_1 norm regularization.

3.5 Multi-label classification

In multi-label classification, an arbitrary number of labels L can be assigned to a sample. Given that, the target $y^l \in \{0,1\}$ indicates the presence (class 1) or absence (class 0) of a label l, the mapping function $f(\cdot)$ (or a set of functions) should assign a set of labels for any $\mathbf{x}$. To solve this problem, it is used a set of N samples, where $\mathbf{x}_i \in \mathbb{R}^d : i \in \{1,\ldots,N\}$ represents the vector of input features and $y_i^k \in \{0,1\} : i \in \{1,\ldots,N\}$, being the samples and $l \in \{1,\ldots,L\}$ being the labels. This problem can be seen as a multiple classification problem where the classes are associated with the same set of attributes as input. Therefore, the main challenge of this problem is to discover the relationship among the labels, aiming at inducing similar classification models to classes exhibiting similar labels for the training dataset.

For the model explored in this thesis, there is a feature vector $\boldsymbol{\theta}^{(l)} \in \mathbb{R}^{d+1}$ for each label l in the logistic model $f(\mathbf{x},\boldsymbol{\theta}^{(l)}) = \frac{e^{\boldsymbol{\theta}^{(l)^\top}\phi(\mathbf{x})}}{1+e^{\boldsymbol{\theta}^{(l)^\top}\phi(\mathbf{x})}}$ for classification. To unify the parameters for all tasks in a single representation, we propose a matricial notation $\Theta = [\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(L)}]$, which will be used in the following steps.

Let us take the Bernoulli as the predictive distribution and suppose that the parameters have a generic prior $P(\Theta|b)$, where b represents the set of parameters of $P(\cdot)$. Considering that their logarithms are given by $Pr(\Theta)$, we find the following likelihood:

$$p(\mathbf{y}|\mathbf{x},\Theta) = \prod_{l=1}^{L}\prod_{i=1}^{N} \mathcal{B}(y_i^{(k)}|f(\mathbf{x}_i,\boldsymbol{\theta}^{(l)}))P(\Theta|b) \tag{3.22}$$

where n_k is the number of samples for a label k. Applying the logarithm, we find the following model:

$$\ln p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) \propto \sum_{l=1}^{L}\sum_{i=1}^{N}\left[y_i^{(l)} \ln\left(\frac{e^{\boldsymbol{\theta}^{(l)^\top}\phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^{(l)^\top}\phi(\mathbf{x}_i)}}\right) + (1-y_i^{(l)})\ln\left(1-\frac{e^{\boldsymbol{\theta}^{(l)^\top}\phi(\mathbf{x}_i)}}{1+e^{\boldsymbol{\theta}^{(l)^\top}\phi(\mathbf{x}_i)}}\right)\right] + \\ +Pr(\Theta) \tag{3.23}$$

which can be generalized to produce:

$$\ln p(\mathbf{y}|\mathbf{x},\Theta) \propto \sum_{l=1}^{L} l(\mathbf{x},\mathbf{y}^{(l)},\boldsymbol{\theta}^{(l)}) + Pr(\Theta). \tag{3.24}$$

This formulation results in a model with independent classifiers linked by a regularization. This assumption will be useful for the models we propose in this thesis, but another

assumption can be made in multi-label classification, resulting in different ways of inducing relationships among the labels.

3.6 Multi-task learning

In multi-task learning, it is assumed the existence of T learning tasks of any nature (here simplified to binary classification) that shares the input $\mathbf{x}^t \in \mathbb{R}^d : \forall t \in \{1,\ldots,T\}$ and output $y^t \in \{0,1\} : \forall t \in \{1,\ldots,T\}$ space. Considering the existence of T tasks, and $n_t : t \in \{1,\ldots,T\}$ samples for each task. Let us assume that $\mathbf{x}_i^{(t)} \in \mathbb{R}^d : t \in \{1,\ldots,T\}, i \in \{1,\ldots,n_t\}$ represents the input feature vector and $\mathbf{y}_i^{(t)} \in \{0,1\} : t \in \{1,\ldots,T\}, i \in \{1,\ldots,n_t\}$ is the output value to predict. The joint learning is here applied since it is expected to explore the similarities among the tasks. And the goal of multi-task learning in this context is to discover the structural relationship among the labels, thus correctly inducing knowledge sharing.

Given that, there is a feature vector $\boldsymbol{\theta}^{(t)} \in \mathbb{R}^{d+1}$ for each task t in the logistic model $f(\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}{1+e^{\boldsymbol{\theta}^\top \phi(\mathbf{x})}}$. So, to unify the parameters for all tasks in a single representation, we propose a matricial notation $\Theta = [\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(T)}]$, which will be used in the following steps.

Let us take the Bernoulli as the predictive distribution and suppose that the parameters have a generic prior $P(\Theta|b)$, where b represents the set of parameters of $P(\cdot)$. Considering that their logarithms are given by $Pr(\Theta)$, we find the following likelihoods:

$$p(\mathbf{y}|\mathbf{x},\Theta) = \prod_{t=1}^{T}\prod_{i=1}^{n_t} \mathcal{B}(\mathbf{y}_i^{(t)}|f(\mathbf{x}_i^{(t)},\boldsymbol{\theta}^{(t)}))P(\Theta|b) \tag{3.25}$$

where n_t is the number of samples for target t. Applying the logarithm, we find the following models:

$$\ln p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) \propto \sum_{t=1}^{T}\sum_{i=1}^{n_t}\left[\mathbf{y}_i^{(t)} \ln\left(\frac{e^{\boldsymbol{\theta}^{(t)\top}\phi(\mathbf{x}_i^{(t)})}}{1+e^{\boldsymbol{\theta}^{(t)\top}\phi(\mathbf{x}_i^{(t)})}}\right) + (1-\mathbf{y}_i^{(t)}) \ln\left(1-\frac{e^{\boldsymbol{\theta}^{(t)\top}\phi(\mathbf{x}_i^{(t)})}}{1+e^{\boldsymbol{\theta}^{(t)\top}\phi(\mathbf{x}_i^{(t)})}}\right)\right] + Pr(\Theta) \tag{3.26}$$

which can be generalized to produce:

$$\ln p(\mathbf{y}|\mathbf{x},\Theta) \propto \sum_{t=1}^{T} l(\mathbf{x}^{(t)},\mathbf{y}^{(t)},\boldsymbol{\theta}^{(t)}) + Pr(\Theta). \tag{3.27}$$

This regularization term can be used to promote knowledge sharing among the tasks. It is worth mentioning that it can be done by making all parameters share the same sparsity

pattern, for instance, by canceling out the contribution of a subset of features for all tasks (GONG *et al.*, 2012), by creating a low-dimensional shared subspace (ANDO; TONG, 2005), or by considering that the parameter vectors of the tasks obey the same Gaussian distribution, thus sharing the same co-variance matrix (GONÇALVES *et al.*, 2015).

3.7 Multi-input learning

Analogous to what was verified when dealing with multiple dealing with multiple outputs, there are situations where identifying the group of inputs may guide to improved performance and interpretability (YUAN; LIN, 2006; MEIER *et al.*, 2008). It is possible to split the input $\mathbf{x}$ into V groups of inputs. The formulation explored here is based on Group LASSO, where a distinct regularization is applied to the parameters of each group:

$$\min_{\boldsymbol{\theta}} \quad l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) + \lambda \sum_{v=1}^{V} \sqrt{\sum_{i \in \mathcal{V}^v} \boldsymbol{\theta}_i^2}, \tag{3.28}$$

being $\mathcal{V}^v$ a set that contains the feature indexes of group v.

This model is capable of switching off groups of inputs that does not help in the prediction.

3.8 Summarizing comments

Focused on linear models and generalized linear models, this section presented the main concepts of the base models used in this thesis. The limitations around linear models are motivated by the possibility of simplifying the optimization step of the learning process and to keep convexity. Convexity will be required to properly explore the Pareto front in the multi-objective formulations.

Additionally, the potential oversimplification of the models will be attenuated by the use of ensemble methods, as will be described in the next chapters, responsible for aggregating multiple models, thus inherently increasing the complexity of the aggregated final model. This strategy was conceived to promote high performance without losing simplicity and scalability of the learning process.

Chapter 4

The proposed framework for multi-objective learning

The main proposal of this thesis is a novel approach to deal with machine learning problems by exploring its multi-objective nature. Three stages are involved: (1) **multi-objective modelling** consists in adapting or modifying machine learning models to uncover the conflicting objectives of the model; (2) **multi-objective training** consists in adopting an a posteriori multi-objective method that will find a diverse sampling of the Pareto-optimal solutions; (3) **model selection** or **ensemble aggregation** deals with the multiple efficient solutions to create a proper learning machine.

The main potential of this methodology is to give more flexibility to the model (for example allowing a flexible weighting of each class loss) and use a multi-objective optimization to find diverse and accurate models. This allows the use of simple cross-validation to select a single model among the candidates, as well as the use of simple ensemble filtering and aggregation techniques to build a learning machine at the end of the process.

4.1 Multi-objective modeling

Many machine learning problems have, in its essence, a multi-objective nature. The role of this section is to modify and extend the machine learning models of Chapter 3 to highlight the conflicting objectives. To facilitate the comprehension of the learning problem as a multi-objective optimization problem, Equation (4.1) presents an objective function in which each conflicting objective $f_i(\zeta)$ is accompanied by a weight coefficient w_i. This format highlights all the conflicting objectives, and is also compatible with the weighted

sum method, allowing a straightforward application of NISE and MONISE, responsible for recursively applying scalarization to find a representation of the Pareto front.

$$\underset{\zeta}{\text{miniminze}} \quad w_1 f_1(\zeta) + \ldots + w_m f_m(\zeta) \tag{4.1}$$

where $f_1(\zeta), \ldots, f_m(\zeta)$ are the m conflicting objectives of the problem and $w_1, \ldots, w_m$ are the m coefficients of these objectives.

Using this notation with weights to be defined, we present: (1) a multi-objective reinterpretation of generalized linear models with regularization in Section 4.1.1; (2) a multi-objective reinterpretation of regularized logistic models in Section 4.1.2; (3) an attempt to explore two aspects of the regularized multinomial logistic regression models in Section 4.1.3; (4) an adaptation of the logistic regression model to deal with the multiple outputs of multi-label classification datasets in Section 4.1.4; (5) a deeper modification of the logistic regression model to deal with multi-task learning problems, which is described in Section 4.1.5; (6) a model capable of transferring knowledge from a source task to a target task in Section 4.1.6; and (7) an illustrative model of a possible extension to multi-view learning that controls the level of importance of each vision of the learning problem in Section 4.1.7.

4.1.1 Generalized linear models with regularization

The adaptation to the multi-objective context in regularized models consists in highlighting, in Equation (3.21) ($l(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$), the conflict between the loss $l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$ and the regularization strength $r(\boldsymbol{\theta})$. Given that, dividing the function by $\frac{1}{1+\lambda}$ and defining $\mathbf{w}_1 = \frac{1}{1+\lambda}$ and $\mathbf{w}_2 = \frac{\lambda}{1+\lambda}$, it is possible to produce the multi-objective format (weighted sum method format) in Equation (4.2):

$$\min_{\boldsymbol{\theta}} \quad w_1 l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) + w_2 r(\boldsymbol{\theta}). \tag{4.2}$$

4.1.2 Regularized logistic regression

Using the model developed in Section 3.2 and adding a regularization developed in Section 3.4, we can find the standard regularized logistic model:

$$\min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{N} - \left[y_i \ln \left(\frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1 + e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}} \right) + (1 - y_i) \ln \left(1 - \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1 + e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}} \right) \right] + \lambda r(\boldsymbol{\theta}). \tag{4.3}$$

Adapting the model to follow the canonical form of (Equation (4.2)), we can produce the **regularized logistic regression** formulation:

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{i=1}^{N} - \left[y_i \ln \left(\frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1 + e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}} \right) + (1 - y_i) \ln \left(1 - \frac{e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}}{1 + e^{\boldsymbol{\theta}^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}). \tag{4.4}$$

4.1.3 Regularized multinomial logistic regression models

Starting with the model developed in Section 3.3 and adding a regularization term developed in Section 3.4, we can find the standard regularized multinomial model:

$$\min_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + \lambda r(\boldsymbol{\theta}), \tag{4.5}$$

where u_k can be used to control the importance of the loss of each class in the optimization. Doing $u_k = 1 \; \forall k \in \{1, \ldots, K\}$ results in the standard model and an ad-hoc balancing consists in doing $u_k = n_k \; \forall k \in \{1, \ldots, K\}$, being n_k the number of samples for the class k.

Simplifying to the standard model and adapting the model to follow the template of Equation (4.2), we can find the **conflicting regularized multinomial logistic regression** formulation (RAIMUNDO *et al.*, 2018):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}). \tag{4.6}$$

This formulation is particularly useful in balanced classification datasets. However, it is possible to think the loss associated with each class $-\frac{1}{u_k} \sum_{i=1}^{N} \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right]$ as a conflicting objective, as well as the regularization component $r(\boldsymbol{\theta})$. With those considerations we can formulate this problem with $K+1$ conflicting objectives, called here as **class-conflicting regularized multinomial logistic regression**:

$$\min_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K} w_k \left[- \sum_{i=1}^{N} y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_{K+1} r(\boldsymbol{\theta}). \tag{4.7}$$

It is possible to see that the formulation at Equation (4.7) generalizes the formulation at Equation (4.6), allowing the search for useful models in imbalanced classification datasets.

4.1.4 Multi-label classification model

Multi-label classification consists in a scenario where each sample can belong to more than one class. Two traditional approaches consist in: (1) consider each label as a binary independent classification problem; (2) consider each existing combination of labels as a class, and create a single classifier. We shall adopt the first approach, but having a single parameter vector for all labels on a single model, called here as **single-parameter label-conflicting regularized multinomial logistic regression** formulation (RAIMUNDO; VON ZUBEN, 2018b):

$$\min_{\boldsymbol{\theta}} \quad \sum_{l=1}^{L} v_l l(\mathbf{x}, \mathbf{y}^l, \boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||^2 \equiv \sum_{l=1}^{L} w_l l(\mathbf{x}, \mathbf{y}^l, \boldsymbol{\theta}) + w_{L+1} ||\boldsymbol{\theta}||^2, \tag{4.8}$$

where $||\boldsymbol{\theta}||^2$ is the regularization component, λ is the regularization parameter, $w_i = \frac{v_i}{\sum_{k=1}^{L} v_k + \lambda} \forall i \in \{1, \ldots, L\}$, and $w_{L+1} = \frac{\lambda}{\sum_{k=1}^{L} v_k + \lambda}$.

To manage the class imbalance of every label, we used a multinomial regression to make an ad-hoc balancing approach to find a parameter $\boldsymbol{\theta}^l$ for each label l (RAIMUNDO; VON ZUBEN, 2018b).

$$\min_{\boldsymbol{\theta}^l} l(\mathbf{x}, \mathbf{y}^l, \boldsymbol{\theta}^l) = -\sum_{i=1}^{N} \left[\frac{1}{n_1^l} \mathbf{y}_i^l \ln\left(f(\mathbf{x}_i, \boldsymbol{\theta}^l)\right) + \frac{1}{n_0^l} (1 - \mathbf{y}_i^l) \ln\left(1 - f(\mathbf{x}_i, \boldsymbol{\theta}^l)\right) \right], \tag{4.9}$$

where n_1^l is the number of 1s in label l and n_0^l is the number of 0s in label l, with softmax function $f(\mathbf{x}, \boldsymbol{\theta}^l) = \frac{e^{{\boldsymbol{\theta}_1^l}^\top \mathbf{x}}}{e^{{\boldsymbol{\theta}_0^l}^\top \mathbf{x}} + e^{{\boldsymbol{\theta}_1^l}^\top \mathbf{x}}} \in [0, 1]$ as the input-output model and $\boldsymbol{\theta}_0^l \in \mathbb{R}^{d+1}$ and $\boldsymbol{\theta}_1^l \in \mathbb{R}^{d+1}$ as the parameters for class 0 and 1 for the label l.

4.1.5 Multi-task learning

Multi-task learning can be seen as a multi-objective optimization problem (LI *et al.*, 2015; BAGHERJEIRAN, 2007). To explore this perspective we first present a general multi-task learning formulation:

$$\min_{\boldsymbol{\theta}} \quad \sum_{t=1}^{T} l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\theta}^{(t)}) + \lambda r(\boldsymbol{\theta}). \tag{4.10}$$

Starting from this, an initial proposal consists in simplifying the parameters $\boldsymbol{\theta}^{(t)}, t \in \{1, \ldots, T\}$ as a single parameter $\boldsymbol{\theta} = \boldsymbol{\theta}^{(1)} = \ldots = \boldsymbol{\theta}^{(T)}$ and consider the regularization as a simple 1-norm regularization $r(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_1$. Interestingly, it is possible to consider the loss

of each task $l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\theta})$ as a conflicting objective as well as the regularization component $r(\boldsymbol{\theta})$, so that we can modify the previous formulation to meet the weighted sum method format, as shown in Equation (4.11), which describes the **single-parameter task-conflicting regularized logistic regression** formulation (RAIMUNDO; VON ZUBEN, 2018a):

$$\min_{\boldsymbol{\theta}} \quad \sum_{t=1}^{T} w_t l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\theta}) + w_{T+1}||\boldsymbol{\theta}||_1. \tag{4.11}$$

Even though the formulation considers a single parameter vector $\boldsymbol{\theta}$, this limitation is easily bypassed by the fact that multi-objective optimization methods generates multiple solutions with distinct trade-offs among the objectives. Given that, generating multiple efficient solutions, it is very likely to find good parameter vectors for the tasks. Then, when filtering and aggregation of these solutions are applied, it is possible to find high quality learning machines for all tasks.

4.1.6 Transfer learning

Considering a set of source tasks $\mathcal{S} \subset \mathcal{P}$ and a target $t \in \mathcal{P}$ task, $t \neq s$ with $s \in \mathcal{S}$, transfer learning consists in transmitting the extracted knowledge from $\mathcal{S}$ to t.

When we consider a single source $s \in \mathcal{S}$, it is possible to define a single parameter for both source and target tasks ($\boldsymbol{\theta} = \boldsymbol{\theta}^s = \boldsymbol{\theta}^t$). In this case, considering that the learning tasks have conflicting objectives, we reach the following formulation, called here as **single-parameter transferring regularized multinomial logistic regression** formulation (BESERRA *et al.*, 2018):

$$\min_{\boldsymbol{\theta}} \quad w_1 \left(l(\boldsymbol{\theta}, \mathbf{x}^s, \mathbf{y}^s) + \lambda r(\boldsymbol{\theta})\right) + w_2 \left(l(\boldsymbol{\theta}, \mathbf{x}^t, \mathbf{y}^t) + \lambda r(\boldsymbol{\theta})\right), \tag{4.12}$$

where the regularization parameter λ can be obtained using cross validation in the source task.

It is also possible to use multiple sources using a formulation similar to that obtained by the multi-task learning formulation.

4.1.7 Group LASSO and multi-view learning

Another scenario where multi-objective formulation can be useful is in multi-view learning. Our approach is inspired by Group LASSO (YUAN; LIN, 2006), and consider each view (subset of features) as a group. Consider that we have V views, and let $\mathcal{V}^v$ be a set that

contains the feature indexes of group v. The first formulation considers the learning loss as conflicting with the regularization for the groups:

$$\min_{\boldsymbol{\theta}} \quad w^l l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) + w^r \sum_{v=1}^{V} \sqrt{\sum_{i \in \mathcal{V}^v} \boldsymbol{\theta}_i^2}. \tag{4.13}$$

Another approach considers each view and the learning loss as conflicting objectives, resulting in the following formulation:

$$\min_{\boldsymbol{\theta}} \quad w^l l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) + \sum_{v=1}^{V} w_v^r \sqrt{\sum_{i \in \mathcal{V}^v} \boldsymbol{\theta}_i^2}. \tag{4.14}$$

Now considering that the parameter vector of each group v is given by $\boldsymbol{\theta}^{(v)}$, we can propose an average classifier for every view, and consider the loss for each view and the regularization of all views as conflicting objectives, thus producing:

$$\min_{\boldsymbol{\theta}} \quad \sum_{v=1}^{V} w_v^l l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}^{(v)}) + w^r \sum_{v=1}^{V} \sqrt{\sum_{i \in \mathcal{V}^v} \boldsymbol{\theta}_i^2}. \tag{4.15}$$

And finally we consider the loss for each view and for each regularization as conflicting, thus producing:

$$\min_{\boldsymbol{\theta}} \quad \sum_{v=1}^{V} w_v^l l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}^{(v)}) + \sum_{v=1}^{V} w_v^r \sqrt{\sum_{i \in \mathcal{V}^v} \boldsymbol{\theta}_i^2}. \tag{4.16}$$

This last formulation is also a general formulation for all other previous formulations in this section: (1) making $w_v^l = w^l \ \forall v \in V$ and $w_v^r = w^r \ \forall v \in V$, Equation (4.16) becomes Equation (4.13); (2) making $w_v^l = w^l \ \forall v \in V$, Equation (4.16) becomes Equation (4.14); and (3) making $w_v^r = w^r \ \forall v \in V$, Equation (4.16) becomes to Equation (4.15).

4.2 Multi-objective training

The multi-objective training is a procedure that guides to efficient learning models exhibiting distinct trade-offs among the conflicting objectives. An example of a three-objective Pareto front representation is given by Figure 17. This example consists in using the model in Equation (4.10) to deal with three well known monks' datasets[1] that share the same feature

[1]Available at <archive.ics.uci.edu/ml/machine-learning-databases/monks-problems/>

vector, characterizing a multi-task learning problem. Each point of this representation is an efficient solution obtained by attributing a specific relative relevance of all the datasets.

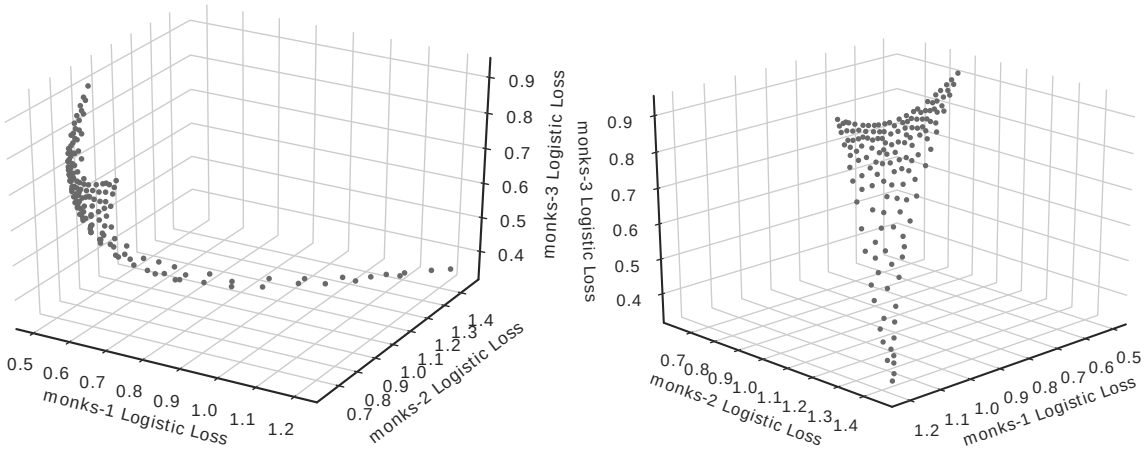


Figure 17 – Two perspectives of the same Pareto front representation, with the logistic error of each learning task as the three objective functions (©2018 IEEE) (RAIMUNDO; VON ZUBEN, 2018a).

To generate the representation of Figure 17, we have adopted MONISE.

4.3 Ensemble filtering and aggregation[2]

Aiming at dealing with multiple efficient models generated by multi-objective optimization, we rely on ensemble methods. Generally, the use of an ensemble involves three steps: generation of learning machines, selection of a proper subset of these machines and composition of the selected machines to achieve a single outcome (ZHOU, 2012). Multi-objective approaches usually address multiple performance metrics (instead of solely model losses or regularization strengths) in the first two steps of the ensemble framework.

4.3.1 Filtering

Since the multi-objective training is already a generator of a diverse set of ensemble components, we now need to select a single model among the multiple efficient learning models or filter/aggregate the ensemble components. **Filtering** is an important step since it can reduce the computational cost in prediction while improving the generalization capa-

[2]This section is an amended version of Exploring multi-objective training in multi-class classification (©2018 IEEE)(RAIMUNDO *et al.*, 2018)

bility (ZHOU, 2012). We explore multiple possibilities, aiming at finding the most accurate classifier:

- **Winner takes all** (wta) - Considering the performance in the validation set (given a chosen performance metric), the best classifier is chosen. This ensemble filtering is equivalent to a model selection.
- **Winner takes all per class** (wtaPL) - Considering the performance in the validation set (given a chosen performance metric in a one versus all approach), the best classifier for each class is chosen.
- **Elite** K (elite) - Considering the performance in the validation set (given a chosen performance metric), the K best classifiers are chosen.
- **Elite** K **per class** (elitePL) - Considering the performance in the validation set (given a chosen performance metric in a one versus all approach), the K best classifiers are chosen for each class.
- **Multi-objective Filtering** (moPL) - Considering the performance in the validation set (given each chosen performance metric), a classifier is selected if there is no other classifier better than this classifier in all metrics (non-dominated classifiers) (KRAUS *et al.*, 2011).
- **Maximum diversity** (max-div) - The components with maximal diversity are selected, given that we want K components.

To better explain **maximum diversity**, it is necessary to further analyze an explicit metric of diversity. The metric of double-fault (SCHAPIRE, 2003) and the adapted some-correct $s_{i,j}$ metric are interesting metrics, because of its pairwise strategy. The some-correct metric has the quality of being simple, only computing the ratio of samples in which, at least one of the learning machines, i or j, is capable of correctly predicting that sample. To compute the diversity of a set of components C, Equation (4.17) was used:

$$\frac{\sum_{i \in C} \sum_{j \in C, j \neq i} s_{i,j}}{|C|^2 - |C|} \tag{4.17}$$

Given this evaluation, the maximum diversity filtering consists in selecting K components from the generated set of components $\mathcal{U}$ with maximal diversity. This procedure is done by solving the problem in Definition 4.1.

Definition 4.1. *Maximal diversity filtering.*

$$\begin{aligned} \underset{\mathbf{u}}{\textit{maximize}} \quad & \frac{\sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{U}}\mathbf{u}_i\mathbf{u}_j s_{i,j}}{K^2-K} \\ \textit{subject to} \quad & \sum_{i\in\mathcal{U}}\mathbf{u}_i = K \\ & \mathbf{u}_i \in \{0,1\}, \forall i \in \{1,\ldots,|\mathcal{U}|\} \end{aligned} \tag{4.18}$$

Finally, in the new set $\overline{\mathcal{U}}$, $\mathbf{u}_i = 1$ indicates that $i \in \overline{\mathcal{U}}$ and $\mathbf{u}_i = 0$ indicates that $i \notin \overline{\mathcal{U}}$, being $\overline{\mathcal{U}}$ the set of selected components.

4.3.2 Aggregation

Finally, after filtering, it is necessary to aggregate the outputs of every classifier. For this stage of the proposal, we implemented some methods:

- **Simple Vote** (svote) - The votes are accounted, and the most frequent is chosen as the final decision (ZHOU, 2012).
- **Weighted Vote** (wv) - The magnitude of the metric used to select the component is used to weight the relative importance of the component to the prediction.
- **Distribution summation** (dsum) - Similar to simple voting, this method sums the confidence of the prediction for each class.
- **Bayesian combination** (bc) - This method is equivalent to weighted voting for the distribution summation scheme, where each confidence is weighted by the quality of the predictor on each class.
- **Stacking** (stk) - This method consists in creating a new classifier on the top of the outputs of all component prediction. To do that, we consider the outputs of the components as features, and train another classifier using this new feature space.

4.4 Summarizing comments

To give a better understanding of the framework operation, Figure 18 shows a representation of the process. In Figure 18-1, it is shown a generic multi-objective unconstrained model weighted by a generic weight vector **w**; Figure 18-2 shows the training procedure, in which MONISE (or NISE) method is responsible for determining the weight vectors

$\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^R$ using an iterative process that generates the parameter vectors $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^R$ which is also represented in a Pareto front determined by these solutions in the objective space determined by the objectives $f_1(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}), f_2(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}), \ldots, f_m(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$. Those solutions $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^R$ are also learning machines in which their predictions $p(\cdot)$ are used to first filter the most proper machines in Figure 18-3 and then, they are aggregated in Figure 18-4, resulting in the final predictor $p_{ens}(\cdot)$.

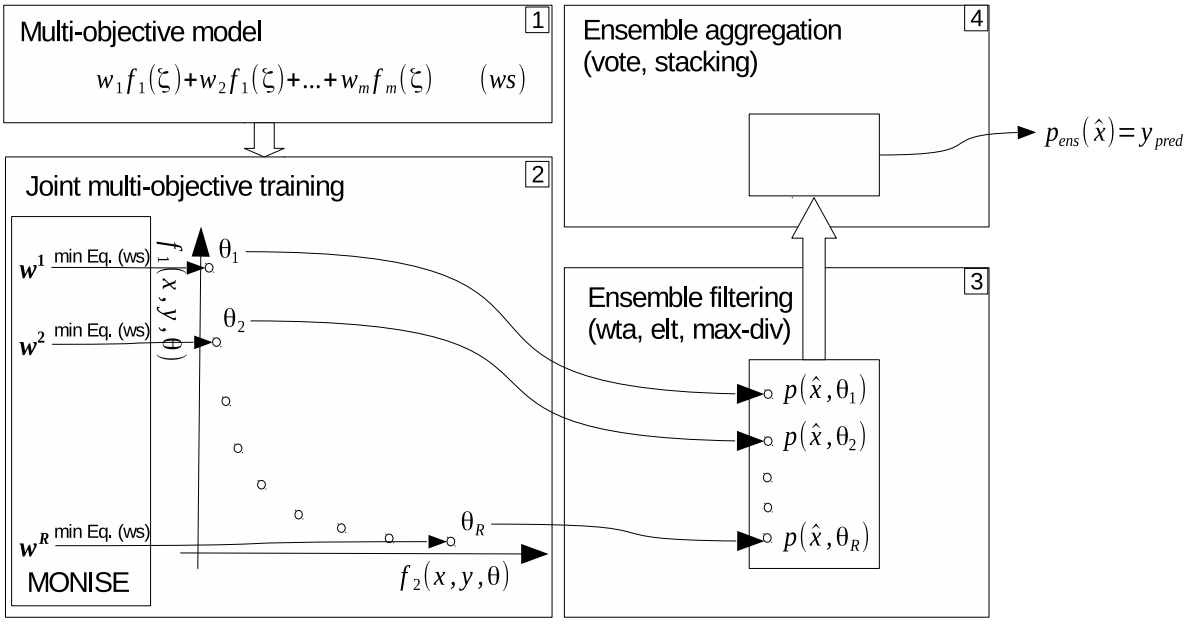


Figure 18 – Overview of the proposed framework for multi-objective learning.

Chapter 5

Related works

5.1 Multi-objective learning in the literature[1]

This thesis proposes a unified framework for multi-objective modelling and training being applied to a vast set of classification tasks in machine learning. Despite the fact that multi-objective optimization methods are not widely used in machine learning, the literature is vast in approaches and scenarios being used to search for both interpretable and accurate models, models generated to have complementary properties, and conflicting loss functions building ensembles (BRAGA *et al.*, 2006; JIN; SENDHOFF, 2008; JIN *et al.*, 2009). It is also used to model selection, ensemble generation, filtering, and aggregation (ZHOU, 2012); to the classification of imbalanced datasets (AKAN; SAYIN, 2014; GARCÍA *et al.*, 2010); and to multi-task learning (BAGHERJEIRAN, 2007).

When models should have more than one desired quality (usually represented by performance metrics), a **model selection** that uses multi-objective methods can be capable of providing efficient options to the user characterized by multiple trade-offs among conflicting metrics (MÜSSEL *et al.*, 2012) and to select models when performance and complexity of the model are simultaneously optimized (MIRANDA *et al.*, 2012; MIRANDA *et al.*, 2014; IGEL, 2005; MAO *et al.*, 2013). Other approaches do not act directly in the model selection, but contribute to this task by creating Pareto front representations that generate models exhibiting diverse trade-offs among more than one performance metric; and after this procedure, another performance metric is used to select the best model among these generated models (FERNÁNDEZ CABALLERO *et al.*, 2010; PILAT; NERUDA, 2013; ROSALES-PÉREZ *et*

[1]This section is an amended version of Exploring multi-objective training in multi-class classification (©2018 IEEE)(RAIMUNDO *et al.*, 2018)

al., 2015).

A similar procedure is used to **generate ensemble** components that correspond to trade-offs between more than one performance metric using different metrics: accuracy for each label (AHMADIAN *et al.*, 2007; ENGEN *et al.*, 2009); recall per label (WANG *et al.*, 2014); prediction error and complexity of the model (OLIVEIRA *et al.*, 2005; SMITH; JIN, 2014); prediction error on different training sets (ABBASS, 2003); false positive and false negative rates (CASTILLO *et al.*, 2006); number of leaves of a genetic program, false positive and false negative rates (NAG; PAL, 2016); error, neural tree size and diversity index (OJHA *et al.*, 2017); between-class and within-class variance on feature extraction (ALBUKHANAJER *et al.*, 2017); precision and recall (EKBAL; SAHA, 2016); precision, recall and number of selected features (SAHA *et al.*, 2016); and different metrics of clustering validation (MUKHOPADHYAY *et al.*, 2009).

Considering models already generated, there are some multi-objective procedures for **ensemble filtering**: filtering by excluding machines with other machines having better performance in all performance metrics (KRAUS *et al.*, 2011; ZHANG *et al.*, 2011); and filtering by selecting components at the elbow on the Pareto front (SMITH; JIN, 2014). Furthermore, some methods select a set of classifiers optimizing different conflicting objectives: prediction error and number of components on the ensemble (AHMADIAN *et al.*, 2007); accuracy for each class (ENGEN *et al.*, 2009); precision and recall (EKBAL; SAHA, 2010; EKBAL; SAHA, 2012); accuracy and diversity (KRAWCZYK; WOZNIAK, 2013; KRAWCZYK; WOŹNIAK, 2014; LÖFSTRÖM *et al.*, 2009; ZHANG *et al.*, 2011); diversity metrics, size of ensemble and prediction error (DOS SANTOS *et al.*, 2008).

Some methods perform **ensemble generation and filtering** simultaneously, by maximizing accuracy and diversity (CHANDRA; YAO, 2004; CHANDRA *et al.*, 2006; CHANDRA; YAO, 2006), or by minimizing prediction error and maximizing diversity (OLIVEIRA *et al.*, 2005; BHOWAN *et al.*, 2011a; BHOWAN *et al.*, 2011b; BHOWAN *et al.*, 2013; NETO *et al.*, 2013).

Some methods deal with classification on **imbalanced** datasets by optimizing multiple error losses such as classification loss in each class (GARCÍA *et al.*, 2010), positive and negative empirical errors in addition to the typical margin maximization (AKAN; SAYIN, 2014), and optimizing the recall in each label (WANG *et al.*, 2014).

In addition to not being a unified framework, most proposals are heuristic-based, with emphasis on NSGA-II (AHMADIAN *et al.*, 2007; SMITH; JIN, 2014; EKBAL; SAHA, 2010; EKBAL; SAHA, 2012; EKBAL; SAHA, 2016; ALBUKHANAJER *et al.*, 2017; SAHA *et al.*, 2016; GARCÍA *et al.*, 2010), although other evolutionary-like multi-objective optimiza-

tion methods have also been considered (MIRANDA *et al.*, 2012; MIRANDA *et al.*, 2014; ISHIBUCHI; NOJIMA, 2013; AHMADIAN *et al.*, 2007; ENGEN *et al.*, 2009; OLIVEIRA *et al.*, 2005; ABBASS, 2003; KRAWCZYK; WOZNIAK, 2013; KRAWCZYK; WOŹNIAK, 2014; LÖFSTRÖM *et al.*, 2009). Many methods directly rely on non-convex performance metrics to optimize the models, thus motivating evolutionary-like approaches. However, machine learning problems are required to be scalable, and convexity of the learning loss is one of the most common strategies to achieve scalability. By imposing convexity, we rely herein on two deterministic multi-objective methods to search for efficient solutions: NISE (non-inferior set estimation (COHON, 1978)) and MONISE (many objective non-inferior set estimation (RAIMUNDO; VON ZUBEN, 2017)).

5.2 Classification

The learning process in classification consists in observing a set of samples/objects and using the characteristics of the samples to determine the membership of each sample to a class or concept. In biology, for example, we discriminate mammals for having hairs and producing milk; birds for having feathers, beaks and laying eggs; and reptiles for being covered with scutes, being cold-blooded and laying eggs. The existence of these characteristics is enough to discriminate most animals in these classes, exemplifying one of our classification processes. However, we need to create an automatic process that enables a machine to discriminate classes by presenting samples (with its respective labels). Those automatic procedures will be approached in this revision.

Starting with the most simple one, k **nearest neighbours** (kNN) consists in assigning the most frequent class between the k nearest samples to the target sample. This method usually uses the Euclidean distance of the feature vectors, but it can be changed depending on the nature of the dataset (for example, for categorical features it can be counted the number of feature matches).

Adding complexity to the model, each node of a **decision tree** contains a rule that forward the evaluated sample to different branches depending on its feature values, and the leaf of the tree is associated with a class. The rules are chosen to have the higher discriminative power in the samples of the training set, and the labeled samples forwarded to a leaf are used to choose the class as that leaf.

Logistic regression and multinomial regression (BISHOP, 2006) separates the samples of the classes using hyperplanes as boundaries with minimal misclassification. **Suport vector machines** (SVMs) (CORTES; VAPNIK, 1995) are similar to logistic regression, but

they create boundaries with maximum margins, which means that the boundary has the maximum distance to all samples. Moreover, SVMs are capable of exploring the kernel trick, which consists in projecting the samples to a higher dimensional space and create a linear separator. When mapped to the original dimension, the boundary becomes non-linear. **Neural Networks** is an intrinsic non-linear mapping constructed by composing decision functions in a structure of layers using the output of the functions of a previous layer as inputs to the next layer.

Another way to construct a classifier is by aggregating multiple classifiers. **Ensembles** take advantage of other classifiers to enhance some qualities of a classifier, for example improving bias with boosting or variance with bagging; other characteristics and properties of ensembles are described in Section 5.4. The proposed framework in this thesis is focused on guiding the creation of diverse logistic and multinomial regression models which can be used to build an ensemble or can be filtered using model selection.

5.3 Model selection [2]

Many of those classification methods have the training procedure, where given some hyper-parameters and the data, the model is adjusted to enable the prediction. Even when the training procedure is not needed, it is necessary to select the hyper-parameters. Highlighting some notable hyper-parameters we consider: k in kNN; the depth and the function that evaluates the splitting in decision trees; the regularization parameter in logistic/multinomial regression; kernel and regularization parameters in support vector machines; number of hidden layers and the type of activation function in a neural network; and number of ensemble components in ensembles.

These hyper-parameters are commonly chosen using knowledge of the expert or by selecting a specific value among a set of values (usually known as manual and grid search, respectively). This last approach can be recommended in low dimensional spaces due to the straightforward parallelization (BERGSTRA; BENGIO, 2012), and due to the simplicity of the procedure (CHANG; LIN, 2011). After selecting the values for each hyper-parameter, or selecting the creation rule, the learning machine is trained (or adjusted) using each candidate set of hyper-parameters, with the best model being selected according to a performance metric in the validation set. Some creation rules to implement grid search includes parameters linearly spaced (LAROCHELLE *et al.*, 2007), arbitrarily spaced (KRSTAJIC *et al.*, 2014), or exponentially spaced (HUANG *et al.*, 2012).

[2]This section is an amended version of Exploring multi-objective training in multi-class classification (RAIMUNDO *et al.*, 2018)

Apart from those simple approaches, more complex methods are usually supported by heuristic and statistical procedures, that conducts the searching by employing the evaluated parameter on each step, and the performance on validation being the guide to find better learning models. The heuristic approaches include a grid search refined with golden search (KULAIF; ZUBEN, 2013); genetic algorithms (CAMILLERI; NERI, 2014); and Nelder-Mead with paired tests (ZHENG; BILENKO, 2013). The statistical approaches include a search in estimated response surfaces (WEIHS *et al.*, 2005); a search in a surface estimated by Gaussian processes or a tree-structured Parzen estimator (BERGSTRA *et al.*, 2011); and a refined random search (BERGSTRA; BENGIO, 2012)

Instead of using a sampling procedure to acquire information about the statistical distribution and estimate it, this work explores another perspective in model selection: by assuming the problem as being multi-objective, it is only necessary to suitably sampling the Pareto front aiming at obtaining a diverse set of efficient learning models. Notice that each Pareto-optimal learning model will be associated with a distinct set of values for the hyper-parameters. Afterwards, it is possible to select the best model according to the performance in the validation set, and use it to make the prediction.

5.4 Ensembles[3]

Even with a good choice of classification and model selection methods, there are intrinsic challenges that single-model machine learning may not be able to surpass. Three of the main challenges that can be improved by the use of ensembles are: (1) scarcity of data: generalization guarantees are typically associated with a significant amount of data; (2) computational difficulties: even with a significant amount of data, the correct model hypothesis might not be found; (3) representational constraints: the subset of possible hypothesis of a statistical model might not contain the ideal hypothesis. The use of ensemble can surpass or at least alleviate the negative effects of these challenges by creating a set of diverse learning machines possibly founded on distinct datasets, distinct model hypothesis and with distinct statistical models that, when aggregated, can promote performance improvement.

A pertinent organization of an ensemble can the described by three steps: (1) generation: a procedure to create a set of learning machines that can compete as candidate components; (2) selection: these candidate components are selected aiming at improving the performance in the aggregation step; and (3) aggregation: this procedure creates a consensus response based on the outputs of the selected learning machines.

[3]This section is an amended version of Exploring multi-objective training in multi-class classification (RAIMUNDO *et al.*, 2018)

The proposed methodology mainly contributes to ensemble generation, only using ensemble filtering and aggregation as tools to build the learning machine. Given that, it is important to notice that, the primary goal in ensemble generation is to promote the diversity of the learning machines (COELHO, 2006). By granting diversity, it is possible to achieve robustness for different scenarios. There are four ways to generate ensemble diversity (ZHOU, 2012): (1) By manipulating the learning parameters. This case includes changing architecture, initialing neural networks with different values, use of different ramification rules in decision trees, and training learning machines with different types and levels of regularization; (2) By disturbing the outputs of some samples, or changing how the output is treated; (3) Training in distinct subsets of features; (4) By presenting distinct views of the training data.

Following the last methodology, bagging consists of training each learning machine with a random sampling of the training set (BREIMAN, 1996). In an iterative approach, boosting consists in using the current ensemble output and weight the samples to prioritize those that were miss-classified (SCHAPIRE, 2009). Random forests are used to train each learning machine by sampling both features and samples (BREIMAN, 2001). Bagging and boosting properties were experimentally evaluated (OPITZ; MACLIN, 1999), indicating that bagging always has a performance better than single learning machines, and boosting, despite having far superior performance in some cases, may guide to over-fitting.

As presented in Section 5.1, each learning machine in multi-objective ensemble generation methods is a trade-off between conflicting objectives: (1) prediction error $\times$ model complexity; (2) prediction error $\times$ cardinality of the training subset; (3) accuracy of distinct classes. The proposal of this thesis uses distinct sets of conflicting objectives, in response to the demands of each machine learning task.

5.5 Imbalanced classification

Imbalanced classification is a usual problem inside classification problems, and it occurs when there is a high distinction in the number of samples associated with each class. This can be a natural issue in scenarios which are implicitly imbalanced such as fraud detection, medical diagnosis, network intrusion detection, detection of oil spills and manufacturing issue detection (SUN *et al.*, 2009). The imbalance of a classification set can deteriorate the performance of a non-specialized classifier. Imbalance issues are generally detected around rates such as 1:10 (SUN *et al.*, 2009).

In those scenarios, it is necessary to create methodologies to handle this problem. Supported by the taxonomy explored in Sun *et al.* (2009), we wanted to discuss some of

those methodologies but focused on three types of proposals: sampling-level methodologies, cost-sensitive approaches, and ensemble/boosting approaches.

Sampling-level approaches consist in cleverly under-sampling the majority class samples and/or over-sampling the minority class samples. The main work on the over-sampling vein, called SMOTE, creates new samples by convexly combining minority samples with their neighbors of the same class (CHAWLA *et al.*, 2002). Each minority sample creates the same number of synthetic samples (CHAWLA *et al.*, 2002); it can be proportional to the ratio of majority samples in the neighborhood (HE *et al.*, 2008); or the generation can be constrained to the samples in the borderline with majority class samples (at least 50% of the neighbors) (HAN *et al.*, 2005).

The under-sampling usually removes some majority class samples from the training set. The selection of the percentage of random under-(and over) sampling can be selected by a grid search (CHAWLA *et al.*, 2002) or using a wrapper algorithm to select the amount of under-re-sampling and SMOTE over-sampling by firstly finding a valid under-sampling followed by a performance improvement SMOTE oversampling (CHAWLA *et al.*, 2005; CHAWLA *et al.*, 2008).

Cost-sensitive approaches consists of weighting the cost of miss-classification for each class and using these costs to guide the learning. The most naïve approach consists in weighting the classes inversely proportional to the frequency of the samples on each class (BRADFORD *et al.*, 1998), which can be accomplished by creating adjustable weight factors on each term of the class loss (LIN *et al.*, 2002; DATTA; DAS, 2015), by changing the boosting weight update to differently calculate the majority and minority class samples (CHAWLA *et al.*, 2003), and by adding class specific terms in the kernel calculation to become cost sensitive (MARATEA *et al.*, 2014).

The **boosting approaches** adapts each step of AdaBoost to correct the bias of the majority class. It can be done by modifying the weight updating to be cost sensitive (SUN *et al.*, 2005), by applying SMOTE to over-sample the minority class on each step (CHAWLA *et al.*, 2003), by under-sampling the majority class on each step (SEIFFERT *et al.*, 2008), by oversampling of the minority class with SMOTE on each step (CHEN *et al.*, 2010), but preferring samples with more neighbours in the majority class, and oversampling hard-to-learn samples (GUO; VIKTOR, 2004), that according to the authors are mostly from the minority class. Other **ensemble** approaches consists in creating each learning machine by sub-sampling only the majority class (LIU *et al.*, 2009b), by also removing the correctly classified samples from the majority class (LIU *et al.*, 2009b), and by bootstrap under-sampling followed by SMOTE over-sampling, thus creating balanced datasets.

The proposal of this thesis for imbalance classification is a particular case of the proposed framework, by imposing every multinomial regression loss of each class to be a conflicting objective. This gives more freedom to the model to find appropriate solutions given the level of imbalance (and even the level of preference of the user).

5.6 Multi-label classification[4]

Multi-label classification is a generalization of the conventional classification problem in machine learning when, instead of assigning a unique, relevant label for each object, it is possible to assign more than one label per object. A straightforward approach, called Binary Relevance (BR), ignores any possible relationship among the labels and learns one classifier per label, for example, using kNN with Bayesian inference (ZHANG; ZHOU, 2007). BR is computationally efficient, but it is not capable of exploring the relations among the labels to increase generalization. The main proposals devoted to promoting task relationship rely on Label Powerset, Classification Chains, and Multi-task Learning.

Label powerset consists in transforming the multi-label problem into a multi-class one by creating a class for each combination of original labels. Despite exploring the relationship of labels, this proposal promotes an exponential growth of classes in the multi-class equivalent problem. Some solutions for this issue were proposed: converting the powerset process in random subsets of labels which are aggregated by simple voting (TSOUMAKAS *et al.*, 2011); excluding the labels on the multi-class equivalent problem characterized by few objects (READ *et al.*, 2008); heuristically subsampling to overcome imbalanced data (CHARTE *et al.*, 2014).

Considering an ordered sequence of labels, **Classification Chains** create a sequence of classifiers, each one considering the predicted relevance of the labels provided by classifiers previously trained. The considered sequence can be nominal or random (READ *et al.*, 2011), and the architecture can be a tree instead of a sequence (RAMÍREZ-CORONA *et al.*, 2016), so that the prediction depends on the parents of the label. Also, the classification can be based on the relevance probability (DEMBCZY, 2010).

Multi-task learning creates binary relevance classifiers by jointly exploring the relation of labels by structure learning (CARUANA, 1997). This can be done by modeling the dependence among the labels using Ising-Markov Random Fields, further applied to restrict the flexibility of the task parameters adjustment (GONÇALVES *et al.*, 2015), or using

[4]This section is an amended version of Many-Objective Ensemble-Based multi-label Classification (RAIMUNDO; VON ZUBEN, 2018b)

a multi-target regression proposal that explores multiple output relations in data streams (JAPKOWICZ; MATWIN, 2015).

Other methods were considered to extend these main proposals. **Ensembles** were proposed to increase robustness by resampling (READ *et al.*, 2008); generating ensemble components using powersets in random sets of labels (TSOUMAKAS *et al.*, 2011), and filtering then using genetic algorithms and rank-based proposals (COSTA; COELHO, 2011); generating multiple classifiers by changing the label order in classification chains (READ *et al.*, 2011), and using many state-of-the-art multi-label classifiers to compose ensembles with different aggregation methods (TAHIR *et al.*, 2012). **Meta-learning** methods, instead of predicting the relevant labels found by binary relevance, predict the labels with higher membership degree, such that the number of predicted labels are estimated by a previously trained cardinality classifier (TANG *et al.*, 2009; SATAPATHY *et al.*, 2015), or by a fixed optimal number of labels (RAMÓN QUEVEDO *et al.*, 2012). **Multi-objective optimization** was used to: create ensembles by optimizing a novel accuracy metric that takes into account the correlation of the labels and a diversity ensemble metric using evolutionary multi-objective optimization (SHI *et al.*, 2011); train an RBF network considering different sets of performance metrics as conflicting objectives (SHI *et al.*, 2012; SHI *et al.*, 2014); and make feature selection in ML-kNN classifiers (YIN *et al.*, 2015).

In this thesis, we propose a novel ensemble method that uses a many-objective optimization approach to generate components exploring the relations among the labels (by weighted averaging the loss on each label), followed by a stacking method to aggregate the components for each label.

5.7 Multi-task learning[5]

Most of the multi-task learning methods encourage knowledge sharing by different types of regularization structures (CARUANA, 1998). When negative transfer is avoided, the constraints produced by regularization tends to promote generalization improvement for the learning tasks involved, when compared to what could be achieved by single-task learning. An algebraically effective strategy for knowledge sharing, when the loss function is linearly related to the parameter vector of each task, is achieved by parameter sharing, so that the parameter vector of a specific task may be conditioned by the parameter vector of the remaining tasks (CARUANA, 1998). However, parameter sharing causes the task losses to be conflicting with each other – the reduction in the loss function for one specific task

[5]This section is an amended version of Investigating multi-objective methods in multi-task classification (©2018 IEEE)(RAIMUNDO; VON ZUBEN, 2018a)

may increase the loss of a subset of some other tasks – creating an issue on how to weight the losses to promote the maximum generalization for that specific task.

Out of the scope of multi-task learning, a possible solution for the weighting problem involving the loss functions of the whole set of tasks was proposed in Engen *et al.* (2009), Wang *et al.* (2014), approaching a multi-class classification by considering the minimization of the multiple learning losses, one for each class, as conflicting objectives, thus resorting to a multi-objective optimization method. Supported by other scenarios when multi-objective optimization methods were used to solve machine learning problems (JIN; SENDHOFF, 2008; JIN *et al.*, 2009), this work also conceives the learning losses as conflicting objectives, but now under the framework of multi-task learning and explicitly adopting parameter sharing, a perspective that from the best of our knowledge still was unexplored.

With or without the multi-objective perspective, the main concern of multi-task learning is to promote joint improvement of performance/generalization of multiple tasks using structural modelling/learning and procedures devoted to knowledge sharing among the tasks. The main idea consists in proposing models which are linear in the adjustable parameters, such as linear regression, logistic regression, and support vector machines (SVMs), a property that is directly explored when designing the regularization components of the learning problem. Basically, those regularization components try to enforce similar tasks to have similar parameter vectors, using advanced matrix manipulations and norms (KIM; PAIK, 2014). The resulting learning problem may be convex or not. Convexity guides to more efficient solvers, while nonconvexity and /or scalability issues will generally require iterative and approximate solutions. As a sample of relevant solvers, we may cite: (1) the approach of Gong *et al.* (2013) to deal with non-smooth convex models; (2) iteratively finding new models to correct the error of previous predictions (CHAPELLE *et al.*, 2011); (3) using alternate optimization (iteratively fixing a subset of variables and optimizing another subset) to deal with more complex (usually bi-convex) models (ANDO; TONG, 2005); and (4) making model relaxations (CHEN *et al.*, 2009).

Among the most effective ways of imposing regularization is to force the reduction of the norm of the parameter vector, generally looking for sparse parameter vectors for the tasks. The most simple approaches restrain $\mathbf{\Theta}$ to: (*i*) create a task shared sparsity on the features (OBOZINSKI *et al.*, 2008; LIU *et al.*, 2009a; GONG *et al.*, 2013; KIM; PAIK, 2014), (*ii*) promote a general sparsity (KIM; PAIK, 2014; GONG *et al.*, 2013), (*iii*) look for a low-rank task shared space (KIM; PAIK, 2014), and (*iv*) encourage smoothness along neighbour tasks (ZHOU *et al.*, 2011).

Other mechanism is to model $\mathbf{\Theta}$ with the help of other auxiliary matrices and al-

gebraic properties. Using the additive approach ($\Theta = U + V$), each matricial parcel would extract different properties of the tasks and their relationship, including (*i*) shared sparsity on the features (GONG *et al.*, 2012; JALALI *et al.*, 2010); (*ii*) sparsity on the tasks, allowing parameters without parameter sharing for the non-zero tasks (allowing outlier tasks) (GONG *et al.*, 2012; CHEN *et al.*, 2011); (*iii*) a low rank task shared space (CHEN *et al.*, 2011; CHEN *et al.*, 2012); (*iv*) clustered tasks (EVGENIOU; PONTIL, 2004; ZHONG *et al.*, 2012); (*v*) general sparsity to allow outlier parameters (JALALI *et al.*, 2010; CHEN *et al.*, 2012); (*vi*) multiple models derived from multiple clustering results (HAN; ZHANG, 2015a); and (*vii*) a tree-structured clustering model in which models at higher levels contain the lower level clusters (HAN; ZHANG, 2015b).

The multiplicative approach ($\Theta = SV + U$) creates a shared subspace S, a task-specific projection of the subspace V, and an additive factor to detect outliers U. Different properties are applied to characterize the shared subspace: (*i*) orthonormality (ANDO; TONG, 2005; ARGYRIOU *et al.*, 2006; ARGYRIOU *et al.*, 2008; CHEN *et al.*, 2009; CHEN *et al.*, 2013; ZHONG *et al.*, 2016); (*ii*) low-dimensional shared subspaces (ANDO; TONG, 2005; CHEN *et al.*, 2009; CHEN *et al.*, 2013); and (*iii*) norms that promote sparsity in parameter sharing (ARGYRIOU *et al.*, 2006; ARGYRIOU *et al.*, 2008; ZHONG *et al.*, 2016).

Other proposals include the manipulation of the covariance matrix established by the parameter vectors of the multiple tasks (ZHANG; YEUNG, 2010; ZHANG; YEUNG, 2014; GONÇALVES *et al.*, 2014; CHARUVAKA; RANGWALA, 2015), the a priori imposition (LI; LI, 2007) or the online definition (YANG *et al.*, 2012) of the graph Laplacian, the online definition of task clusters (ZHOU *et al.*, 2011; ZHOU; ZHAO, 2016) and the use of multiple shared models (BAI *et al.*, 2009; CHAPELLE *et al.*, 2011; SIMM *et al.*, 2014).

In the ensemble vein, there are works that explore the intrinsic multi-task characteristic of neural networks (CARUANA, 1997) to propose a ensemble generation methodology for non-multi-task learning models (QIANG; MUNRO, 2006; WANG; ZHANG, 2010). To do that, it is proposed that the *i*-th component of the ensemble has as outputs the *i*-th feature of the problem and the output whose prediction is desired. This procedure is well succeeded because, when the neural network is forced to jointly learn another task (in this case, to learn a feature), a new bias is forced, generating ensemble diversity (WANG; ZHANG, 2010). Finally, it is important to highlight some methods that explore ensemble for multi-task learning such as: an adaptation of random forests (BREIMAN, 2001) for multi-task learning (SIMM *et al.*, 2014); an adaptation of *adaboost* for decision trees (QUINLAN, 1993), by modifying the rules of information (FADDOUL *et al.*, 2012) to integrate multiple tasks.

Transfer-learning is a related field that also explores knowledge sharing between

tasks, but instead of having equal importance tasks, there are source tasks T_i, $i \in \mathcal{S}$, that will provide knowledge to a target task T_t. Given that, transfer learning aims at improving the generalization capability of the target task using the knowledge contained in the instance, feature representation, parameter or relational knowledge of the source tasks (PAN; YANG, 2010). Given that, mainly in the context of inductive transfer learning (PAN; YANG, 2010), there are a few differences between the methodologies of transfer and multi-task learning. However, since transfer learning is solely focused in improving the predictive capability of the target task, any initiative to reach this objective is well fitted.

This work aims to study the impact of multi-objective optimization on multi-task learning founded on parameter sharing. Starting from all tasks having the same parameter vector, we generate multiple Pareto-optimal shared models from different views of the data tasks. In fact, our proposal can be properly characterized as being a many-objective formulation, because the loss function of each task is taken as a single objective. The methodology is promptly applicable to regression and classification problems, but here we will focus on classification. The Many-Objective Noninferior Set Estimation (MONISE) algorithm (RAIMUNDO; VON ZUBEN, 2017), which is an extension of the well-known NISE algorithm (COHON *et al.*, 1979) to deal with more than two objectives, is taken to automatically sample efficient solutions in the Pareto front. Notice that each efficient solution is a learning model resulting from attributing a distinct importance to the data coming from multiple tasks. Being a deterministic solver, MONISE will explore the particular conformation of the Pareto front toward a better distribution of the Pareto-optimal solutions, thus promoting very distinct perspectives for parameter sharing. Finally, for each specific task, a selected subset of the obtained learning models will compose an ensemble, creating a classifier.

5.8 Multi-view learning

The primary challenge of this field is to explore the multitude of data without overfitting, which could happen when the method concatenates the data features. This behavior can be avoided by constructing learners that integrate features without directly sharing them. The main strategies are guided by two principles (XU *et al.*, 2013): (*i*) consensus: that tries to maximize the agreement between the learning machines, and (*ii*) complementarity: that tries to use the pieces of information of one view to complement the features of another view.

The primary methods to deal with this problem consists of co-training: enhancing consensus by enforcing the classes from different views to be similar (FARQUHAR *et al.*, 2005), and enhancing complementarity by using the predicted labels of a classifier in one

view as training samples of another view (NIGAM; GHANI, 2000; WANG; ZHOU, 2007); kernel learning (MEMISEVIC *et al.*, 2012) and subspace learning (SHON *et al.*, 2006; XIA *et al.*, 2010). However, multi-view learning keeps some similarities with ensemble learning (XU *et al.*, 2013). Still, ensemble learning is applied to multi-view data by aggregating the votes of random forests associated with each view (FRATELLO *et al.*, 2017).

Many multi-view insights are contemplated by our proposal, although this multi-view approach is mainly based on a simple ensemble aggregation. The general purpose classifier tries to explore the common features along all views; the specific purpose classifier creates individual classifiers per view; and the transfer learning classifier tries to find a specific view regularized by the common features through all views. These approaches can explore many characteristics of the data, and the final classifier produced by the aggregation of all views benefits from this diversity.

5.9 Summarizing comments

The primary strength of the proposed methodology relies on its meta-learning behavior grounded on multi-objective concepts that we suppose to express good properties by creating a good sampling of the hyper-parameters as well as by producing diverse models. These properties make the proposed method similar to model selection and ensemble generation methods, differing from the literature by exploring the multi-objective characteristics of generalized linear models with regularization. Compared with multi-objective meta-learning methods, our methodology stands out. In addition to the characteristics mentioned above, it relies upon deterministic multi-objective algorithms, such as NISE and MONISE, reinforcing the applicability for convex machine learning problems.

Our methodology is also very general, being applicable to every formulation that can be expressed as a generalized linear model with regularization, by highlighting the conflicting objectives of the original problems. The combination of exploring the multi-objective nature of generalized linear models with regularization, as well as the combination of multiple models (each one corresponding to a trade-off solution of conflicting aspects of the learning process) with ensembles were shown to exhibit very attractive cost/benefit rates in a wide range of problems (multi-class classification, imbalanced classification, multi-label classification, multi-task learning, transfer learning, and multi-view learning), and may be interpreted as being a robust framework to deal with those and possibly many other problems in machine learning.

Chapter 6

Experiments

The usefulness of the multi-objective learning framework is attested by a series of relevant classification problems in machine learning. Experiments are designed to verify model selection, ensemble diversity, multi-class classification, class imbalance, multi-label classification, multi-task learning, and multi-view learning.

6.1 Multi-class classification [1]

Since Fernández-Delgado *et al.* (2014) have made an extensive experimental analysis, considering a wide range of classifiers as contenders, this section incorporates their results in further analyses that consider many aspects of multi-class classification. Remembering that in multi-class classification a sample can be assigned to a single class, the primary goal of this problem is to find classifiers with the highest rate of correct assignments. Taking into account that the proposed method generates multiple models using multiobjective optimization, this experimental design evaluates these models observing three aspects: (1) evaluating the quality of the model selection (Section 6.1.2); (2) exploring the diversity of the models and their effectiveness when composing an ensemble (Section 6.1.4); and (3) evaluating the quality of the resultant classification using distinct ensemble filtering and aggregation (Section 6.1.4). An additional fourth experiment deals with class imbalance (Section 6.1.5), scenario where the rate of correct assignments can be misleading, since the number of samples from a single class is so high that this evaluation can select classifiers incapable of learning the patterns of the class with a small number of samples. With this diversity of problems, we aimed at

[1] This chapter is an amended version of Exploring multi-objective training in multi-class classification (RAIMUNDO *et al.*, 2018)

investigating many aspects, qualities, and limitations of the proposed methodology.

With this diversity of problems, we aimed at investigating many aspects, qualities, and limitations of the proposed methodology.

6.1.1 Datasets description

This section explains and defines the datasets of the experiments of model selection, ensemble diversity, multi-class classification, and class imbalance. The database adopted for those experiments are based on Fernández-Delgado *et al.* (2014), in which a group composed of 121 datasets was used to benchmark multiple classifiers. Inside this group, 19 datasets have an additional separated test set.

The original data, except for four datasets, came from the UCI's repository[2]. They are also available in the author's site[3], as well as the Matlab scripts that pre-process the original data.

The experiments of **model selection**, **ensemble diversity**, and **multi-class classification** will follow the methodology in Fernández-Delgado *et al.* (2014) where it was used a four-fold cross-validation to generate four distinct sets, and each set has three-quarters of data kept to training/validation and one quarter to test. However, in the 19 datasets that have an additional separated test set, the one quarter test set designed by the previous procedure is replaced by the additional set.

An issue in the split procedure in Fernández-Delgado *et al.* (2014) might inflate the performance of some classifiers of that paper (WAINBERG *et al.*, 2016). It was used a 50/50 split of the complete dataset to make the parameter tuning procedure. Since this data is the same as the 4-fold splitting, some samples of the test set could have been used in the parameter tuning procedure, contaminating the whole methodology. Aiming at not falling in the same mistake, we used the dataset seen in training/validation in only the three quarters after the 4-fold splitting, adopting a 70/30 ratio for training and validation respectively.

We conducted the experiments of model selection (Section 6.1.2) and ensemble generation (Section 6.1.3) in the set of 19 datasets exhibiting a separated test partition; and the experiment of ensemble filtering and aggregation (Section 6.1.4) in the set of 121 datasets (which includes the 19 datasets with separated test partition).

It the experiment of **class imbalance** we separate 25% of the data for test in cases without a separate test set. The remaining data (75% of data without a separate test set and

[2]https://archive.ics.uci.edu/ml/index.html
[3]http://persoal.citius.usc.es/manuel.fernandez.delgado/papers/jmlr/

100% of data with a separate test set) will be split using the 75%/25% partition to training and validation respectively.

6.1.2 Model selection

6.1.2.1 Proposed method

To emphasize the importance of taking the multi-objective nature of the regularized multinomial regression into consideration for model selection, the following sequence of steps is proposed: (1) the models are formulated as the **conflicting regularized multinomial logistic regression** (Equation (4.6), Section 4.1.3); (2) P models are generated by NISE; (3) the best model on validation set is selected; (4) the accuracy in validation is compared to the one produced by other model selection techniques.

Remembering Equation (4.6):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}), \tag{4.6 revisited}$$

6.1.2.2 Baseline

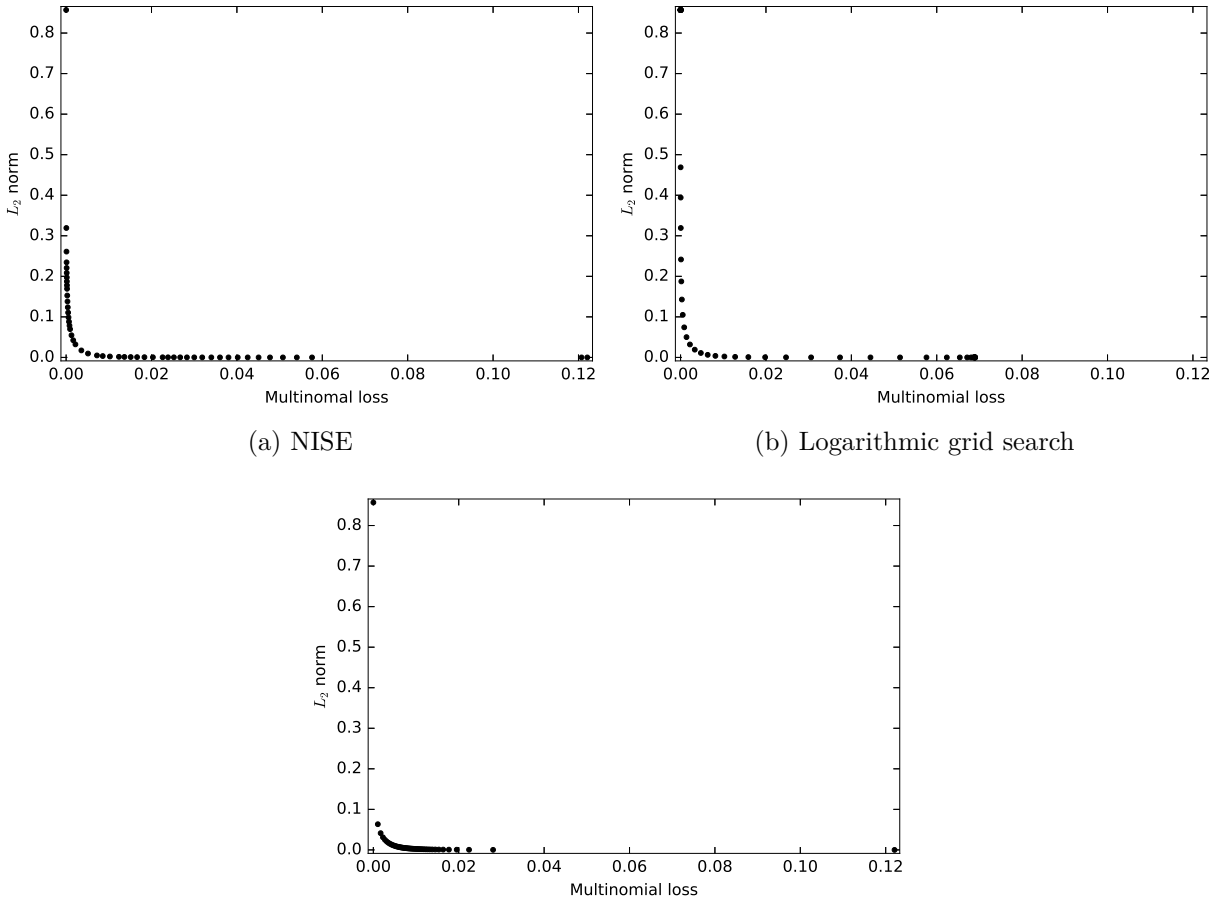
The algorithms used to make the **comparison** involves two types of grid search and two types of global search. The first grid search (called logarithmic grid search) consists in evaluating models taking constant steps on a logarithmic scale ($\lambda \in \{2^{-\frac{P}{2}}, 2^{-\frac{P}{2}+1}, \ldots, 2^{\frac{P}{2}-1}, 2^{\frac{P}{2}}\} \cup \{0\}$) (HUANG *et al.*, 2012). The second grid search (called constant grid search) takes constant steps varying $\mathbf{w}_2$ on the set $\{0, \frac{1}{P-1}, \ldots, \frac{P-2}{P-1}, 1\}$. Finally, two zero-order hyperparameter optimizers were applied, the Nelder-Mead global optimization method and a statistical method based on tree-structured Parzen estimator and random search, which is called hyperopt[4]. Both approaches are restricted to P evaluations, and the performance index consists in the accuracy in the validation set of a model using λ as the regularization parameter.

Those methods were **evaluated** in each of the four folds on the 19 datasets that contain an independent test set. The best model found by those techniques are compared constraining P to 50, 25, and 10 evaluations. The performances in all datasets for all four replications were compared using Friedman test (FRIEDMAN, 1937), with $p = 0.01$ as a threshold to indicate the statistical difference, and using Finner *posthoc* test (FINNER, 1993) with the same threshold.

[4]Available at <https://github.com/hyperopt/hyperopt>

6.1.2.3 Results

Figures 19 and 20 show the Pareto front (Multinomial loss vs L_2 norm of the vector of parameters) for two different datasets using NISE, logarithmic constant step grid search (log grid) and constant step grid search (grid). Confirming the hypothesis presented in Section 2.1, these scenarios show that NISE clearly creates a better representation of the Pareto front, which can help on finding a richer set of non-dominated models.



(a) NISE

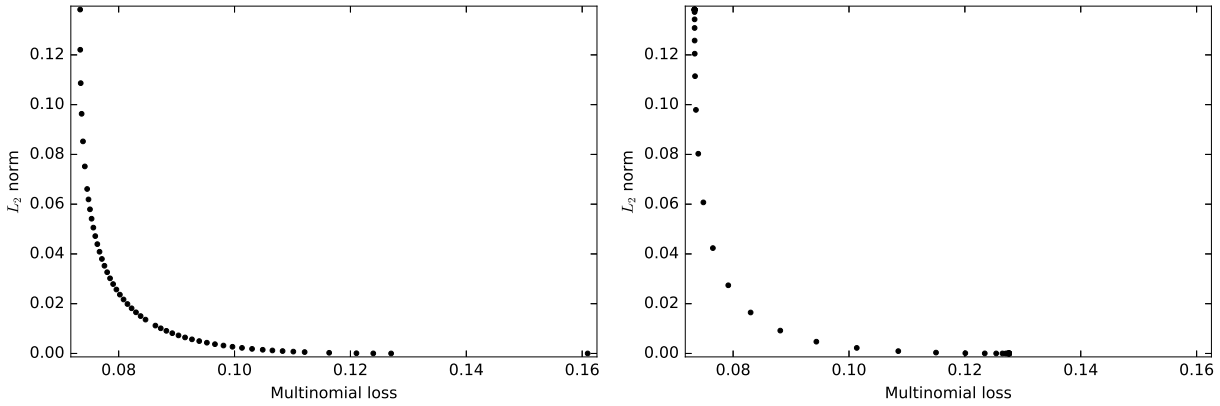
(b) Logarithmic grid search

(c) Constant grid search

Figure 19 – Pareto front representation for the *low-res-spect* dataset.

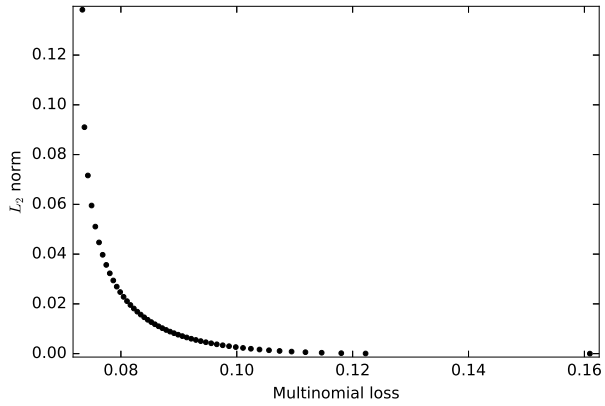
To provide a stronger evidence supporting the relevance of NISE for model selection, we used the test proposed in Section 6.1.2. In this experiment, we compare NISE, logarithmic grid search, constant grid search, Nelder-Mead and hyperopt for 10, 25 and 50 evaluations, analyzing if there is any statistical difference with Friedman test (threshold of $p = 0.01$), and all versus all with Finner post-hoc test (threshold of $p = 0.01$).

Since the Friedman test rejected the null hypothesis, Table 1 summarizes the post-hoc Finner test comparisons. The table provides information for each evaluated method



(a) NISE

(b) Logarithmic grid search



(c) Constant grid search

Figure 20 – Pareto front representation for the *heart-cleveland* dataset.

with corresponding number of evaluations (indicated in the **method** and **evals** columns): the average rank (in the **rank** column); the number of methods better than the evaluated method (in the #< column); the number of methods worse than the evaluated method (in the #> column). This ordering relation (better and worse) is accounted only if there is statistical significance according to the Finner post-hoc test.

The rows of Table 1 are sorted by the average rank, and the rank is directly used by the Finner post-hoc test, as such, the #< column indicates how many methods of the rows above the current row are better than the method at that row (e.g., the first four methods are better than *const grid 50*), and the #> column indicates how many methods of the rows below the current row are worse than the method at that row (e.g., the last six methods are worse than *const grid 50*).

Table 1 – Statistical comparison involving five model selection methods with three different number of evaluations.

method	evals	rank	#<	#>
NISE	50	6.25	0	10
log grid	50	6.46	0	10
Hyperopt	50	6.71	0	10
NISE	25	6.73	0	10
log grid	25	6.97	0	9
const grid	50	7.70	4	6
Hyperopt	25	7.77	5	6
NISE	10	7.81	5	6
const grid	25	8.27	5	3
log grid	10	8.75	8	1
Nelder-Mead	50	8.80	8	1
Nelder-Mead	25	8.93	8	1
const grid	10	9.24	9	1
Hyperopt	10	9.45	9	0
Nelder-Mead	10	10.15	13	0

6.1.2.4 Discussion

These results show the usefulness of an effective multi-objective optimization technique for model selection. The quality of our technique is highlighted by the fact that NISE, with the limit of 50 evaluations, was never outperformed in all tested scenarios. Moreover, notice that NISE, with the limit of 25 evaluations, even though being the fourth in the ranking, exhibits results with no statistical difference to the three best methods. Hence, fewer evaluations are admissible for this technique.

6.1.3 Ensemble generation

6.1.3.1 Proposed method

In this section, instead of choosing the best classifier given a performance metric, the models generated by the optimizer are used to compose an ensemble. One of the most important aspects in ensemble generation is linked to the capability of generating multi-objective diversity. Given a set of classifiers, the diversity measure should capture the difference between those classifiers.

After formulating the models as the **conflicting regularized multinomial logistic regression** (Equation (4.6), Section 4.1.3). We create two versions of the **proposed** method: consider the 10 first models generated by NISE; and choose the 10 most diverse models,

between 50 models generated by NISE, and solving the maximum diversity filtering problem (Definition 4.1).

Remembering Equation (4.6):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}), \tag{4.6 revisited}$$

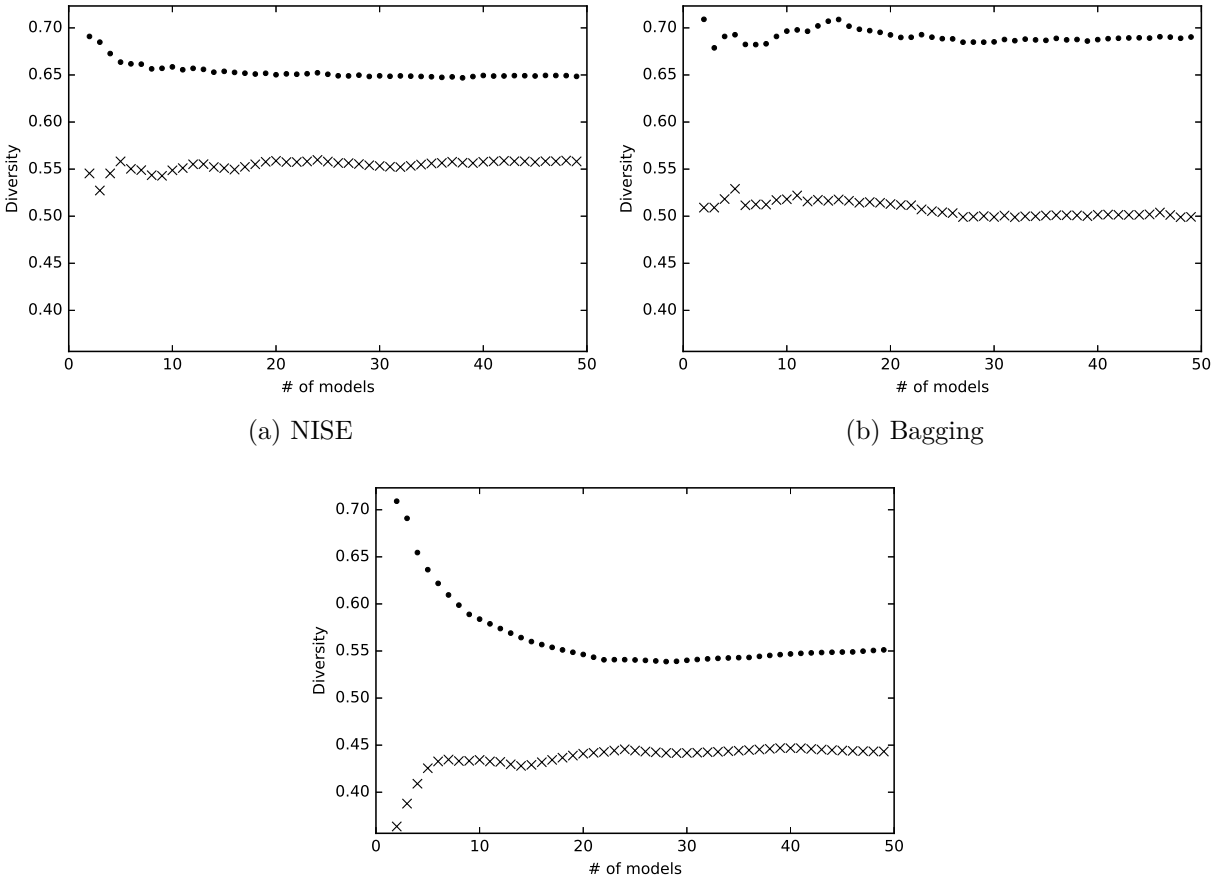
6.1.3.2 Experimental setup, baselines and evaluation metrics

These methods are compared **against** bagging and boosting with 10 components. All methods used a multinomial regression as the base classifier. To make a comparison between ensemble generators, the mean some-correct (Equation (4.17)) and mean one-correct measures were used. Those methods were **evaluated** in each of the four folds on the 19 datasets that contain an independent test set. The evaluated mean some-correct diversity measure for the three ensemble generation methods was compared using Friedman test (FRIEDMAN, 1937), with $p = 0.01$ as a threshold to indicate a statistical difference, and using Finner *posthoc* test (FINNER, 1993) with the same threshold.

6.1.3.3 Results and discussion

Figure 21 shows the evolution of both diversity measures (some-correct and both-correct) when increasing the number of components, for NISE, bagging and boosting ((a) to (c), respectively), using multinomial regression as base classifiers over the *heart-cleveland* dataset. It can be seen that the multi-objective procedure has a competitive performance when compared to bagging (being slightly worse in some-correct and slightly better in both-correct metrics) and having a consistently better performance than boosting. Those results can be explained by the behavior of each algorithm: bagging is based on randomness; it allows this procedure to generate different components by random re-sampling. Boosting is the opposite, it is focused on improving the accuracy of the composed ensemble by inserting new biased components. In turn, NISE searches for components with different complexity×accuracy trade-offs. Given that, NISE is not only focused on accuracy; it also tends to find diverse components by properly sampling the Pareto front.

The next experiment shows that NISE is not only capable of generating diverse components but also exhibits a consistent behavior across several datasets. First, 50 models are generated using the NISE method. After that, the 10 best models are selected using maximum diversity filtering (Definition 4.1) as described in Section 4.3. We refer to this approach



(a) NISE

(b) Bagging

(c) Boosting

Figure 21 – Evolution of the some-correct (•) and both-correct (×) diversity measures by increasing the number of generated components for *heart-cleveland diversity* dataset

as tuned multi-objective component generator. The first 10 models generated by the multi-objective procedure are denoted regular multi-objective components.

Figure 22 shows a bar graph where each group of four bars is a comparison between four ensemble generation methods, for each of the 19 datasets with test partition, all labeled in the abscissa. The gray bar is the double-correct measure, and the black bar on top of it is one-correct measure. Their sum, indicated by the top of the bar, corresponds to the some-correct measure. Each bar from the dataset group corresponds to a different method. From left to right: boosting, bagging, the regular multi-objective component generator using NISE, and the tuned multi-objective component generator using NISE. One of the approaches using NISE is always the best one in terms of diversity, or equivalent to other approaches, except for a single dataset (*hayes-roth*).

To further enhance the comparative analysis, a Friedman test followed by a Finner

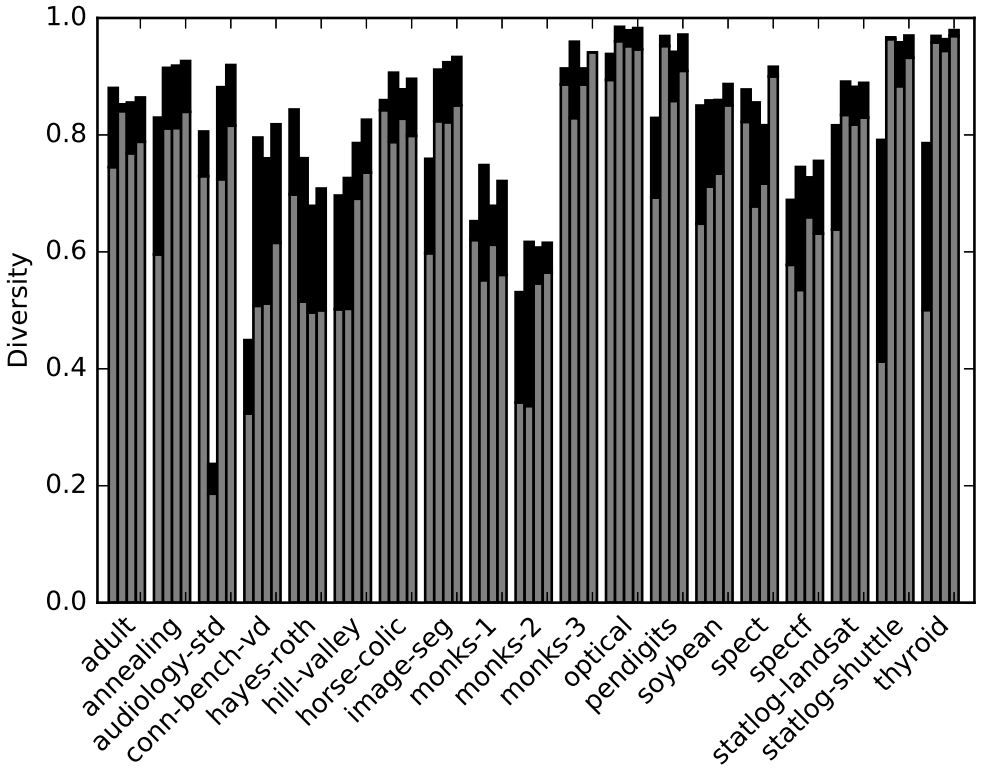


Figure 22 – Bar chart comparing the diversity behavior of four techniques devoted to ensemble generation. From left to right, bars correspond to: boosting, bagging, regular multi-objective component generator using NISE, and tuned multi-objective component generator using NISE.

post-hoc test was applied, covering all folds for all datasets. Tuned multi-objective is better than all the others; boosting is worse than all the others; and bagging is better than regular multi-objective.

The competitiveness of regular and tuned multi-objective using NISE is enforced by the necessary association of multi-objective solutions with efficient solutions, a favorable condition to enhance the whole performance of the ensemble.

6.1.4 Ensemble filtering and aggregation

6.1.4.1 Proposed method

Supported by the diversity expected (and observed in the experimental studies) from our methodology, we **propose** a learning machine resulting from the following sequence of steps: (1) the models are formulated as the **conflicting regularized multinomial logistic**

regression (Equation (4.6), Section 4.1.3); (2) the ensemble generation of 50 models using NISE; and (3) the final predictor can be a selected model (as discussed in Section 6.1.2) or an ensemble composition (as described in section 6.1.3). Two steps are involved: (1) filtering: first some components are selected, using all filters described in Section 4.3; and (2) aggregation: after filtering, simple voting and distribution summing, described in Section 4.3, are used, except for winner takes all that only uses simple voting.

Remembering Equation (4.6):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}), \qquad \text{(4.6 revisited)}$$

6.1.4.2 Experimental setup, baselines and evaluation metrics

Afterward, those models were **compared with 179 classification** methods evaluated in Fernández-Delgado *et al.* (2014), on their experimental setup. To deal with missing data (absence of results for some algorithms), the comparison was made using Skilling-Mack test (SKILLINGS, 1981), $p = 0.01$ as the threshold, for accuracy and kappa metrics (COHEN, 1960). Given an **evaluation** metric for the performance of a classifier in a dataset, this measure creates a rank by sorting classifiers for a specific dataset, replacing the rank of the missing value by a median rank for that block. Then, it is created a table sorted by the mean, presenting how many algorithms are worse or better than the algorithm under analysis, considering a statistical significance of $p = 0.01$ as the threshold.

6.1.4.3 Results

Supported by the results in Section 6.1.2.3 and Section 6.1.3.3, we have evidence that the proposed framework is suitable for model selection and diversity generation. We then compare the classifiers based on generating components with multi-objective optimization (presented in Section 6.1.4) with multiple classifier methodologies with alternative implementations cataloged in Fernández-Delgado *et al.* (2014). The description and names of the classifiers not proposed in this paper are listed in reference Fernández-Delgado *et al.* (2014). Using this description it is possible to catalog the methods that possibly had the performance inflated, in the results reported in Wainberg *et al.* (2016), by hyperparameter tuning being done in a set with some samples of the test set.

Since the Skilling-Mack test rejected the null hypothesis, Tables 2 and 3 summarizes the post-hoc Finner test comparisons. The tables present information for each evaluated method (indicated in the **method** column): the average rank (in the **rank** column); the

Table 2 – Friedman rank (average) considering the accuracy metric. Top 60 out of 190 classifiers (proposed methods in bold).

method	rank	#<	#>	method	rank	#<	#>
rf_caret*	37.75	0	164	pnn_matlab*	63.78	7	110
parRF_caret*	38.03	0	162	cforest_caret*	63.79	7	110
svm_C*	40.69	0	161	gausssprRadial_R*	64.45	7	107
svmPoly_caret*	41.50	0	160	**wta_svote**	64.46	7	107
elm_kernel_matlab*	43.40	0	160	RandomForest_weka	65.14	7	107
svmRadialCost_caret*	43.47	0	160	svmLinear_caret*	66.56	12	102
rforest_R	44.38	0	160	dkp_C*	66.62	12	102
svmRadial_caret*	46.10	0	155	MultiBoostAB_RandomForest_weka	67.50	12	101
elite_svote	46.14	0	155	mlp_C*	68.97	16	98
elite_dsum	46.24	0	155	fda_caret	69.20	16	96
elitePL_dsum	47.13	0	155	RandomCommittee_weka	69.47	16	96
elitePL_svote	47.14	0	155	knn_caret*	69.49	16	96
max-div_svote	48.83	0	152	mlpWeightDecay_caret*	69.59	17	96
C5.0_caret*	48.93	0	152	Decorate_weka	69.78	18	95
avNNet_caret*	49.01	0	152	MultiBoostAB_MultilayerPerceptron_weka	70.43	18	94
moPL_dsum	49.11	0	152	rda_R*	70.97	20	94
max-div_dsum	50.38	0	148	gcvEarth_caret	71.34	22	94
nnet_caret*	50.49	0	147	multinom_caret*	71.68	22	94
wtaPL_dsum	51.39	0	145	knn_R*	72.11	23	93
Bagging_LibSVM_weka*	51.58	0	145	MultiBoostAB_PART_weka	72.20	23	93
moPL_svote	51.99	0	144	glmnet_R	72.43	24	92
pcaNNet_caret*	52.12	0	144	treebag_caret	72.56	24	92
mlp_caret*	52.83	0	142	svmlight_C*	72.60	24	92
RotationForest_weka	53.29	0	140	mda_caret	72.66	24	92
wtaPL_svote	55.52	0	133	ClassificationViaRegression_weka	72.67	24	92
RRF_caret*	55.81	0	132	Bagging_PART_weka	73.12	24	91
MultiBoostAB_LibSVM_weka*	57.12	1	128	elm_matlab*	74.09	24	91
RRFglobal_caret*	57.17	1	128	SimpleLogistic_weka	74.75	25	89
LibSVM_weka*	58.29	2	126	pda_caret*	75.03	26	88
adaboost_R	60.42	3	119	rbfDDA_caret*	75.59	26	86

*Methods with hyperparameter tuning done in a set with some samples of the test set, a process that, in some sense, leads to contamination of training/test samples, as reported in Wainberg *et al.* (2016).

number of methods better than the evaluated method (in the #< column); and the number of methods worse than the evaluated method (in the #> column). This ordering relation (better and worse) is accounted only if there is statistical significance according to the Finner post-hoc test. Also, the rows of Tables 2 and 3 are sorted by the average rank, columns #< and #> follow the same interpretation adopted in Table 1.

In Tables 2 and 3, the proposed methods are highlighted in bold and the names follows the structure **filtering_aggregation** whose codes were presented in Section 4.3. Also, there are some methods marked with a footnote (e.g., *parRF_caret**) expressing the methods with performance possibly inflated by having access to some samples of the test set (WAINBERG *et al.*, 2016).

6.1.4.4 Discussion

It is possible to see that our methods have a comparable performance with the best classifiers, even with the classifiers with an inflated performance (WAINBERG *et al.*, 2016). We can see that only the ensemble with a single component (**wta_svote**) is worse than the

Table 3 – Friedman rank (average) considering the kappa metric. Top 60 out of 190 classifiers (proposed methods in bold).

method	rank	#<	#>	method	rank	#<	#>
parRF_caret*	40.48	0	164	MultiBoostAB_PART_weka	63.74	3	111
rf_caret*	40.58	0	164	RandomCommittee_weka	66.47	4	104
svm_C*	43.88	0	161	MultiBoostAB_J48_weka	67.63	7	101
rforest_R	46.81	0	159	treebag_caret	67.64	7	101
elite_dsum	47.50	0	158	Bagging_PART_weka	67.80	7	101
mlp_caret*	47.78	0	158	LibSVM_weka*	67.82	7	101
elite_svote	48.27	0	158	MultiBoostAB_LibSVM_weka*	67.84	7	101
elitePL_dsum	49.07	0	151	RandomForest_weka	68.22	7	100
nnet_caret*	49.32	0	151	fda_caret	68.22	7	100
elitePL_svote	49.63	0	150	AdaBoostM1_J48_weka	68.69	9	97
elm_kernel_matlab*	50.13	0	149	rda_R*	69.05	10	97
C5.0_caret*	50.14	0	149	**wta_svote**	69.74	12	96
svmPoly_caret*	50.66	0	148	MultiBoostAB_RandomForest_weka	69.99	13	96
moPL_dsum	50.89	0	147	Bagging_RandomTree_weka	70.34	14	95
max-div_svote	51.73	0	145	mlp_C*	70.79	14	95
avNNet_caret*	51.79	0	145	gcvEarth_caret	71.02	16	95
svmRadialCost_caret*	52.28	0	144	multinom_caret*	72.10	19	95
RRF_caret*	52.29	0	144	MultilayerPerceptron_weka	72.75	19	92
wtaPL_dsum	52.92	0	144	Bagging_J48_weka	73.18	19	91
pcaNNet_caret*	54.08	0	140	svmLinear_caret*	73.19	19	91
moPL_svote	54.20	0	139	mda_caret	73.27	20	91
max-div_dsum	54.39	0	139	mlpWeightDecay_caret*	73.57	22	91
RRFglobal_caret*	54.60	0	138	gausssprRadial_R*	74.31	23	90
svmRadial_caret*	55.71	0	137	MultiBoostAB_RandomTree_weka	74.96	24	90
wtaPL_svote	57.67	0	128	pda_caret*	75.43	24	89
RotationForest_weka	59.18	0	126	pnn_matlab*	75.45	24	89
logitboost_R	62.13	2	115	ClassificationViaRegression_weka	75.50	24	88
adaboost_R	62.52	2	114	glmnet_R	75.66	24	88
MultiBoostAB_MultilayerPerceptron_weka	62.73	2	114	SMO_weka	75.75	24	88
Decorate_weka	63.48	3	111	knn_caret*	75.81	24	88

*Methods with hyperparameter tuning done in a set with some samples of the test set, a process that, in some sense, leads to contamination of training/test samples, as reported in Wainberg *et al.* (2016).

best classifiers from the literature, with statistical significance. All ensembles are comparable with the best classifiers from the literature. It is important to notice that there is only one method without sample contamination (*rforest_R*) better than our best method (**elite_dsum**). Furthermore, the number of methods worse than these methods with the highest performance, are similar (160 versus 155 for the accuracy and 159 versus 158 for the kappa metric) strengthening the hypothesis that our method has a competitive performance when compared to the best-ranked classifiers from the literature.

6.1.5 Imbalanced classification

6.1.5.1 Proposed method

To explore the potential of the multi-objective framework in the imbalanced scenarios, instead of using the simple regularized multinomial model, we are going to adopt the multinomial model with the loss of every class as conflicting objectives (called **class-conflicting regularized multinomial logistic regression** and presented in Equation (4.7), Section

4.1.3).

Remembering Equation (4.6):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}), \tag{4.6 revisited}$$

and Equation (4.7):

$$\min_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K} w_k \left[- \sum_{i=1}^{N} y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_{K+1} r(\boldsymbol{\theta}), \tag{4.7 revisited}$$

This formulation is used to generate 50 efficient models using MONISE, that we call here as **manyobj**. However, since it could not find the models with a flat preference of classes (models with equal preference for all objectives), as well as a preference that balances the importance with the inverse of the number of samples for each class, we created an approach called **all** with 50 models of each approach (manyobj, standard multinomial models, and balanced multinomial models). The standard multinomial models are formulated with the **regularized multinomial logistic regression** (Equation (4.6)) with $u_k = 1$, and balanced multinomial models are formulated with the **regularized multinomial logistic regression** (Equation (4.6)) with $u_k = n_k$; both methods are trained using NISE and generating 50 models. To show a baseline the approach **both** used models coming from standard multinomial models, and balanced multinomial models.

6.1.5.2 Experimental setup, baselines and evaluation metrics

The algorithms used to make the **comparison** are the most relevant in imbalanced classification literature[5]: SMOTE, ADASYN, ENN, Tomek-links, ENN, SMOTEENN, SMOTETL, SMOTEBoost, RAMOboost, random undersampling and random oversampling. Given that those methods actuate as meta-learners, we used a balanced regularized multinomial regression as its base classifier, and following a similar procedure of (SÁEZ *et al.*, 2015), the number of neighbors was kept as $k = 5$, and the over-sampling and under-sampling was targeted to achieve $\frac{N}{K}$ samples. Since the majority of classes were not changed in the over-sampling (and the minority in the under-sampling case), this procedure only reduces the level of imbalance, but it did not make the dataset completely balanced. Also, after the sampling the training procedure evaluates models taking constant steps on a logarithmic scale ($\lambda \in \{2^{-\frac{P}{2}}, 2^{-\frac{P}{2}+1}, \ldots, 2^{\frac{P}{2}-1}, 2^{\frac{P}{2}}\} \cup \{0\}$) (HUANG *et al.*, 2012); and we used cross-validation to select the number of candidate models in the boosting approaches.

[5]Available at <contrib.scikit-learn.org/imbalanced-learn>

Table 4 – Friedman rank (average) considering the gmean, kappa and F1 metric for all datasets.

	gmean			kappa			F1		
	rank	#<	#>	rank	#<	#>	rank	#<	#>
balanced	6.66	0	14	8.53	0	6	8.65	0	5
imbalanced	11.23	8	4	8.70	0	6	8.75	0	5
wta_manyo	13.58	15	0	14.09	14	0	14.52	15	0
elite_manyo	14.91	15	0	14.79	15	0	15.63	16	0
wta_both	6.89	0	14	7.91	0	7	8.21	0	6
elite_both	7.07	0	12	8.09	0	7	8.02	0	6
wta_all	7.17	0	12	7.96	0	7	8.38	0	6
elite_all	7.07	0	12	8.13	0	7	8.04	0	6
ENN	10.08	5	4	10.35	7	4	9.79	0	5
random_us	10.36	5	4	8.69	0	6	8.44	0	6
TL	10.50	5	4	8.24	0	7	8.33	0	6
smote	9.18	5	5	8.21	0	7	8.29	0	6
adasyn	9.33	5	4	8.95	0	6	8.78	0	5
random_os	8.90	2	5	8.15	0	7	7.79	0	6
SMOTETomek	8.97	2	5	8.69	0	6	8.52	0	6
SMOTEENN	9.82	5	4	11.00	12	4	10.61	9	4
RAMOBoost	13.85	15	0	13.68	14	0	13.67	14	0
SMOTEBoost	13.66	15	0	13.43	14	0	13.49	14	1
EE	10.66	5	4	12.30	12	1	11.99	13	2

Those methods were **evaluated** in all 121 datasets with 25% for the test set and 25% of the rest of the dataset separated to validation. The evaluation was done using distinct metrics: kappa, gmean (it consists in the geometric mean of the recall for every class) and F1. The same metric used to present the performance was also presented beforehand to select the models. The performances in all datasets were compared using Friedman test (FRIEDMAN, 1937), with $p = 0.01$ as a threshold to indicate the statistical difference, and using Finner posthoc test (FINNER, 1993) with the same threshold.

6.1.5.3 Results

The Friedman test rejected the null hypotheses for the class imbalance experiment. The post-hoc Finner test was summarized by Tables 4, 5 and 6, which show, respectively, the experiment with all datasets; all datasets with weighted Friedman test; and for the 20 most imbalanced datasets. The weighted Friedman test consists in weighting the average rank with the imbalance-degree metric with total variance (ORTIGOSA-HERNÁNDEZ *et al.*, 2017), and the imbalance-degree metric also selects the most imbalanced datasets with total variance (ORTIGOSA-HERNÁNDEZ *et al.*, 2017). The tables present information for

Table 5 – Friedman rank (weighted average) considering the gmean, kappa and F1 metric for all datasets. The Friedman rank is weighted by the imbalance-degree metric with total variance (ORTIGOSA-HERNÁNDEZ *et al.*, 2017).

	gmean			kappa			F1		
	rank	#<	#>	rank	#<	#>	rank	#<	#>
balanced	5.92	0	14	9.49	0	5	9.20	0	5
imbalanced	12.83	12	0	9.10	0	5	9.32	0	5
wta_manyo	13.38	13	0	13.69	14	0	14.64	16	0
elite_manyo	14.40	14	0	14.38	14	0	15.59	17	0
wta_both	6.08	0	14	7.55	0	7	8.16	0	6
elite_both	6.82	0	14	8.40	0	6	8.15	0	6
wta_all	6.35	0	14	7.64	0	7	8.16	0	6
elite_all	6.75	0	14	8.40	0	6	8.06	0	6
ENN	9.90	5	6	9.83	2	5	9.02	0	5
random_us	11.27	7	4	9.19	0	5	8.84	0	5
TL	11.91	11	1	8.36	0	6	8.61	0	5
smote	9.59	5	6	8.20	0	6	8.60	0	5
adasyn	9.64	5	6	9.03	0	6	9.04	0	5
random_os	8.92	5	7	8.35	0	6	7.79	0	6
SMOTETomek	9.37	5	6	9.05	0	6	8.44	0	6
SMOTEENN	9.10	5	7	11.10	9	2	10.64	6	3
RAMOBoost	13.74	13	0	12.82	13	0	12.71	14	1
SMOTEBoost	13.46	13	0	12.38	13	0	12.43	13	2
EE	10.48	5	5	12.93	13	0	12.49	13	2

each evaluated method (indicated in the **method** column): the average rank (in the **rank** column); the number of methods better than the evaluated method (in the #< column); and the number of methods worse than the evaluated method (in the #> column). This ordering relation (better and worse) is accounted only if there is statistical significance according to the Finner *posthoc* test.

In Tables 4, 5 and 6, the proposed methods are highlighted in bold and the names follow the structure **filtering_generation**. The filtering codes were presented in Section 4.3, and the generation was presented in Section 6.1.5.

6.1.5.4 Discussion

First of all, the traditional *ad-hoc* balancing performs well in the gmean metric in all scenarios, and any change focused on dealing with imbalance datasets does not improve much performance. In the kappa and F1 metric, it is possible to observe an improvement from some imbalance methods coming from the literature (TL, smote, random oversampling), as well as

Table 6 – Friedman rank (average) considering the gmean, kappa and F1 metric for the 20 most imbalanced datasets (according to imbalance-degree metric with total variance (ORTIGOSA-HERNÁNDEZ *et al.*, 2017).

	gmean			kappa			F1		
	rank	#<	#>	rank	#<	#>	rank	#<	#>
balanced	5.74	0	8	11.27	0	0	10.12	0	1
imbalanced	12.52	3	0	9.27	0	1	9.90	0	1
wta_manyo	14.12	7	0	13.57	2	0	14.99	10	0
elite_manyo	14.59	7	0	15.32	9	0	16.32	15	0
wta_both	5.52	0	8	6.59	0	3	8.15	0	3
elite_both	7.65	0	3	9.74	0	0	8.44	0	3
wta_all	5.54	0	8	6.17	0	3	7.74	0	3
elite_all	7.45	0	3	9.34	0	1	8.19	0	3
ENN	10.55	0	0	8.67	0	2	8.74	0	3
random_us	12.44	3	0	10.02	0	0	10.30	0	1
TL	12.17	3	0	8.47	0	2	8.44	0	3
smote	9.22	0	0	7.95	0	2	8.77	0	3
adasyn	9.59	0	0	8.02	0	2	8.49	0	3
random_os	7.45	0	3	8.30	0	2	6.99	0	3
SMOTETomek	9.59	0	0	9.57	0	0	8.17	0	3
SMOTEENN	8.02	0	2	11.90	0	0	10.34	0	1
RAMOBoost	13.19	6	0	9.94	0	0	9.92	0	1
SMOTEBoost	12.82	3	0	11.07	0	0	10.90	0	0
EE	11.72	3	0	14.72	7	0	14.97	10	0

some proposed methods (*_both and *_all). This statement shows that proper manipulation of the weights in the optimization of an L_2 regularized multinomial logistic regression helps to build good classifiers in imbalanced datasets.

It is important to notice that the methods **both** are ensembles composed of balanced and imbalanced models. Those methods are straightforward and consistent and achieve this excellent performance by only changing the objective function. The methods generated by many-objective (**manyo**) training were not capable of achieving a good performance; however, when they are associated with balanced and imbalanced models (**both**) they improve the performance of the classifier. The result for the most imbalanced datasets, depicted in Table 6 for kappa and F1 metrics, is the scenario where this quality is most prevalent, showing a more profound relevance. This profile of performance can be explained by the fact that this kind of optimization might not be able to find more trivial models (no matter their intrinsic performance) and is capable of finding more challenging models (when the trivial models were not enough to achieve a good performance).

6.2 Detection of epileptic seizures [6]

In this section, we focus on the problem of seizure detection in epileptic brain recordings. Seizures, being scarce events, pose a defiance to machine learning algorithms both regarding biological variability of their types and also regarding the formation of datasets with balanced sample classes.

Concerning the nature of the features that are usually passed to such classifiers, there is no consensus: energies of subbands of the Fourier spectrum have been shown to be descriptive of seizure events (SHOEB; GUTTAG, 2010) as well as wavelet expansions (LATKA *et al.*, 2003), entropic measures, and other linear and nonlinear features (GIANNAKAKIS *et al.*, 2015). More recently, there is an increasing interest in extracting features capable of unveiling the interdependence between multiple channels' recordings, forming synchronization graphs (DHULEKAR *et al.*, 2015). All these extraction methods can be used to generate explanatory features that a learning machine would make use to detect seizures. Depending on particular characteristics of the state of the patient under evaluation, some feature extraction methods may outperform others. These considerations motivate us to explore multiple extractions and construct ensemble approaches, which are capable of weighting multiple points of view, automatically, giving preference to distinct classifiers depending on how each sample is positioned in the feature space.

Given the multitude of patients and extractions, two experiments were conducted subject to distinct characteristics of the problem: (1) focused on addressing strategies for information sharing among patients, we adopted a transfer learning method, which uses MOO to share data from a source patient to a target one. Furthermore, we resort to ensembles to aggregate these transfer-learned classifiers coming from multiple feature extractions; (2) focused on addressing strategies for aggregating multiple feature extractions, we proposed models that explicitly deals with all feature extractions at once, weighting the influence of each extraction in the loss function and the regularization.

6.2.1 Datasets description

The data analyzed in this experiment was extracted from electroencephalographic (EEG) recordings in the Physionet Database (GOLDBERGER *et al.*, 2000; SHOEB, 2009). Among its available patients and channels, here we work with the following subset of channels $\mathcal{C}$ present in all patients $\mathcal{P} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 20, 21, 22, 23\}$, for uniformity

[6] This section is an amended version of Ensembles of Multiobjective-Based Classifiers for Detection of Epileptic Seizures (BESERRA *et al.*, 2018), a research made in partnership with Fernando dos Santos Beserra

across patient's settings. Whenever convenient, the EEG recording at a given channel $c \in C$ and time t will be denoted by $data(t)[c]$. The set of extraction methods are defined by $\mathcal{E} \equiv \{gph, fou, wlt\}$ corresponding respectively to synchronization graph-based, Fourier transform-based, wavelet expansion-based extractions.

Wavelet-based features are extracted by a wavelet transform of each data channel. Let $h_i(t) = data(t)[i],\ t = 0, \ldots, (N-1)\delta t$ be the recording of the i-th channel at time $t = T$, where $\delta_t = 1/256$ is the sampling time precision. Then, a wavelet transform of $h_i(t)$ is defined as

$$W_i(n, a) = \sum_{j=0}^{N-1} h_i(j\delta t) f\left(\frac{(j-n)\delta_t}{a}\right) \tag{6.1}$$

where $f(\cdot)$ is the Sombrero Wavelet (LATKA *et al.*, 2003): $f(t) = \sqrt{\frac{\delta_t}{a}}\frac{2}{\sqrt{3}}\pi^{-1/4}(1-t^2)e^{-t^2/2}$. The adopted *a-scales* are A = $\{0.031$, 0.033, 0.037, 0.049, 0.080 , 0.165, 0.392, $1\}$.

The entire generation of a feature vector concatenates two shifts of one second of the W coefficients obtained at t and $t-1$. At each of these time instants, one second of data is used for the transform and the $N = 256$ coefficients are reduced to a pair of means and standard deviations, per channel and per *a-scale*.

Synchronization graph-based features were inspired by the works of Dhulekar *et al.* (2015), Kramer *et al.* (2008). This procedure divides windows of 10s into twenty intervals of the form $I(n) = [t-(1+n/20), t-L(1+n/20)+1](s)$. For each of these subintervals and for every pair of electrodes, the Pearson correlation coefficients between these electrodes' signals were computed, the maximum absolute values of these correlations were retained and used to create a weighted graph, which is then converted to three unweighted undirected graphs, using $\tau = [0.4, 0.5, 0.6]$ as discretization thresholds.

The metrics extracted from each of these graphs were composed of all those outlined in Section 3.1 of Dhulekar *et al.* (2015), adding: number of $\lambda = 1$ eigenvalues of the Laplacian Matrix; average connected component size; adjacency matrix spectral radius; adjacency matrix trace; adjacency matrix energy; clustering coefficient; eccentricity; radius; number of edges; normalized Laplacian Energy; the ratio between the first non-zero and the largest eigenvalue of the normalized Laplacian; the second largest eigenvalue of the Laplacian matrix.

Finally, the features of the kind **Fourier transform-based** are extracted following the procedure defined in Shoeb (2009). They have information of the channels along the interval $[t-4, t+2]$, for a feature characterizing time t.

The feature extraction stage generated a set of pairs $(\boldsymbol{x}_p(t), y_p(t))$, with time discretized

in windows of size $\Delta t = 1s$, and labels generated according to Van Esbroeck *et al.* (2016), where only the first $20s$ of a seizure are marked with $y = 1$ and time instants outside a seizure receive label $y = 0$.

Target patients $p \in \mathcal{P}$ had their data divided into training, validation and out-of-time test sets, by assigning to the test sets all the samples whose time instants were $t \geq t^p_{seiz} = t^*_p - 5\ min$, t^*_p being the lowest time index of a sample occurring after the n^*_p-th last seizure, $n^*_p = max(1, \lfloor n^{seiz}_p/4 \rfloor)$, n^{seiz}_p denoting the number of annotated seizures for patient p. All patients were tested in a set containing a single seizure interval mark, except for patients $6, 12, 14, 20$, whose test sets contained $6, 10, 2, 2$ annotated test seizures, respectively. Training and validation data consisted of a random split (70% / 30%) of the samples whose extraction times were $t < t^p_{seiz}$ and, due to the large amount of data, we retained only a random quarter of the $y = 0$-labeled samples. Source patients $p \in \mathcal{P}$ had their non-seizure training samples with times $t < t^p_{seiz}$ again subsampled by a quarter, while their validation data consisted of all samples with associated $t \geq t^p_{seiz}$.

6.2.2 Transfer learning applied to the detection of epileptic seizures

6.2.2.1 Proposed methods

6.2.2.1.1 Single-task predictor generated by multi-objective optimization

Considering a unique target-task $t \in \mathcal{P}$ and a unique feature extraction $e \in \mathcal{E} \equiv \{gph, fou, wlt\}$, with $\mathcal{P}$ and $\mathcal{E}$ defined in Section 6.2.1. the models are trained using NISE on a balanced **regularized multinomial logistic regression** formulation (Equation (4.6), Section 4.1.3). This procedure generates a set of 25 models, where the predictor is selected using the harmonic mean between sensitivity and specificity in a validation set. Here, three predictors are created: STNISE_gph, STNISE_fou and STNISE_wlt, corresponding to single-task learning strategies using, respectively, graph, Fourier and wavelet feature extractions.

Remembering Equation (4.6):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}), \qquad \text{(4.6 revisited)}$$

6.2.2.1.2 Ensemble of single-task learned models

For every feature extraction $e \in \mathcal{E}$ on target-task $t \in \mathcal{P}$, the models are trained using NISE on a balanced **regularized multinomial logistic regression** formulation (Equation (4.6), Section 4.1.3). Then, the 25 models for every extraction $e \in \mathcal{E}$ are gathered to compose an ensemble, selected (wta) or filtered using the 10 best models (elt) and aggregated by summing the distributions (better explained in Section 4.3). This pipeline is depicted in Figure 23, where each NISE box represents the training of an extraction $e \in \mathcal{E}$ to a target-task $t \in \mathcal{P}$. Here, two predictors are created: STNISE_wta and STNISE_elt.

Remembering Equation (4.6):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[y_i^k \ln \left(\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 r(\boldsymbol{\theta}), \qquad \text{(4.6 revisited)}$$

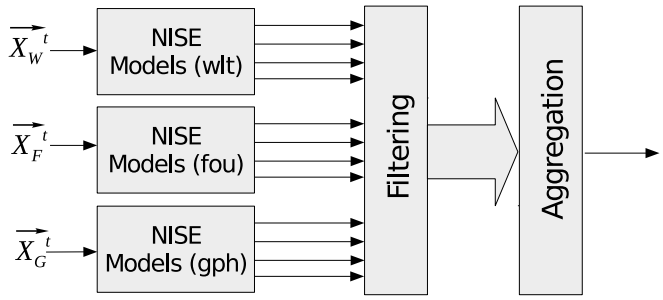


Figure 23 – Representation of the ensemble operation involving single-task learned models.

6.2.2.1.3 Ensemble of transfer-learned models

For every feature extraction $e \in \mathcal{E}$ and for every transfer for a unique source-task $s \in \mathcal{P}$ to a unique target-task $t \in \mathcal{P}$, the models are trained using NISE with a **single-parameter label-conflicting regularized logistic regression** formulation (Equation (4.12), Section 4.1.6). This pipeline is represented in Figure 24 where each NISE box represents the training of an extraction $e \in \mathcal{E}$, all using a transfer from a source-task $s \in \mathcal{P}$ to a unique target-task $t \in \mathcal{P}$. Then, the 25 models for every source-task $s \in \mathcal{P}$ and every extraction $e \in \mathcal{E}$ are gathered to compose an ensemble, selected (wta) or filtered using the 10 best models (elt) and aggregated by summing the distributions (better explained in Section 4.3). Here, two predictors are created: TLNISE_wta and TLNISE_elt.

Remembering Equation (4.12):

$$\min_{\boldsymbol{\theta}} \quad w_1 \left(l(\boldsymbol{\theta}, \mathbf{x}^s, \mathbf{y}^s) + \lambda r(\boldsymbol{\theta}) \right) + w_2 \left(l(\boldsymbol{\theta}, \mathbf{x}^t, \mathbf{y}^t) + \lambda r(\boldsymbol{\theta}) \right) \qquad \text{(4.12 revisited)}$$

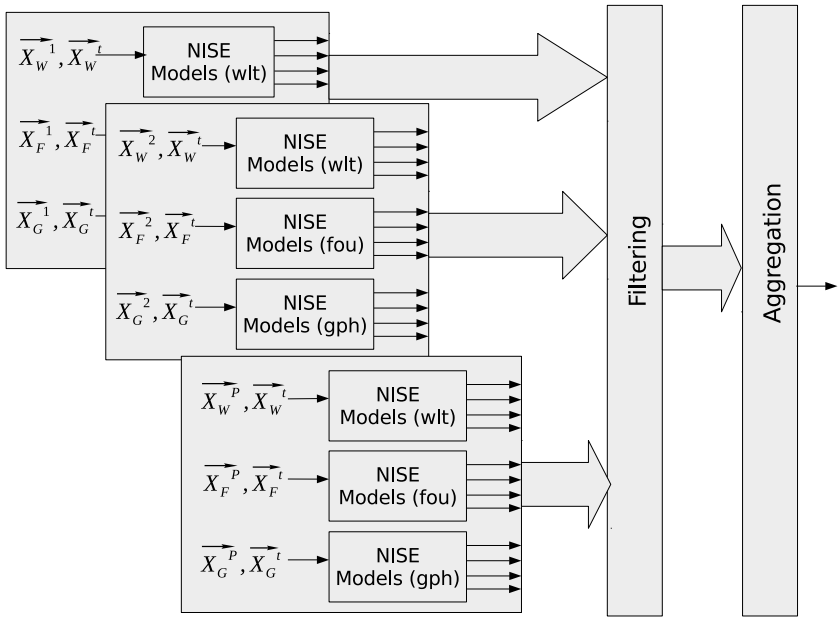


Figure 24 – Representation of the ensemble operation involving transfer-learned models.

6.2.2.2 Experimental setup, baselines and evaluation metrics

Jointly with STNISE_{gph,fou,wlt} single-task baseline methodologies, three other comparison benchmarks were developed: SVM_gph, SVM_fou and SVM_wlt, characterized by the results of the best chosen configurations of Support Vector Machines, in a validation set, with a regularization parameter varying in $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{+2}\}$ and kernels belonging to sigmoid or radial basis function, for each of the feature extraction methods treated individually.

The metrics used to evaluate the performance were sensitivity, denoted by SEN, the ratio of true positives to the total of positives; specificity, denoted by SPE, the ratio of true negatives to the total of negatives; latency, denoted by LAT, the required time to detect a given seizure since its onset mark; the area under the receiver operating characteristic curve, denoted by AUC; and the number of entirely undetected seizures, i.e., cases where all the 20 positive instances of a seizure were missed, across all patients, denoted by UND.

Table 7 – Friedman rank and average values for SEN, SPE, LAT, AUC and UND metrics.

	SEN		SPE		LAT		AUC		UND
	rnk	mean ± std	rnk	mean ± std	rnk	mean ± std	rnk	mean ± std	sum
SVM_fou	5.7	0.77 ± 0.24	4.0	0.87 ± 0.31	6.3	3.96 ± 4.65	3.9	0.89 ± 0.23	0
SVM_gph	5.7	0.81 ± 0.21	7.4	0.83 ± 0.26	5.2	2.23 ± 3.96	6.5	0.88 ± 0.15	0
SVM_wlt	6.5	0.74 ± 0.25	8.1	0.80 ± 0.23	6.7	3.23 ± 3.64	8.1	0.81 ± 0.20	2
stNISE_fou	3.9	0.86 ± 0.17	4.5	0.86 ± 0.21	3.9	1.89 ± 3.23	3.4	0.95 ± 0.08	0
stNISE_gph	7.6	0.61 ± 0.24	7.2	0.89 ± 0.14	7.9	5.73 ± 4.63	7.3	0.88 ± 0.09	1
stNISE_wlt	6.5	0.72 ± 0.27	5.6	0.91 ± 0.17	5.7	2.47 ± 3.31	5.7	0.91 ± 0.09	1
stNISE_wta	4.5	0.80 ± 0.20	4.5	0.90 ± 0.18	4.5	2.80 ± 3.52	6.8	0.85 ± 0.11	0
stNISE_elt	4.9	0.82 ± 0.17	4.7	0.96 ± 0.06	5.5	2.84 ± 3.72	3.2	0.96 ± 0.07	0
tlNISE_wta	4.9	0.80 ± 0.20	4.5	0.92 ± 0.15	4.6	2.45 ± 3.39	7.5	0.86 ± 0.10	0
tlNISE_elt	4.9	0.82 ± 0.18	4.5	0.92 ± 0.16	4.8	2.55 ± 3.58	2.7	0.96 ± 0.07	0

6.2.2.3 Results

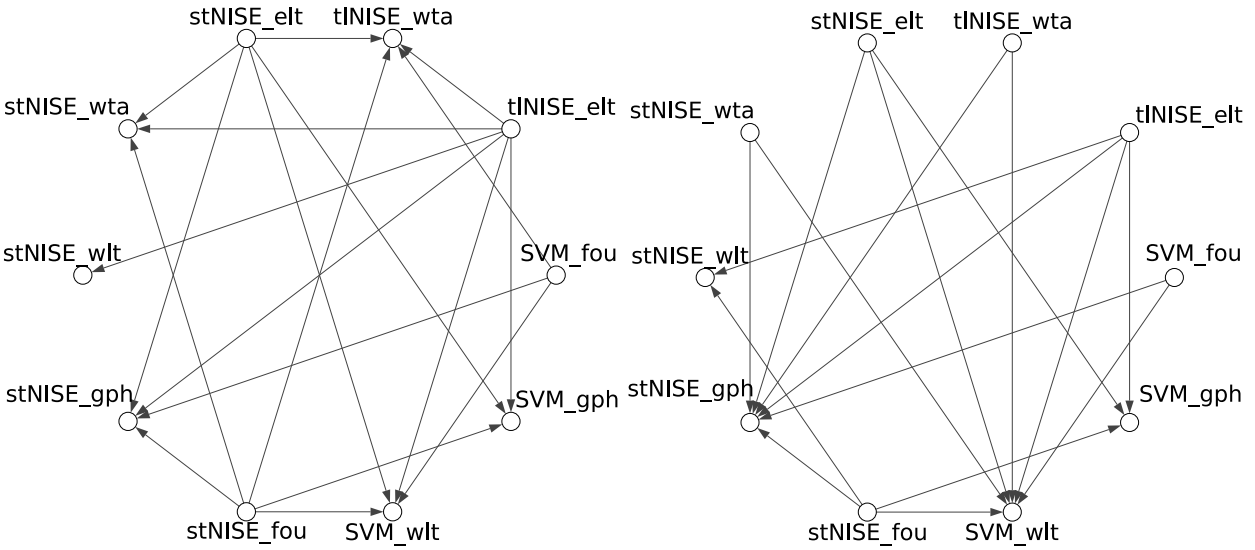
The first column of each metric in Table 7 corresponds to the average Friedman rank, used in the Friedman test, and the second corresponds to the mean ± standard deviation, except for the undetected seizures metric (UND) which indicates the sum of the undetected seizures across all patients. For all metrics, lower ranks indicate better methods, higher means of SEN, SPE and AUC values indicate better methods, and for latency and UND, lower values indicate better methods. The obtained results for all the detailed metrics are exhibited in Table 7.

To make more assertive comparisons, we use a Friedman test, first to detect whether all the methods are similar, or reject the null hypothesis. If the null hypothesis is rejected, as a post-hoc stage, we used the Finner test. The latter test compares pairs of methods and, when the similarity hypothesis is rejected, the method with lower rank is considered better than the method with higher rank. Both tests consider a significance level of $t = 0.01$.

The Friedman test rejected the null hypotheses for SPE, LAT, AUC and a global test, that compared all metrics, for all the patients. The Finner *post-hoc* test for the metrics AUC and global are in Figures 25-(a) and 25-(b), in a directed graph format, where the arrow goes from the better method to the (pairwise) worse method. This *post-hoc* test also indicates that SVM_fou performed better than SVM_wlt for SPE, and for LAT, stNISE_fou performed better than stNISE_gph.

6.2.2.4 Discussion

From the statistical point of view, and considering the evaluation of AUC and global metrics, presented at Figures 25a and 25b, it is possible to infer that the top three algorithms,



(a) Finner testing AUC (b) Finner testing global

Figure 25 – Graphs denoting the results of a Finner *post-hoc* test, indicating the methods hierarchy obtained for AUC, and the global comparison of metrics.

those with the highest out-degrees, are tlNISE_elt, stNISE_elt and stNISE_fou. These algorithms keep a good performance on SPE and LAT metrics, not being statistically worse than any method. Additionally, analyzing the performances *per se* at Table 7, these classifiers have different facets. stNISE_fou is specialized in sensitivity and latency but keeping a fair specificity, stNISE_elt and tlNISE_elt have a higher specificity keeping a reasonable sensitivity and latency. Moreover, none of these algorithms had completely missed seizures, since their UND value is 0. Interestingly, the model selection approaches (stNISE_wta and tlNISE_wta) did not achieve the performance level obtained by ensemble approaches (*_elt), even though selecting from the same set of classifiers.

Analysing classifiers with specific feature extractions, when it is evaluated the in-degree and out-degree on graphs of Figure 25 for both SVM and stNISE, it is possible to see that the Fourier feature extraction produces the best classifiers, and the graph feature extraction produces the worst classifiers.

6.2.3 Multi-view learning applied to detection of epileptic seizures

This experiment aims at investigating the impact of multi-objective training using Group LASSO based formulations, where the result of each feature extraction procedure is considered as a group (or view) to be properly weighted by the learning model.

6.2.3.1 Proposed methods

Supported by the formulations presented in Section 4.1.7, we want to make an exhaustive comparison across all formulations, allowing each view to have a specific loss and/or a specific penalty. The importance for the learning is automatically determined by a multi-objetive formulation (NISE for bi-objective and MONISE for many-objective formulations).

Given that, we proposed four classes of methods which are determined by adjusting the parameters of Equation (4.16): (1) making $w_v^l = w^l \ \forall v \in V$ and $w_v^r = w^r \ \forall v \in V$, we find models which consider only the loss as conflicting with all regularization terms, called here as **SPSL_*; (2) making $w_v^l = w^l \ \forall v \in V$, we find models which considers a single loss conflicting with the regularization terms for each group; called here as **MPSL_***; and (3) making $w_v^r = w^r \ \forall v \in V$, we find models which consider a single regularization for all groups conflicting with the loss for each group; called here as **SPML_***; and finally (4) all weights are left free and then, we find models which consider the loss function for each group as conflicting with the regularization terms for each group, called here as **MPML_***.

Remembering Equation (4.16):

$$\min_{\boldsymbol{\theta}} \quad \sum_{v=1}^{V} w_v^l l(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}^{(v)}) + \sum_{v=1}^{V} w_v^r \sqrt{\sum_{i \in \mathcal{V}^v} \boldsymbol{\theta}_i^2}, \tag{4.16 revisited}$$

Each formulation is trained by NISE or MONISE to find $50 \times |\mathcal{E}|$ models, where $|\mathcal{E}|$ is the number of feature extraction procedures. After that, they are filtered and aggregated by three methods: (1) ***_wta**: multi-objective procedure followed by a model selection; (2) ***_elt**: multi-objective procedure followed by an elite selection and a distribution summation (ROKACH, 2010); (3) ***_stk**: multi-objective procedure followed by a stacking training.

These combinations generate twelve possible methods.

6.2.3.2 Experimental setup, baselines and evaluation metrics

These methods are compared with the single-task baseline methodologies, already presented in the previous experiment in Section 6.2.2.1.1, called here SV_{gph,fou,wlt}, trained to find 50 models and also aggregated with the same ensemble methodologies {wta, elt, skt}.

The metrics used to evaluate the performance were sensitivity, denoted by SEN, the ratio of true positives to the total of positives; specificity, denoted by SPE, the ratio of true negatives to the total of negatives; latency, denoted by LAT, the required time to detect a given seizure since its onset mark; the area under the receiver operating characteristic curve,

Table 8 – Friedman rank and average values for SEN, SPE, LAT, AUC and UND metrics.

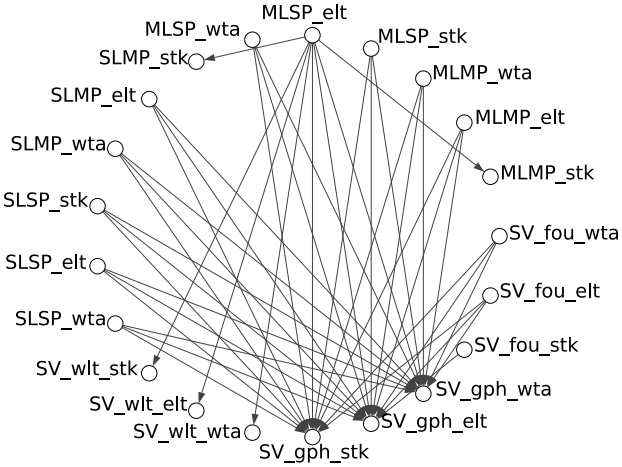
		SEN		SPE		LAT		AUC		UND
gen	agr	rnk	mean ± std	rnk	mean ± std	rnk	mean ± std	rnk	mean ± std	sum
SV_fou	wta	10.9	0.833 ± 0.166	8.4	0.953 ± 0.039	11.1	2.634 ± 3.509	11.7	0.953 ± 0.046	0
	elt	11.3	0.826 ± 0.182	8.2	0.951 ± 0.045	12.6	2.810 ± 3.750	9.7	0.955 ± 0.061	0
	stk	8.6	0.853 ± 0.174	11.4	0.935 ± 0.051	9.0	2.029 ± 3.427	9.3	0.955 ± 0.053	0
SV_gph	wta	12.9	0.783 ± 0.218	17.6	0.857 ± 0.080	12.1	3.130 ± 4.362	18.1	0.891 ± 0.086	0
	elt	13.5	0.775 ± 0.263	18.2	0.863 ± 0.073	13.2	3.739 ± 5.304	18.8	0.883 ± 0.116	0
	stk	14.8	0.735 ± 0.252	18.0	0.886 ± 0.070	12.8	3.995 ± 4.624	17.8	0.882 ± 0.138	0
SV_wlt	wta	13.9	0.745 ± 0.216	12.3	0.924 ± 0.064	11.0	2.432 ± 3.288	15.5	0.901 ± 0.101	0
	elt	13.3	0.771 ± 0.232	12.0	0.934 ± 0.047	10.4	2.302 ± 3.440	14.3	0.922 ± 0.084	0
	stk	12.4	0.789 ± 0.220	13.9	0.916 ± 0.038	10.9	2.394 ± 4.130	14.4	0.918 ± 0.095	0
SLSP	wta	11.2	0.829 ± 0.213	9.2	0.953 ± 0.033	12.4	3.088 ± 4.472	9.8	0.947 ± 0.086	0
	elt	11.5	0.822 ± 0.214	8.6	0.954 ± 0.035	12.6	3.117 ± 4.487	10.2	0.949 ± 0.084	0
	stk	9.9	0.836 ± 0.204	10.4	0.943 ± 0.034	9.4	2.117 ± 4.117	9.4	0.945 ± 0.090	0
SLMP	wta	9.9	0.831 ± 0.205	7.8	0.959 ± 0.032	10.5	2.697 ± 4.169	9.9	0.949 ± 0.084	0
	elt	9.9	0.839 ± 0.202	7.5	0.960 ± 0.033	10.5	2.373 ± 4.145	7.7	0.956 ± 0.086	0
	stk	7.8	0.862 ± 0.176	12.4	0.921 ± 0.079	9.9	2.180 ± 3.247	9.9	0.947 ± 0.089	0
MLSP	wta	10.5	0.825 ± 0.181	7.4	0.958 ± 0.034	10.0	1.953 ± 3.471	9.0	0.953 ± 0.073	0
	elt	11.0	0.815 ± 0.194	5.2	0.964 ± 0.032	10.8	2.453 ± 3.896	5.1	0.964 ± 0.074	0
	stk	8.4	0.856 ± 0.194	10.5	0.933 ± 0.055	9.3	2.218 ± 3.905	7.0	0.952 ± 0.082	0
MLMP	wta	11.5	0.816 ± 0.196	7.7	0.957 ± 0.035	11.2	2.491 ± 4.065	9.3	0.954 ± 0.076	0
	elt	10.4	0.825 ± 0.195	9.8	0.952 ± 0.043	11.9	3.079 ± 4.219	7.2	0.960 ± 0.076	0
	stk	6.3	0.881 ± 0.192	13.2	0.926 ± 0.051	8.2	1.689 ± 3.552	5.8	0.958 ± 0.078	0

denoted by AUC; and the number of entirely undetected seizures, i.e., cases where all the 20 positive instances of a seizure were missed, across all patients, denoted by UND.

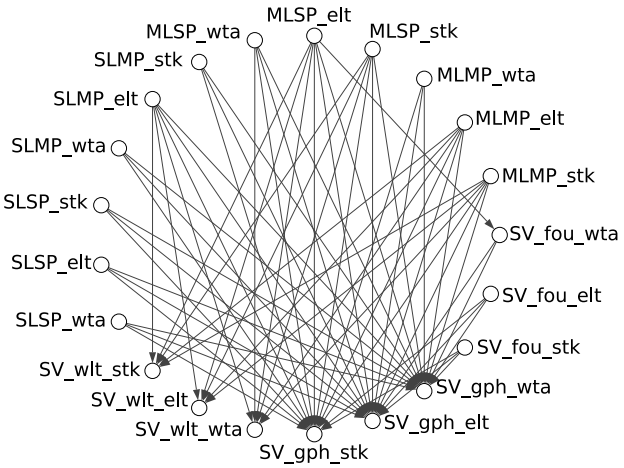
6.2.3.3 Results

The obtained results for all the detailed metrics are exhibited in Table 8. The first column of each metric in Table 8 corresponds to the average Friedman rank, used in the Friedman test, and the second corresponds to the mean ± standard deviation, except for the undetected seizures metric (UND) which indicates the sum of the undetected seizures across all patients. For all metrics, lower ranks indicate better methods, higher means of SEN, SPE and AUC values indicate better methods, and for latency and UND, lower values indicate better methods.

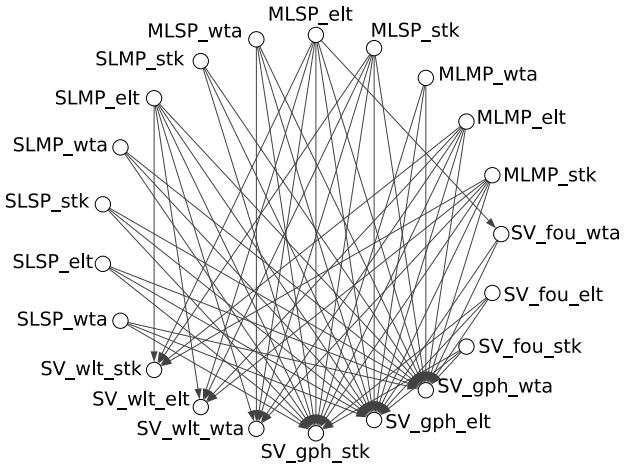
To make more assertive comparisons, we use a Friedman test, first to detect whether all the methods are similar, or reject the null hypothesis. If the null hypothesis is rejected, as a post-hoc stage, we used the Finner test. The latter test compares pairs of methods and, when the similarity hypothesis is rejected, the method with lower rank is considered better



(a) Finner testing SPE



(b) Finner testing AUC



(c) Finner testing global

Figure 26 – Graphs denoting the results of a Finner *post-hoc* test, indicating the pairwise comparison of methods considering SPE, AUC and global metrics.

than the method with higher rank. Both tests consider a significance level of $t = 0.01$.

The Friedman test rejected the null hypotheses for SEN, SPE, AUC and a global test, that compared all metrics, for all the patients. The Finner *post-hoc* test for the metrics SPE, AUC and global are in Figures 26-(a), 26-(b) and 26-(c), in a directed graph format, where the arrow goes from the better method to the (pairwise) worse method. Despite the fact that Friedman test rejected the null hypotheses for SEN, no significant difference was found between the methods.

6.2.3.4 Discussion

From the statistical point of view, and considering the evaluation of SPE, AUC and global metrics, presented at Figures 26-(a), 26-(b) and 26-(c), it is possible to infer that the algorithms SLMP_elt, MLSP_elt and MLMP_elt as the best algorithms, and the algorithms with wavelet and graph based feature extraction as the worst algorithms. Additionally, analyzing the performances *per se* at Table 8, these classifiers have different facets. MLSP_stk, MLMP_stk, SLMP_stk and SV_fou_stk are specialized in SEN and LAT and MLSP_elt, MLMP_wta, MLMP_elt and SV_fou_elt are specialized in SPE. It is worth mention that SLMP_elt is a quite competent detector in all SEN, SPE, LAT and AUC metrics being a good tradeoff solution.

Additionally, it is possible to see that the performance is bounded by the Fourier-based extraction procedure. However, the methods that used more feature extractions were capable of keeping and even improving the performance compared to classifiers that use only the Fourier-based extraction procedure. It opens the possibility of improving the performance of the classification by properly combining multiple views.

6.3 Multi-label classification [7]

In multi-label classification, a sample can be assigned to an arbitrary number of labels (among L possible labels). The main challenge in this problem is to correctly find the learning relations among the labels to improve the performance.

In this experimental design, we choose the logistic or multinomial regression as base classifiers for meta-learning design that tries to induce joint learning among the labels. Fixing the base classifiers, the objective is to isolate the influence of the meta-learners and their ability to improve the classification performance.

6.3.1 Datasets description

To evaluate the potential of the proposed multi-objective ensemble-based methodology we consider six datasets[8]. Table 9 provides the main aspects of these datasets. Aiming at obtaining better statistical results, we used a 10-fold split to create 10 independent test sets with 10% of the samples, and the remaining samples are randomly divided into 75% for

[7] This section is an amended version of Many-Objective Ensemble-Based Multilabel Classification (RAIMUNDO; VON ZUBEN, 2018b)

[8] Available at <mulan.sourceforge.net/datasets-mlc.html>

Table 9 – Description of the benchmark datasets.

Name	Instances	Atributes	Labels	Cardinality	Density of 1s
emotions	593	72	6	1.869	0.311
scene	2407	294	6	1.074	0.179
flags	194	19	7	3.392	0.485
yeast	2417	103	14	4.237	0.303
birds	645	260	19	1.014	0.053
genbase	662	1186	27	1.252	0.046

training and 25% for validation for the baseline algorithms and 50% for T_1 set and 50% for T_2 set for the proposed method. T_1 set was used to create the ensemble components, and T_2 set to train the stacked classifiers. T_1 set was used again to select the model in the stacking training procedure.

6.3.2 Proposed method

Stacking is an ensemble methodology that uses the outcome of the ensemble components (learning machines trained by an ensemble generation methodology, represented in Step 1 of Figure 27a) to train another classifier (Step 2 of Figure 27a) which will be responsible for making the prediction. In our proposal, we chose the model formulation to the **single-parameter label-conflicting regularized multinomial logistic regression** (Equation (4.8), Section 4.1.4), and the many-objective optimization method is responsible for generating R classifiers to compose the ensemble.

Remembering Equation (4.8):

$$\min_{\boldsymbol{\theta}} \quad \sum_{l=1}^{L} v_l l(\mathbf{x}, \mathbf{y}^l, \boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||^2 \equiv \sum_{l=1}^{L} w_l l(\mathbf{x}, \mathbf{y}^l, \boldsymbol{\theta}) + w_{L+1} ||\boldsymbol{\theta}||^2. \tag{4.8 revisited}$$

and a stack classifier is responsible for predicting each label l using logistic regression as the classification model:

$$\min_{\widehat{\boldsymbol{\theta}}^l} \; -\sum_{i=1}^{N} \left[\frac{1}{k_1^l} \mathbf{y}_i^l \ln \left(f(\mathbf{z}_i, \widehat{\boldsymbol{\theta}}^l) \right) + \frac{1}{k_0^l} (1 - \mathbf{y}_i^l) \ln \left(1 - f(\mathbf{z}_i, \widehat{\boldsymbol{\theta}}^l) \right) \right] \tag{6.2}$$

where $\mathbf{z}_i^j$ is the degree of membership predicted by the j-th ensemble component with relation to the i-th sample.

The **proposed** method consists in: **Step 1**, ensemble components ($\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_l\}$) are generated by finding a set of efficient solutions, to the formulation of Equation (4.8) using the methodology described in Section 4.2 (Figure 27b). This generator of many-objective

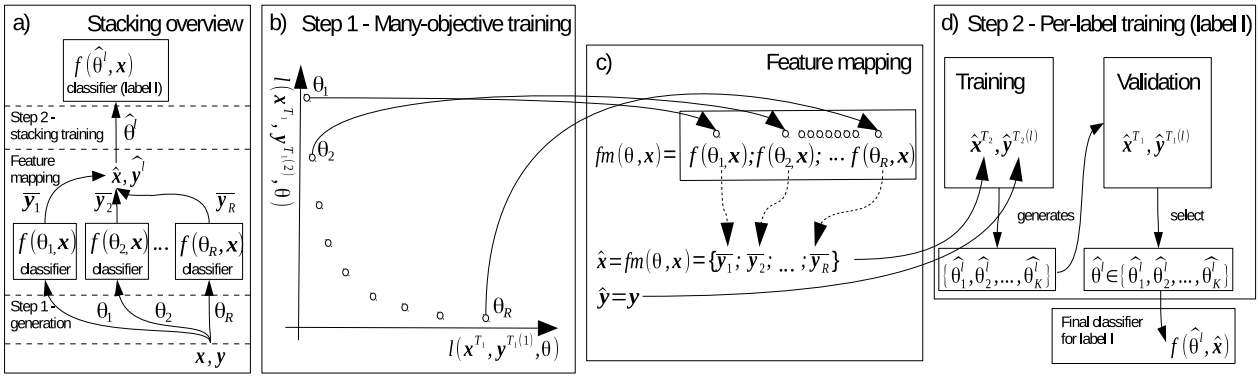


Figure 27 – Many-objective training followed by a stacking aggregation representation.

ensemble components can be seen as a **feature vector mapping** ($fm(\boldsymbol{\theta}, x)$), so that each mapped feature is a classifier (Figure 27c) associated with a distinct weight vector. From Equation (4.8) we can realize that each classifier will give a distinct weight to the loss at each label and also to the regularization term. In **Step 2** (represented in Figure 27d), the output of all efficient solutions are aggregated using a stacking approach (ROKACH, 2010), which can be seen as a cross-validation procedure in the mapped feature space $\widehat{x}$ using the model of Equation (6.2). The training procedure in the second step is done for each label l employing the same feature vector $\widehat{x}$, but adopting the output $\widehat{y}^l$ of the worked label. The set of classifiers were generated by a weighted average of the label losses. For this reason, not all classifiers will have a good performance for a specific label, thus requiring a more flexible aggregation, such as stacking, to create a final classifier.

6.3.3 Experimental setup, baselines and evaluation metrics

To create a **solid baseline** we compared our method with 5 other approaches: Binary Relevance (BR), Classification Chains (CC) (READ *et al.*, 2011), RakelD (TSOUMAKAS *et al.*, 2011), Label Powerset (LP)[9]. All of those methods were implemented using Logistic Regression[10] as the base classifier and had their parameters selected using hyperopt[11] with 50 evaluations to tune regularization strength and 50 more evaluations if the method involves another parameter (RakelD). Since the proposed and baseline algorithms use logistic regression as the base classifier, the attributes are considered as a vector of real numerical values.

[9]Implementations available at <http://scikit.ml/>
[10]Available at <http://scikit-learn.org>
[11]Available at <https://github.com/hyperopt/hyperopt>

In our proposal, we generate $10 * (L + 1)$ ensemble components, and the parameter selection on the stacking phase was implemented by cross-validation with 50 evaluations. We developed two versions of our proposal. These versions were created by balancing or not the importance of a label according to the stacking by changing the constants k_1^l and k_0^l on Equation (6.2). In the imbalanced approach (described as MOn) k_1^l and k_0^l were set to 1, and in the balanced approach (described as MOb) k_1^l was set to the number of 1s for this labels and k_0^l for that specific number of 0s.

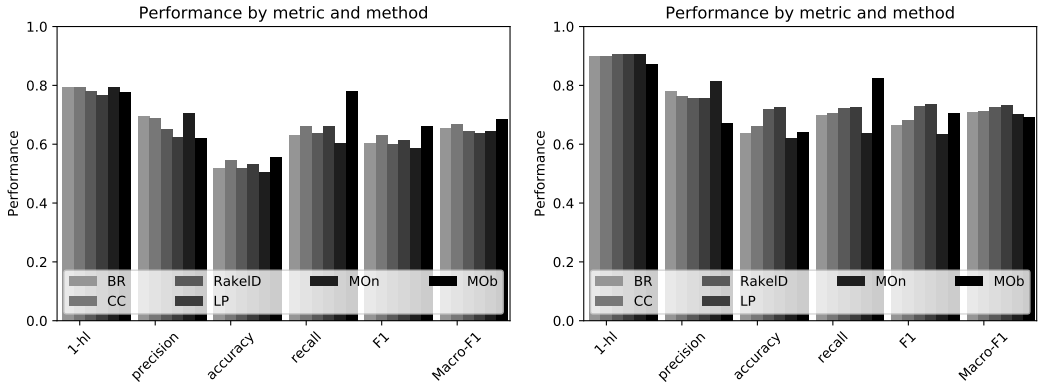
The used **evaluation** metrics were: 1-Hamming Loss (1-hl), precision, accuracy, recall, F1 and Macro-F1 (MADJAROV *et al.*, 2012; GONÇALVES *et al.*, 2015), all of those metrics associated with a quality measure in the interval $[0, 1]$ so that higher values indicate a better method.

6.3.4 Results

To promote an extensive comparison we presented the results from two perspectives. Figure 28 presents the average performance, calculated over the 10-folds, for all evaluated metrics for each dataset. And to make a more incisive evaluation, we used a Friedman paired test with $p = 0.01$ comparing all folders for all datasets, followed by a Finner posthoc test with the same p, if Friedman test were rejected. Table 10 contains the evaluated method in the rows, and, for each performance metric, the Friedman ranking in the first column, how many methods are worse than the evaluated metric in the second column, and how many methods are better in the third column. This ordering relation (better and worse) is accounted only if there is a statistical significance according to the Finner posthoc test. Looking to RakelD for the precision score, it is statistically significantly better than the worst ranked method: MOb (4.92), and statistically significantly worse than the three better-ranked methods: MOn (1.99), BR (2.74) and CC (2.94).

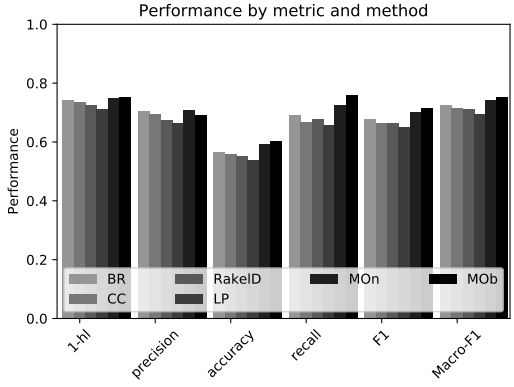
Table 10 – Average ranking and statistical comparisons for each metric.

	1-hl			precision			accuracy			recall			F1			Macro-F1		
method	rank	>	<	rank	>	<	rank	>	<	rank	>	<	rank	>	<	rank	>	<
BR	3.29	1	1	2.74	3	0	3.79	0	1	3.73	1	1	3.68	0	0	3.22	0	0
CC	3.29	1	1	2.94	3	1	2.64	3	0	3.16	1	1	2.9	1	0	2.87	0	0
RakelD	3.64	1	1	3.95	1	3	3.35	0	0	3.59	1	1	3.3	1	0	3.45	0	0
LP	3.89	0	1	4.43	0	3	3.12	1	0	3.85	1	1	3.54	0	0	4.01	0	0
MOn	2.23	5	0	1.99	4	0	4.31	0	2	4.96	0	5	4.37	0	3	3.75	0	0
MOb	4.63	0	4	4.92	0	4	3.75	0	1	1.69	5	0	3.17	1	0	3.66	0	0

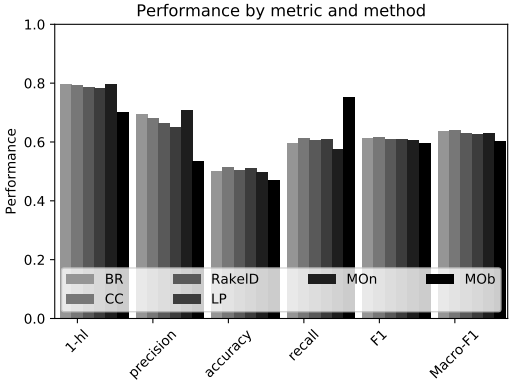


(a) emotions

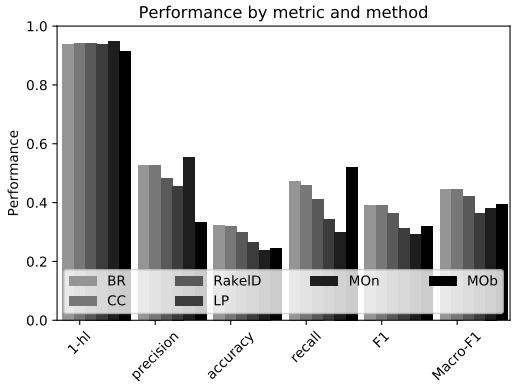
(b) scene



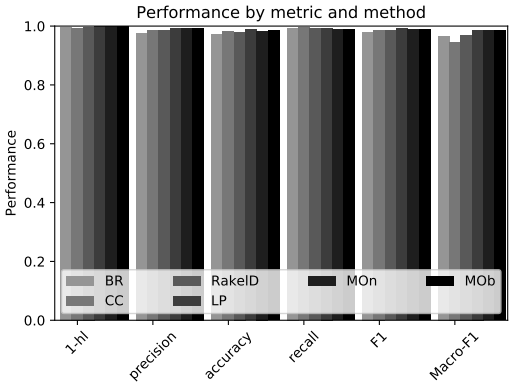
(c) flags



(d) yeast



(e) birds



(f) genbase

Figure 28 – Average performance of the evaluated methods for each metric in each dataset.

6.3.5 Discussion

In Section 6.3, we successfully proposed a many-objective ensemble-based classifier to multi-label classification. **Analyzing** both Figure 28 and Table 10, it is possible to see that MOn is the best-ranked classifier on 1-hl and precision but falling away on recall, accuracy,

and F1. MOb is one of the best-ranked classifiers on recall and F1, but has difficulties on 1-hl and precision. These findings indicate that these two classifiers are biased for some metrics, exhibiting complementary performance. This behavior is due to the low density of the datasets, and to the fact that the non-balanced stacked model focuses the prediction on the 0s, thus producing high precision, as long as the balanced approach is predicting 1s more frequently, explaining the high recall score.

This scenario where an approach has a good performance on specific metrics at the expense of performance loss for other metrics can be useful in some applications. Given the absence of a dominant method for all metrics, our proposals can be seen as valuable choices in metric-driven scenarios. Also, since the complementary behavior was generated changing parameters, further exploration using ensembles of many-objective trained classifiers can promote good classifiers with different performance profiles.

6.4 Multi-task learning [12]

In multi-task learning, the primary challenge is to promote joint learning among multiple learning tasks. The task relations can be structured in any topology, not always existing among all task altogether. This experimental design considers this and explores a large set of synthetic and real-world datasets. The synthetic datasets try to cover as many scenarios as possible to deeply compare the proposed method against a large set of other methods.

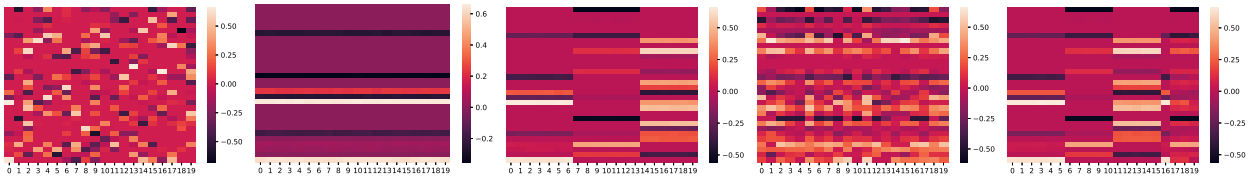
Taking into account that the proposed method generates a set of classifiers with distinct sharing relations, we evaluate these models into distinct perspectives: (1) composing those models using ensemble aggregation techniques and compare with other multi-task learning methods in diverse datasets (Section 6.4.4); (2) evaluating the sensitivity of the methodology w.r.t. the number of generated ensemble components (Section 6.4.5); (3) evaluating the sharing relations of the best ensemble components to analyze if the task relations of the datasets are recovered (Section 6.4.6).

With this diversity of problems, we aimed at investigating many aspects, qualities, and flaws of the proposed methodology.

[12] This section is an amended version of Investigating multi-objective methods in multi-task classification (©2018 IEEE)(RAIMUNDO; VON ZUBEN, 2018a)

6.4.1 Datasets description

6.4.1.1 Synthetic datasets



(a) independent (b) single cluster (c) three clusters (d) subspace (e) structures

Figure 29 – A heatmap representation of the task parameters with distinct sharing structures. Parameters are located at the ordinate axis, and tasks at the abscissa axis.

Aiming at ensuring an experimental analysis with diversity of sharing structures, we decided to construct synthetic datasets following a procedure already considered in the literature (ZHONG *et al.*, 2012) but expanding to more datasets. We designed synthetic datasets with $T = 20$ classification tasks and $d = 30$ attributes. Any sample i for any task t is generated by a multivariate normal distribution $\mathbf{x}_i^{(t)} \sim \mathcal{N}(\mathbf{0}_d, I_{d\times d})$. Considering the parameters $\boldsymbol{\theta}^{(t)} \in \mathbb{R}^d$ of the t-th task (further explained), we first calculate the probability $\mathbf{p}_i^{(t)} \sim \frac{1}{1+\exp^{\mathbf{x}_i^{(t)\top}\boldsymbol{\theta}^{(t)}}}$, and we consider the output $\mathbf{y}_i^{(t)} = 1$ if $\mathbf{p}_i^{(t)} > 0.5$ and $\mathbf{y}_i^{(t)} = 0$, otherwise. We also impose a 10% misclassification rate by simply inverting this aforementioned rule.

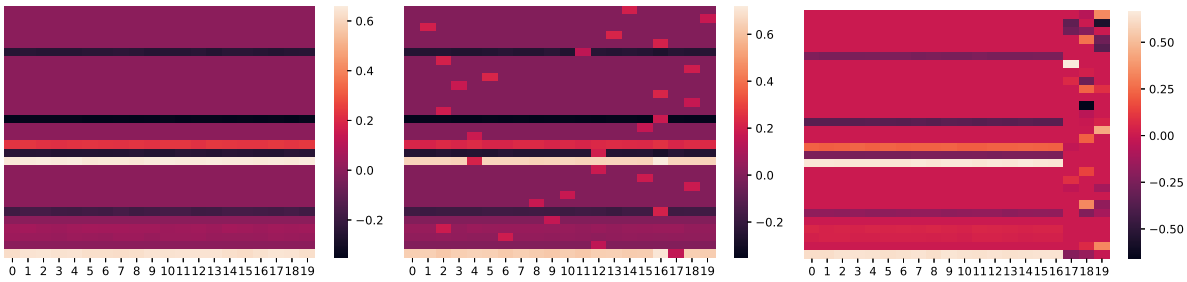
To investigate distinct properties of the multi-task methods, we proposed five groups of multi-task datasets. Each one of these dataset groups is constructed by generating the target parameters $\Theta \equiv \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(T)}\}$ with a procedure that emulates distinct task-sharing properties:

- **Independent** tasks - All tasks are independent: $\boldsymbol{\theta}^{(t)} \sim \mathcal{N}(\mathbf{0}_d, 25I_{d\times d})$ for all t, with 30% of the parameters set to 0 to create a sparse scenario.
- **Single cluster** of tasks - Considering a cluster prototype $\gamma \sim \mathcal{N}(\mathbf{0}_d, 25I_{d\times d})$, with 30% of the parameters set to 0. All tasks share the same cluster with a noise that does not act on the nullified parameters: $\boldsymbol{\theta}^{(t)} \sim \gamma + \mathcal{N}(\mathbf{0}_d, I_{d\times d})$ for all t.
- **Three clusters** of tasks - Same procedure as **single cluster** but creating three distinct clusters.
- Tasks sharing the same **shared subspace** - A subspace $U \in \mathcal{R}^{d\times K}$ was generated considering each column as $\mathbf{u}^{(k)} \sim \mathcal{N}(\mathbf{0}_d, 25I_{d\times d})$ for all $k \in \{1, \ldots, K\}$, with 30% of the

parameters set to 0 to create a sparse parameter set. Given that, each $\boldsymbol{\theta}^{(t)}$ is a random convex combination of the columns of U.

- Groups of tasks with **multiple structures** of sharing - The dataset groups were created imposing some clusters of tasks (three clusters) and making some other groups of tasks to share a subspace.

These distinct groups of tasks are represented in Figure 29 using a heatmap to express the parameter vector of each task.



(a) Gaussian noise (b) Dirty noise (c) Outliers

Figure 30 – A heatmap representation of the different noise profiles applied to the single cluster sharing structure (Figure 29-b). Parameters are located at the ordinate axis and tasks at the abscissa axis.

For each one of those groups of datasets we impute three different noise profiles:

- **None** - No noise is applied
- **Dirty** - Inspired by the structure captured in Jalali *et al.* (2010), and using a procedure proposed by Zhong *et al.* (2012): for each feature j, one task t is chosen to have that feature degenerated doing $\theta_j^{(t)} \sim 10 + \mathcal{N}(0,1)$
- **Outliers** - Inspired by the structure captured in Gong *et al.* (2012), the last three tasks does not share the other structures, being generated as independent tasks $\boldsymbol{\theta}^{(t)} \sim \mathcal{N}(\mathbf{0}_d, 25I_{d\times d})$.

These noise profiles, applied to the single cluster dataset group, are represented in Figure 30 using a heatmap of the task parameter vectors. We generated 15 datasets (three noise profiles applied to five groups of datasets), and each dataset was investigated constraining the sample size of training and validation sets to $n_t \in \{10, 25, 50, 100, 250, 500, 1000\}$, and keeping the same test sample size of 1000 examples.

6.4.1.2 Real datasets

The synthetic datasets work on a wide variety of controlled scenarios, stressing most of the relevant task sharing structures that could be observed in real-world applications. Nonetheless, we also considered real datasets to complement our experimental analysis. The real datasets are: landmine detection[13]; and ECML/PKDD spam detection challenge[14] with 3 and 15 users. We selected 100 features per task using the maximal mutual information[15] to reduce the dimension of spam datasets. The test set for landmine contains from 45 to 290 data points (depending on the task), and the test sets for spam, with 3 and 15 users, contain 500 and 100 samples, respectively. Training and validation datasets have the same size, varying from 10 to the maximum available number of samples, a procedure already adopted for the synthetic datasets.

6.4.2 Proposed method

Given that our multi-task approach is characterized by an L_1 **regularized logistic regression with a single parameter vector for all tasks**, see Section 4.1.5, more specifically Equation (4.11), two steps are necessary: (1) the multi-objective optimization procedure that generates $R = 50{\times}T$ models; (2) a model selection or the ensemble synthesis according to the methodology outlined in Section 4.3. Since the stacking procedure needs another training step, this training was made using the validation dataset and evaluated using the training dataset, in an attempt to avoid overfitting to the training dataset.

Remembering Equation (4.11):

$$\min_{\boldsymbol{\theta}} \quad \sum_{t=1}^{T} w_t l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\theta}) + w_{T+1}||\boldsymbol{\theta}||_1. \qquad \text{(4.11 revisited)}$$

Overall, we **proposed** three methods (see Section 4.3): (1) **mo-wta**: multi-objective procedure followed by a model selection; (2) **mo-elt**: multi-objective procedure followed by an elite selection and a distribution summation (ROKACH, 2010); (3) **mo-stk**: multi-objective procedure followed by a stacking training.

An alternative proposal was also made by changing the first step, where instead of generating $R = 50{\times}T$ models, we generate $R = 25{\times}T$ models using our multi-task multi-objective procedure and generating 25 models per task using a single-task multi-objective procedure (the conflicting objectives are the specific task loss against the regularization strength). This

[13] Available at <ece.duke.edu/~lcarin/LandmineData.zip>
[14] Available at <ecmlpkdd2006.org/challenge.html>
[15] Available at <scikit-learn.org>

procedure generates three methods: (1) **stmo-wta**; (2) **stmo-elt**; (3) **stmo-stk**. The ensemble aggregation is the same as previous methods but using both single-task and multi-task learning models as components.

6.4.3 Experimental setup, baselines and evaluation metrics

Aiming at making a consistent analysis of the potential of our proposal, we designed a **comparison** with three single-task learning methods, and six multi-task learning methods. The single-task learning methods are the following: an L_1 regularized logistic regression model, a bagging of 10 L_1 regularized logistic regression models, and stacking of the models produced by bagging. The multi-task models are regularized logistic models and the methods of regularization are: Joint Feature Selection (OBOZINSKI *et al.*, 2008), Dirty Model (JALALI *et al.*, 2010), Trace Norm Regularization (CHEN *et al.*, 2012), Clustered Multitask Learning (ZHOU *et al.*, 2011), Alternating Structure Optimization (CHEN *et al.*, 2009), and Robust Multitask Feature Learning (GONG *et al.*, 2012)[16].

The hyper-parameters (parameters such as regularization strengths and the number of task clusters) was tuned using hyperopt[17] using 50 evaluations per task. The learning model was synthesized using the training dataset and considering the evaluated criterion. The performance of each learning model was then captured using accuracy on the validation dataset. Therefore, each single-task learning model was trained with 50 evaluations, including each model from the ensemble methods (bagging and stacking), and the stacking training also had 50 evaluations. The multi-task models include $50 \times T$ evaluations to tune the hyper-parameters.

6.4.4 General performance

6.4.4.1 Results

The average and the standard deviation of accuracy are plotted for each size of the training dataset in each experimental scenario. These results for synthetic datasets are shown in Figures 31 and 32; and for real datasets in Figure 33. To enable more detailed plots, we also normalized the average accuracies by subtracting the average accuracy of the single-task method. For visualization purposes, we plotted the following results: the L_1 single-task learning method (identified as **stl**); a single plot for the three single-task methods (identified

[16]The multi-task learning models are implemented as members of the MALSAR toolbox, available at yelab.net/software/MALSAR

[17]Available at github.com/hyperopt

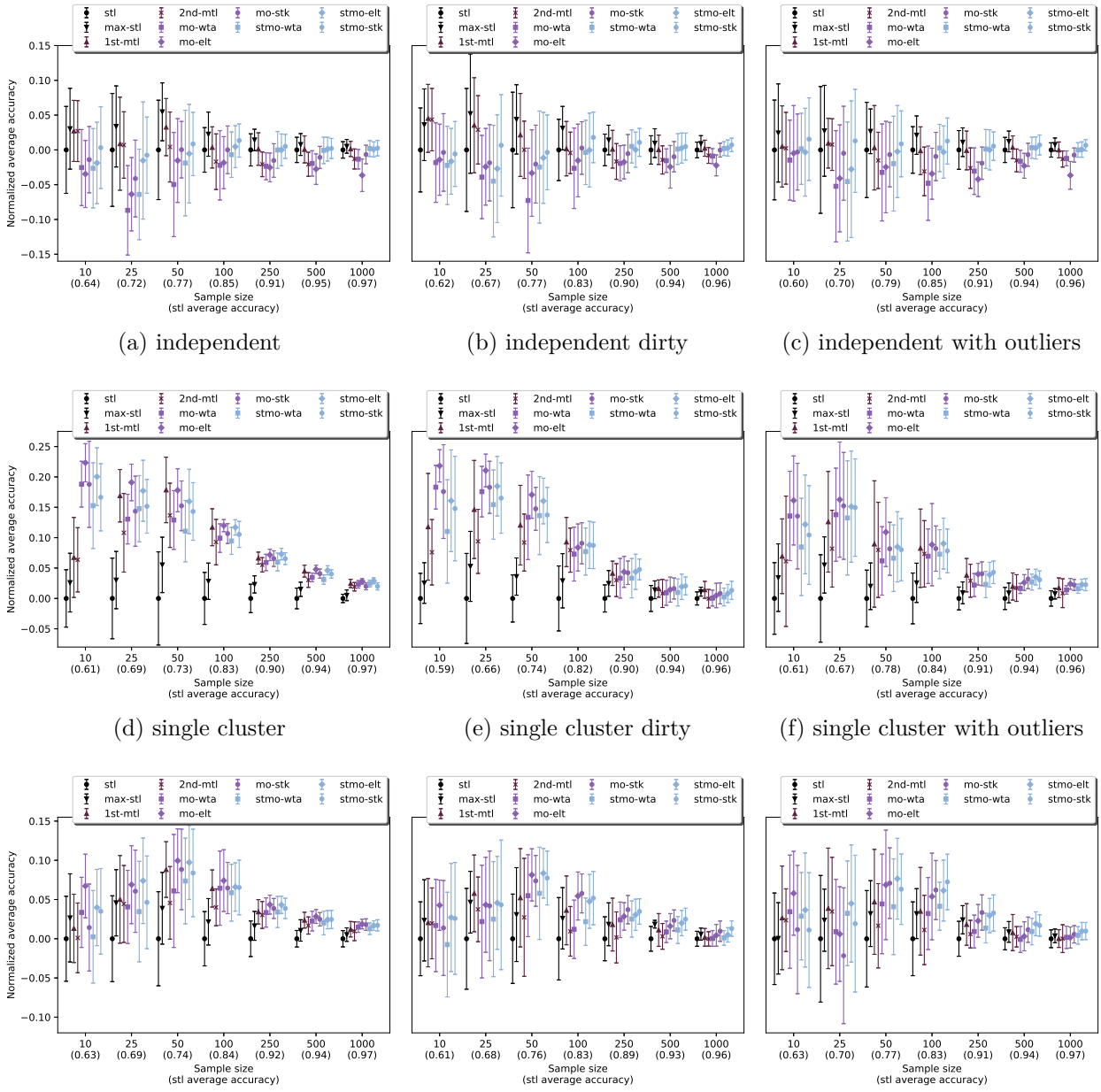
as **max-stl**) showing only the best average (and its correspondent standard deviation); two plots for the six multi-task methods (identified as **mtl-1st** and **mtl-2nd**) showing the best and second best averages (and their correspondent standard deviations); and the averages (and their correspondent standard deviations) of our six proposed methods (identified as **mo-wta**, **mo-elt**, **mo-stk**, **stmo-wta**, **stmo-elt** and **stmo-stk**).

Notice that we are purposely imposing the best possible scenario for the contenders, when taking **mtl-1st** and **mtl-2nd**. Those best and second-best methods are not the same for each sample-size/dataset and are not known a priori, being selected after evaluating all the available configurations for the models. Therefore, overcoming or being competitive with **mtl-1st** and **mtl-2nd** is an expressive result.

6.4.4.2 Discussion

Aiming at making a deeper **analysis**, we are going to, firstly, make an overall analysis of all the base groups with different noise profiles. The proposed single model approach **mo-wta**, apart from the independent dataset, has better performance compared to the single-task single model **stl** on almost all datasets and sample sizes, only being outperformed in larger sample sizes. This statement shows the capability of knowledge sharing over tasks. However, the single-task ensemble approaches and the multi-task learning approaches from the literature usually outperform **mo-wta**. The proposed multi-task ensemble approaches (**mo-elt** and **mo-stk**) are usually better than **mo-wta**, clearly indicating the relevance of ensemble methods. Moreover, except for the independent datasets, our ensembles also have at least a comparable performance with **mtl-2nd** and **max-stl**, frequently figuring as the best method. Our proposal combining single-task and multi-task trained models as ensemble components (**stmo-wta**, **stmo-elt** and **stmo-stk**) had the same performance profile of multi-task only models, but they are more robust, losing performance in rare cases and increasing the performance in harder scenarios for multi-task methods, such as the ones with **independent** or **outlier** tasks.

Our proposals do not achieve good performance on **independent** tasks when compared to single-task methods (and also compared to the best multi-task method), delivering a comparable performance against **mtl-2nd**. The **stmo** techniques were capable of achieving a performance comparable with **mtl-1st**, guiding to good solutions for scenarios with non-shared task datasets. On **single cluster** and **subspace** datasets, our **elt** methods have the best performance on small-size datasets being outperformed and replaced by our **stk** methods on large-size datasets. On **three cluster** and **relationship** tasks, **elt** methods have at least a comparable performance with the best methods from the literature, frequently being the



(a) independent (b) independent dirty (c) independent with outliers

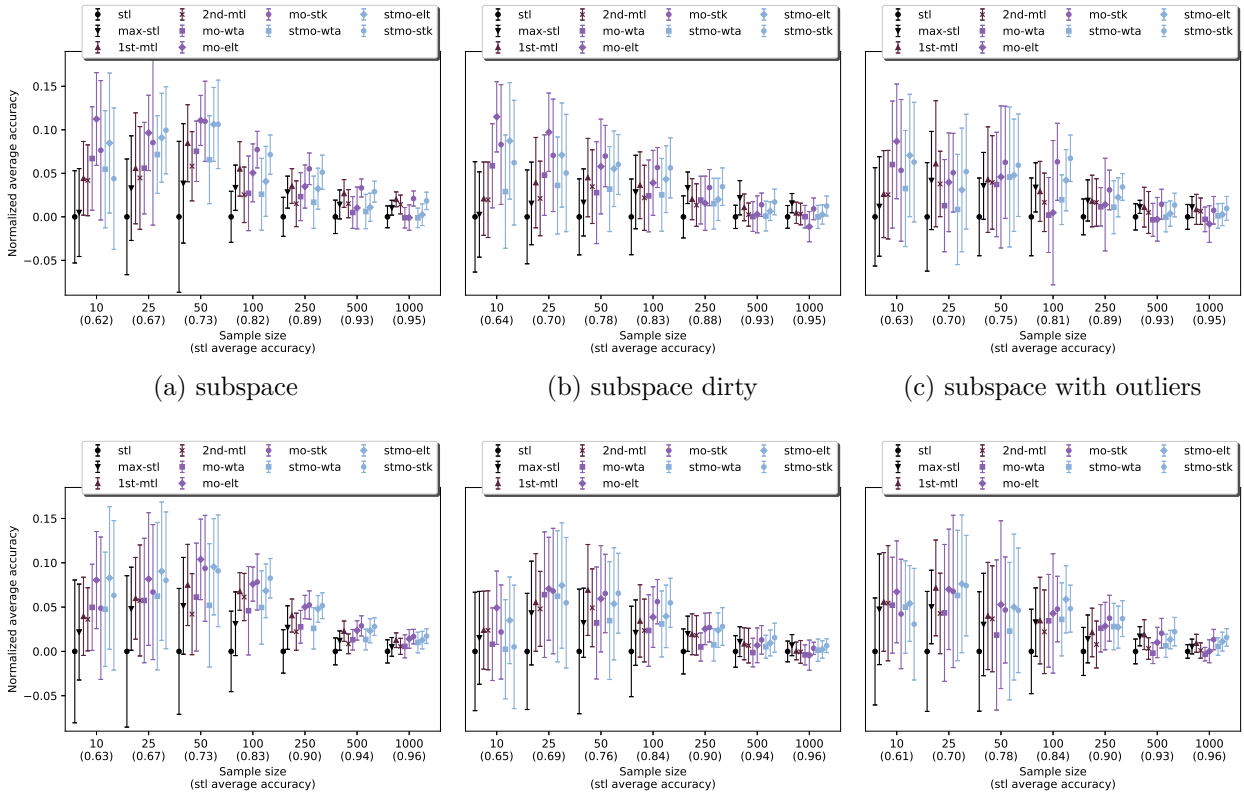
(d) single cluster (e) single cluster dirty (f) single cluster with outliers

(g) three clusters (h) three clusters dirty (i) three clusters with outliers

Figure 31 – Normalized average accuracy (and standard deviation) for distinct classifiers grouped by sample size for synthetic datasets (Part I). On each group label there is the sample size, and the accuracy of **stl**, inside parenthesis, which was subtracted from every method's average accuracy inside that group.

best method. Besides, **stk** methods also have its comparative performance improved when the size of the dataset increases, even outperforming **elt**.

The different noise profiles do not seem to heavily interfere in the relative performance of the methods under analysis. But it is noticeable a drop in the performance gap between



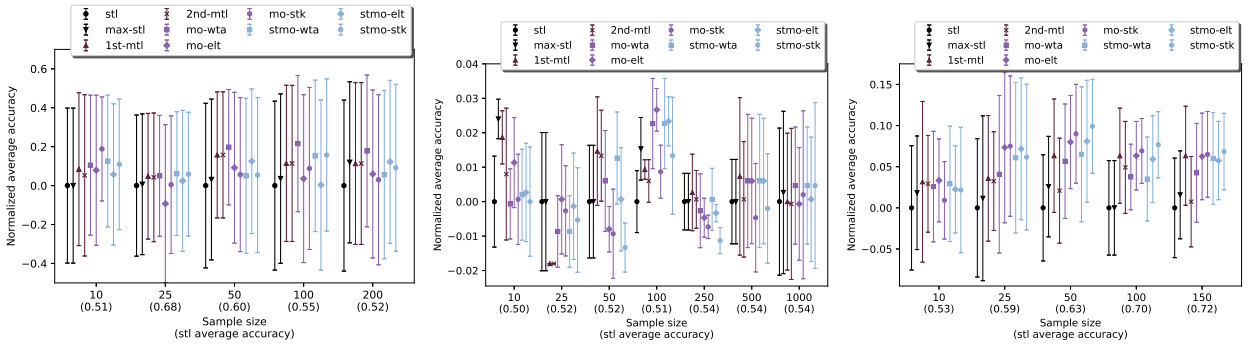
(a) subspace (b) subspace dirty (c) subspace with outliers

(d) relationship (e) relationship dirty (f) relationship with outliers

Figure 32 – Normalized average accuracy (and standard deviation) for distinct classifiers grouped by sample size for synthetic datasets (Part II). On each group label there is the sample size, and the accuracy of **stl**, inside parenthesis, which was subtracted from every method's average accuracy inside that group.

single-task learning and multi-task learning methods when comparing tasks with and without outliers. This effect sometimes acts more heavily on the multi-task-only proposed methods (**mo-wta**, **mo-elt** and **mo-stk**). The mixed single-task and multi-task models (**stmo-wta**, **stmo-elt** and **stmo-stk**) taken as components of the ensemble helped on dealing with outliers as well as independent tasks. The hybrid aggregation of single and multi-task models has proved advantageous on these scenarios.

On the real datasets, we can detect three distinct behaviours. The landmine dataset seems to be an "easy" dataset, guiding to a high performance even for single-task models on small datasets. In this scenario, we did not notice an increase in performance when applying our methods. The performance on the spam dataset with three users indicates a positive transfer on small datasets (reaching the performance of 1st-mtl), but a negative transfer on large datasets. Finally, the performance on the spam dataset with 15 users is robust and guides to the best performance on almost all sample sizes. Reduced datasets and /or an



(a) landmine (b) spam with 3 users (c) spam with 15 users

Figure 33 – Normalized average accuracy (and standard deviation) for distinct classifiers grouped by sample size for real datasets. On each group label there is the sample size, and the accuracy of **stl**, inside parenthesis, which was subtracted from every method's average accuracy inside that group.

increase in the number of tasks, both contributing to more challenging scenarios, seem to favour our proposal.

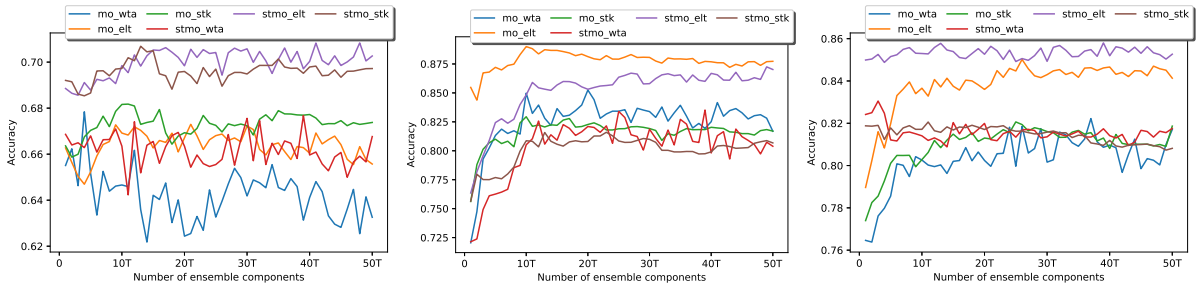
6.4.5 Sensitivity to the number of ensemble components

6.4.5.1 Results

To better explain the performance of the proposed method, we design an experiment to study the impact of the number of ensemble components generated by the multi-objective training of multi-task models. Looking at the results of Section 6.4.4, it is possible to see distinct behaviors in terms of performance for models with and without single-task trained models (**stmo**). Given that, we decided to pick some scenarios with these distinct behaviors to better study the impact of the number of ensemble components:

- **Independent tasks with 25 samples** - In this dataset, it is possible to observe, in Figure 31, that there is some negative transferring mitigated by stl models.
- **Tasks related in a single cluster with 25 samples** - In this dataset, it is possible to observe, in Figure 31, that there is positive transferring hurt by stl models.
- **Tasks related in three clusters with outlier and 50 samples** - In this dataset, it is possible to observe, in Figure 31, that there is positive transferring improved by stl models.

In Figure 34 we show the experiment for each one of these scenarios, where the number of ensemble components generated by the multi-task multi-objective approach varies from $1 \times T$ to $50 \times T$ components. In the **stmo_*** approaches, these components complement the 25 models generated by the single-task trained models, unlike **mo_*** approaches that only have these multi-task components.



(a) Independent with 25 samples (b) Single cluster with 25 samples (c) Three clusters with 50 samples

Figure 34 – Performance of the proposed method varying the number of ensemble components generated by the multi-objective procedure.

6.4.5.2 Discussion

In Figure 34-a, negative transferring was verified when it is used only multi-task learning models (**mo_***). However, when the single-task learning models are applied, it is possible to see a better performance, and the performance is marginally improved with the addition of multi-task learning models. In Figure 34-b, positive transferring was verified when it is used only multi-task learning models (**mo_***). It is important to notice the increase in performance during the addition of multi-task learning models. It occurs in both methods, **mo_*** and **most_***, but the performance of the method that starts with single-task learning models was never able to achieve the same performance of **mo_***. In Figure 34-c, positive transferring was verified when it is used only multi-task learning models (**mo_***). The particularity of this case is the substantial improvement when it is applied single-task learning models, starting with a high performance that is only marginally improved by adding multi-task learning models.

Supported by these experiments, some hypotheses are raised . First of all, knowing that the single-task learning models from all tasks are jointly considered to compose the ensemble for each task, two behaviours are possible to observe: (1) in the independent tasks, the cross-validation procedure were not able to select the correct models and achieve the performance of default stl contender (notice that the performance of **stl** in Figure 31-a, was not achieved by **most_*** at the beginning of Figure 34-a, where there is only the influence

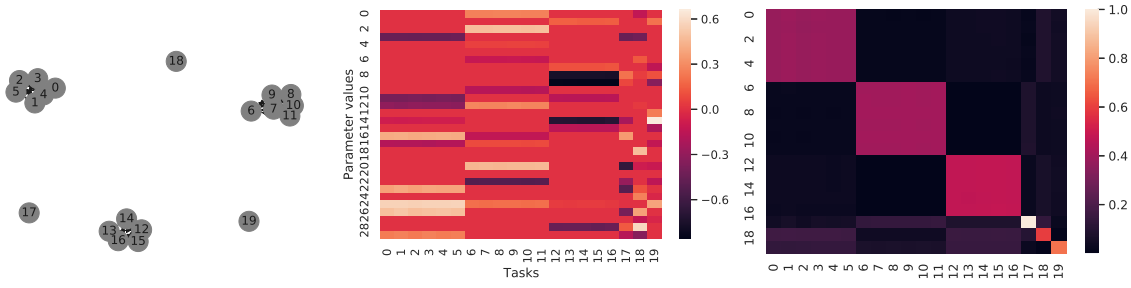
of single-task trained models); (2) in the datasets with task sharing, this aggregation works as a rudimentary knowledge sharing (it can be noticed comparing the performance of **stl** in Figure 31-i with the performance of **most_*** at the beginning of Figure 34-c).

Making an analysis of the increase in the number of ensemble components, it is possible to observe the lack of robustness coming from **wta** approaches, as well as the solid robustness of **elt** and **stk** approaches that seems, with little variation, to always take advantage of the increase in the number of components.

6.4.6 Analysis of knowledge sharing relations

6.4.6.1 Results and discussion

As we can see in Section 6.4.1.1, the definition of the synthetic datasets obeys some sharing structures. Aiming at studying the nature of this kind of sharing relations, we decided to investigate if the sharing structures, employed in this thesis, retain and recover these sharing relationships. The synthetic datasets are, again, a useful tool to make this investigation, since it is possible to check the structures that generated the datasets. For didactic purposes, Figure 35-a shows the generative graph of relations between the tasks of the base **three clusters** with outliers, Figure 35-b shows the heat-map of parameters for this base, and Figure 35-c shows a normalized similarity matrix of parameters for this base.



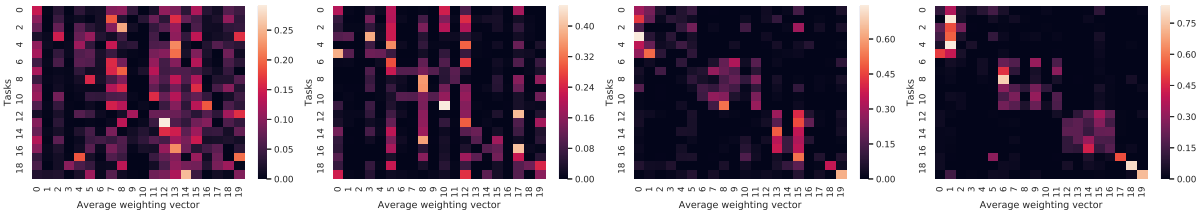
(a) Generative graph (b) Resultant parameters (c) Similarity matrix

Figure 35 – Representation of the generated relations and parameters for the dataset three clusters with outliers.

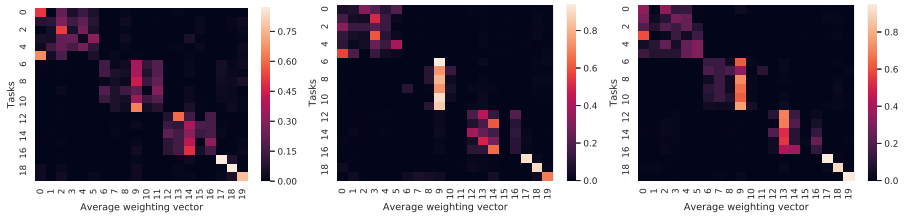
Since the **elt** ensemble is the most robust classifier in our methodology, we will use the 10 classifiers with higher performance in the validation to make the investigation. First of all, we are going to define the mean influence $\mathbf{u}^t$ of the weighted sum method's parameter $\mathbf{w}$ for all 10 best classifiers. To do so, we define the set $\mathcal{B}^t$ as the set in indexes of $\mathbf{w}$ which

generates the best models for the task t with respect to the performance in validation.

$$\mathbf{u}^t = [\frac{1}{10}\sum_{i\in\mathcal{B}^t}\mathbf{w}_1^{(i)}, \ldots, \frac{1}{10}\sum_{i\in\mathcal{B}^t}\mathbf{w}_T^{(i)}]^\top \tag{6.3}$$



(a) 10 samples (b) 25 samples (c) 50 samples (d) 100 samples



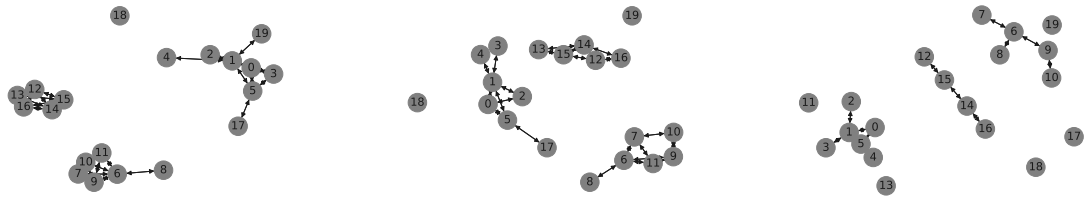
(e) 250 samples (f) 500 samples (g) 1000 samples

Figure 36 – Representation of resultant mean influence of the many-objective trained multi-task models for the dataset three clusters with outliers.

Given that, we can plot a matrix $U : U_i^t = \mathbf{u}_i^t$ in Figure 36 for every size of training set. We can see that, with the increase in the dataset size in training, the similarity with the correct relations increase.

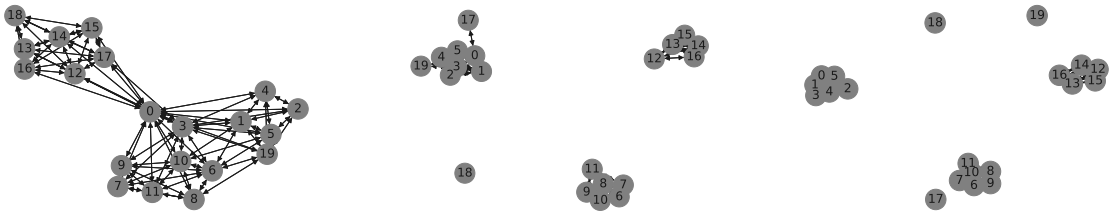
To clean the noise of the relations and reconstruct the graph of relations, we proposed a rule that tries to infer if there is a relation between two tasks. So if $u_i^j \geq \epsilon$ or $u_j^i \geq \epsilon$, we consider that there exists an edge connecting the tasks i and j in both directions. We call this rule "**w influence**". For the three clusters with outliers and 100 samples, we plotted the graph in Figure 37.

However, it is possible to see in Figure 37 that, despite the cluster structure is being captured by the method, not always there exists a pairwise relation involving the members of each cluster. To mitigate that, we proposed another two methods to infer these relations: (1) "**w similarity**" calculates a normalized distance of the mean influence, resulting in the following rule $\frac{2-||\mathbf{u}^i-\mathbf{u}^i||_1}{2} \geq \epsilon$, and having the recovered relation shown in Figure 38; (2) "**component influence**" calculates the number of ensemble members that each task share, resulting in the following rule $\frac{|\mathcal{B}^i\cup\mathcal{B}^j|}{10} \geq \epsilon$, and having the recovered relations shown in Figure 39.



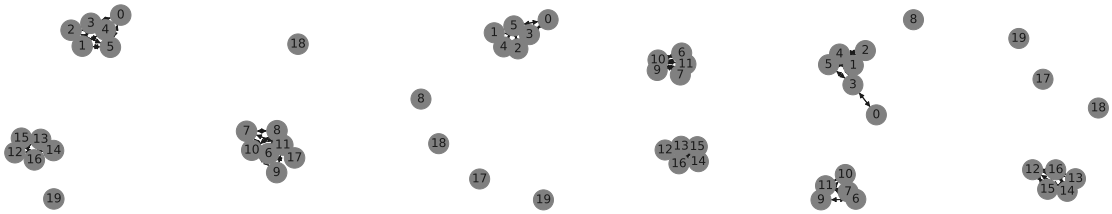
(a) $\epsilon = 0.1$ (b) $\epsilon = 0.2$ (c) $\epsilon = 0.3$

Figure 37 – Representation of the generated relations and parameters for the dataset three clusters with outliers, using "w influence".



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.2$ (c) $\epsilon = 0.3$

Figure 38 – Representation of the generated relations and parameters for the dataset three clusters with outliers, using "w similarity".



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.2$ (c) $\epsilon = 0.3$

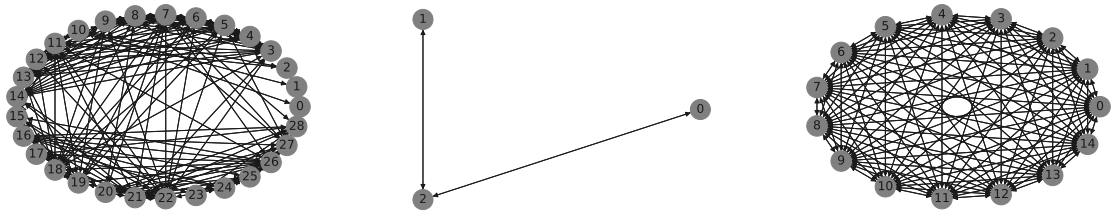
Figure 39 – Representation of the generated relations and parameters for the dataset three clusters with outliers, using "component influence".

Aiming at making an extensive study of the recovering of task relations, we fixed the number of samples in training as 100 samples and varied, for every synthetic dataset and rule of inference of the recovering method, the threshold of those methods $\epsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The accuracy, calculated by the number of correctly recovered edges, is shown in Table 11.

We can see that the **w similarity** is the most effetive method to recover the task

Table 11 – Rate of correctly recovered connections between the tasks in w.r.t. the generative relationships.

	w influence					**w similarity**					**component influence**				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
independent	0.920	0.989	0.994	0.989	0.987	0.239	0.764	0.939	0.989	0.994	0.930	0.979	0.989	0.999	0.999
independent_dirty	0.922	0.974	0.989	0.984	0.974	0.104	0.584	0.869	0.974	0.994	0.949	0.989	0.989	0.994	0.994
independent_outlier	0.874	0.964	0.989	0.989	0.974	0.080	0.405	0.874	0.979	0.999	0.939	0.989	0.994	0.999	0.999
1_cluster	0.307	0.124	0.085	0.020	0.005	0.999	0.999	0.999	0.959	0.775	0.499	0.195	0.094	0.059	0.050
1_cluster_dirty	0.260	0.130	0.077	0.042	0.014	0.999	0.994	0.925	0.764	0.515	0.560	0.354	0.160	0.094	0.070
1_cluster_outlier	0.477	0.357	0.307	0.284	0.275	0.749	0.839	0.915	0.959	0.819	0.874	0.609	0.434	0.349	0.325
3_clusters	0.869	0.827	0.729	0.692	0.675	0.714	0.949	0.984	0.994	0.949	0.984	0.920	0.824	0.770	0.729
3_clusters_dirty	0.869	0.795	0.747	0.709	0.689	0.704	0.989	0.984	0.969	0.930	0.959	0.900	0.834	0.810	0.770
3_clusters_outlier	0.925	0.874	0.824	0.800	0.780	0.790	0.984	0.999	0.979	0.959	0.979	0.949	0.939	0.910	0.869
subspace	0.262	0.102	0.052	0.027	0.002	0.989	0.854	0.569	0.330	0.170	0.225	0.089	0.059	0.054	0.050
subspace_dirty	0.232	0.112	0.059	0.020	0.010	0.999	0.879	0.609	0.379	0.200	0.344	0.160	0.085	0.059	0.050
subspace_outlier	0.472	0.359	0.299	0.275	0.275	0.719	0.650	0.624	0.579	0.499	0.460	0.414	0.354	0.325	0.320
relationship	0.572	0.472	0.455	0.419	0.419	0.660	0.785	0.714	0.704	0.685	0.689	0.665	0.634	0.609	0.574
relationship_dirty	0.592	0.499	0.465	0.427	0.412	0.714	0.800	0.729	0.665	0.589	0.604	0.564	0.540	0.494	0.465
relationship_outlier	0.714	0.634	0.602	0.587	0.574	0.744	0.829	0.819	0.759	0.724	0.704	0.689	0.694	0.680	0.650
mean	0.618	0.548	0.512	0.484	0.471	0.680	0.821	**0.837**	0.799	0.720	0.714	0.631	0.575	0.547	0.528
heterogeneous	0.757	0.684	0.637	0.606	0.592	0.721	**0.889**	0.872	0.845	0.806	0.820	0.781	0.744	0.712	0.676
single sharing	0.429	0.337	0.297	0.273	0.258	0.756	**0.770**	0.753	0.665	0.534	0.567	0.436	0.348	0.314	0.302
outlier	0.647	0.556	0.508	0.486	0.476	0.751	0.826	**0.839**	0.820	0.751	0.754	0.666	0.606	0.566	0.541



(a) landmine (b) spam a (c) spam b

Figure 40 – Representation of the recovered task relations for the real datasets.

relations, mainly with $\epsilon = 0.2$ or $\epsilon = 0.3$. Supported by these results we run the method **w similarity** with $\epsilon = 0.3$ for the real datasets where there are the most active positive transferring for each one of the datasets (landmine detection with 10 samples, spam with 3 users and 50 samples, spam with 15 users and 50 samples). These results are depicted in Figure 40 suggesting a single sharing structure for the datasets *spam a* and *spam b*. The dataset *landmine* has a known structure of two clusters of tasks 0 to 15 and 16 to 28. This structure was partially recovered having a strong separation between the clusters but the density inside each cluster was not very strong.

Overall, the robust performance and interpretation capability of the multi-objective method proposed in this thesis turns this methodology into a competent method for multi-task learning.

Chapter 7

Conclusion

The main scientific contribution of this thesis was to study the impact of multi-objective optimization in machine learning. To deeply investigate this, we proposed a framework that formulates machine learning models as multi-objective problems and solves it with a posteriori multi-objective methods. Since this class of methods generates a pool of solutions, we saw these solutions as candidates to compose an ensemble and explored a diverse set of efficient candidate solutions with ensemble filtering and aggregations methods. We proposed multi-objective formulations for the following relevant problems in machine learning: multi-class classification, multi-class classification with imbalanced classes, multi-label classification, multi-task classification, multi-view learning and transfer learning. Supported by this vast application scenario, it is safe to say that the proposed framework supports the possibility of formulating other machine learning models as a multi-objective problem.

In addition to being general and flexible, the proposed framework is also compelling. The experimental results support the sampling capability for model selection, sampling that was also capable of generating diversity for the ensemble components. These results are relevant since they connect the sampling and diversity concepts of multi-objective optimization with equivalent machine learning concepts, and allows an in-depth looking at relevant trade-offs in machine learning. As expected, these capabilities supported a consistent performance in multi-class classification, the problem that was further explored by enhancing the flexibility of the model by facing the loss of each class as a conflicting objective. The results showed that, by just promoting more flexibility to the class losses, it could enhance performance in the class imbalance context.

It is worth mentioning the originality and simplicity of the models for multi-task classification, multi-label classification, and transfer learning: it was considered a single parameter

vector for all tasks. Even with that simplicity of the model, the multi-objective framework was able to promote good levels of knowledge transfer, supported by vast scenarios induced into synthetic datasets that were designed to be as diverse as possible. Multiple real-world datasets also supported it in the detection of epileptic seizures, multi-label classification, and multi-task classification. It is important to highlight the relevance of ensemble approaches in our framework, with the application of multiple models being the key to promote good behavior in datasets characterized by more than one sharing structure. This framework also allowed the insertion of single-task trained models, which was fundamental to mitigate negative transferring as well as improve the performance in scenarios with outlier tasks.

Another relevant connection between multi-objective optimization and machine learning resides in the final experiments of multi-task learning: we find that increasing the quality of the representation in the multi-objective perspective also increases the quality of the classifier until a certain threshold level, with a stable performance after that threshold; both findings are significant, it highlights the connection between the fields and also shows the robustness of the framework, because it is necessary only to give enough candidates to achieve a threshold from which the maximal performance is achieved, keeping a reasonable performance after that. We also found that scalarization weights are capable of depicting the transferring level between the tasks. It creates another connection now between weighting sum method and multi-task learning, showing the capability of analyzing the relations between the tasks only using the weights of the multi-objective scalarization.

The promising performance of the proposed framework in several case studies helps shedding more light on the interplay of multi-objective optimization and machine learning. Convexity is certainly a key aspect of the adopted learning models, and a straightforward extension involves considering other convex learning models such as linear regression and kernel regression.

Bibliography

ABBASS, H. A. Pareto Neuro-Evolution: Constructing Ensemble. In: *Congress on Evolutionary Computation*. [S.l.: s.n.], 2003. v. 3, p. 2074–2080. Cited 2 times on pages 66 and 67.

AHMADIAN, K.; GOLESTANI, A.; MOZAYANI, N.; KABIRI, P. A new multi-objective evolutionary approach for creating ensemble of classifiers. In: *IEEE International Conference on Systems, Man and Cybernetics*. [S.l.: s.n.], 2007. p. 1031–1036. Cited 2 times on pages 66 and 67.

AKAN, A.; SAYIN, S. SVM classification for imbalanced data sets using a multiobjective optimization framework. *Ann Oper Res*, v. 216, p. 191–203, 2014. Cited 2 times on pages 65 and 66.

ALBUKHANAJER, W. A.; JIN, Y.; BRIFFA, J. A. Classifier ensembles for image identification using multi-objective Pareto features. *Neurocomputing*, Elsevier B.V., v. 238, p. 316–327, 2017. Cited on page 66.

ANDO, R. K.; TONG, Z. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, v. 6, p. 1817–1853, 2005. Cited 3 times on pages 54, 74, and 75.

ARGYRIOU, A.; EVGENIOU, T.; PONTIL, M. Multi-Task Feature Learning. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2006. p. 41—-48. Cited on page 75.

ARGYRIOU, A.; EVGENIOU, T.; PONTIL, M.; ARGYRIOU, A.; PONTIL, M.; EVGENIOU, T. Convex multi-task feature learning. *Machine Learning*, v. 73, n. 3, p. 243–272, jan 2008. Cited on page 75.

BAGHERJEIRAN, A. *Multi-objective multi-task learning*. Tese (Doutorado) — University of Houston, 2007. Cited 2 times on pages 58 and 65.

BAI, J.; ZHOU, K.; XUE, G.; ZHA, H.; SUN, G.; TSENG, B.; ZHENG, Z.; CHANG, Y. Multi-task learning for learning to rank in web search. In: *ACM conference on Information and knowledge management*. New York, New York, USA: ACM Press, 2009. p. 1549–1552. Cited on page 75.

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KÉGL, B. Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems (NIPS)*, p. 2546–2554, 2011. Cited on page 69.

BERGSTRA, J.; BENGIO, Y. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, v. 13, p. 281–305, 2012. Cited 2 times on pages 68 and 69.

BESERRA, F. S.; RAIMUNDO, M. M.; B, F. J. V. Z. Ensembles of Multiobjective-Based Classifiers for Detection of Epileptic Seizures. In: *Lecture Notes in Computer Science, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017*. [S.l.: s.n.], 2018. v. 10657, p. 575–583. Cited 2 times on pages 59 and 94.

BHOWAN, U.; JOHNSTON, M.; ZHANG, M. Ensemble learning and pruning in multi-objective genetic programming for classification with unbalanced data. In: *Advances in Artificial Intelligence*. [S.l.: s.n.], 2011. p. 192–202. Cited on page 66.

BHOWAN, U.; JOHNSTON, M.; ZHANG, M. Evolving ensembles in Multi-objective Genetic Programming for classification with unbalanced data. In: *Genetic and Evolutionary Computation Conference*. [S.l.: s.n.], 2011. p. 1331–1338. Cited on page 66.

BHOWAN, U.; JOHNSTON, M.; ZHANG, M. Comparing ensemble learning approaches in genetic programming for classification with unbalanced data. In: *Conference Companion on Genetic and Evolutionary Computation*. [S.l.: s.n.], 2013. p. 135. Cited on page 66.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006. ISBN 9780387310732. Cited 2 times on pages 45 and 67.

BRADFORD, J.; KUNZ, C.; KOHAVI, R.; BRUNK, C.; BRODLEY, C. Pruning decision trees with misclassification costs. *Machine Learning: ECML- 98*, v. 1398, p. 131 – 136, 1998. Cited on page 71.

BRAGA, A. P.; TAKAHASHI, R. H. C.; COSTA, M. A.; TEIXEIRA, R. D. A. Multi-Objective Algorithms for Neural Networks Learning. In: *Multi-Objective Machine Learning*. [S.l.: s.n.], 2006. v. 16, n. Part II, p. 151–171. Cited on page 65.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, n. 2, p. 123–140, 1996. Cited on page 70.

BREIMAN, L. Random forests. *Machine learning*, v. 45, n. 1, p. 5–32, 2001. Cited 2 times on pages 70 and 75.

CAMILLERI, M.; NERI, F. An Algorithmic Approach to Parameter Selection in Machine Learning using Meta-Optimization Techniques. *WSEAS Transactions on Systems*, v. 13, p. 203–212, 2014. Cited on page 69.

CARUANA, R. Multitask Learning. *Machine Learning*, v. 75, n. 1, p. 41–75, 1997. Cited 2 times on pages 72 and 75.

CARUANA, R. A Dozen Tricks with Multitask Learning. In: *Neural Networks: Tricks of the Trade*. [S.l.: s.n.], 1998. p. 165–191. Cited on page 73.

CASTILLO, P.; ARENAS, M.; MERELO, J.; RIVAS, V.; ROMERO, G. Multiobjective optimization of ensembles of multilayer perceptrons for pattern classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 4193 LNCS, p. 453–462, 2006. Cited on page 66.

CHANDRA, A.; CHEN, H.; YAO, X. Trade-off between diversity and accuracy in ensemble generation. In: *Studies in Computational Intelligence*. [S.l.: s.n.], 2006. v. 16, p. 429–464. Cited on page 66.

CHANDRA, A.; YAO, X. DIVACE: Diverse and accurate ensemble learning algorithm. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 3177, p. 619–625, 2004. Cited on page 66.

CHANDRA, A.; YAO, X. Ensemble learning using multi-objective evolutionary algorithms. *Journal of Mathematical Modelling and Algorithms*, v. 5, n. 4, p. 417–445, 2006. Cited on page 66.

CHANG, C.; LIN, C. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, v. 2, p. 1–39, 2011. Cited on page 68.

CHAPELLE, O.; SHIVASWAMY, P.; VADREVU, S.; WEINBERGER, K.; ZHANG, Y.; TSENG, B. Boosted multi-task learning. *Machine Learning*, v. 85, n. 1-2, p. 149–173, 2011. Cited 2 times on pages 74 and 75.

CHARTE, F.; RIVERA, A. J.; Del Jesus, M. J.; HERRERA, F. MLeNN: A first approach to heuristic multilabel undersampling. *Lecture Notes in Computer Science*, v. 8669 LNCS, p. 1–9, 2014. Cited on page 72.

CHARUVAKA, A.; RANGWALA, H. Convex multi-task relationship learning using hinge loss. In: *Symposium on Computational Intelligence and Data Mining*. [S.l.: s.n.], 2015. p. 63–70. Cited on page 75.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O. SMOTE : Synthetic Minority Over-sampling Technique. *Artificial Intelligence*, v. 16, p. 321–357, 2002. Cited on page 71.

CHAWLA, N. V.; CIESLAK, D. A.; HALL, L. O.; JOSHI, A. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, v. 17, n. 2, p. 225–252, 2008. Cited on page 71.

CHAWLA, N. V.; HALL, L. O.; JOSHI, A. Wrapper-based computation and evaluation of sampling methods for imbalanced datasets. *Proceedings of the 1st international workshop on Utility-based data mining - UBDM '05*, p. 24–33, 2005. Cited on page 71.

CHAWLA, N. V.; LAZAREVIC, A.; HALL, L. O.; BOWYER, K. W. SMOTEBoost : Improving Prediction. *Lecture Notes in Computer Science*, v. 2838, p. 107–119, 2003. Cited on page 71.

CHEN, J.; LIU, J.; YE, J. Learning Incoherent Sparse and Low-Rank Patterns from Multiple Tasks. *ACM Transactions on Knowledge Discovery from Data*, v. 5, n. 4, p. 1–31, 2012. Cited 2 times on pages 75 and 113.

CHEN, J.; TANG, L.; LIU, J.; YE, J. A convex formulation for learning shared structures from multiple tasks. In: *International Conference on Machine Learning*. New York, New York, USA: ACM Press, 2009. p. 137—-144. Cited 3 times on pages 74, 75, and 113.

CHEN, J.; TANG, L.; LIU, J.; YE, J. A convex formulation for learning a shared predictive structure from multiple tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 5, p. 1025–38, may 2013. Cited on page 75.

CHEN, J.; ZHOU, J.; YE, J. Integrating low-rank and group-sparse structures for robust multi-task learning. In: *International Conference on Knowledge Discovery and Data Mining*. [S.l.]: ACM Press, 2011. p. 42–50. Cited on page 75.

CHEN, S.; HE, H.; GARCIA, E. A. RAMOBoost: Ranked minority oversampling in boosting. *IEEE Transactions on Neural Networks*, v. 21, n. 10, p. 1624–1642, 2010. Cited on page 71.

COELHO, G. P. *Geração, Seleção e Combinação de Componentes para Ensembles de Redes Neurais Aplicadas a Problemas de Classificação*. Tese (Doutorado), 2006. Cited on page 70.

COHEN, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, v. 20, n. 1, p. 37–46, 1960. Cited on page 87.

COHON, J. *Multiobjective programming and planning*. [S.l.: s.n.], 1978. v. 140. Cited 3 times on pages 33, 36, and 67.

COHON, J. L.; CHURCH, R. L.; SHEER, D. P. Generating multiobjective trade-offs: An algorithm for bicriterion problems. *Water Resources Research*, v. 15, n. 5, p. 1001–1010, 1979. Cited 5 times on pages 27, 28, 39, 44, and 76.

CORTES, C.; VAPNIK, V. Support-Vector Networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. Cited on page 67.

COSTA, N.; COELHO, A. L. V. Genetic and ranking-based selection of components for multilabel classifier ensembles. *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*, p. 311–317, 2011. Cited on page 73.

DAS, I.; DENNIS, J. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, v. 14, n. 1, p. 63–69, 1997. Cited on page 34.

DATTA, S.; DAS, S. Near-Bayesian Support Vector Machines for imbalanced data classification with equal or unequal misclassification costs. *Neural Networks*, Elsevier Ltd, v. 70, p. 39–52, 2015. Cited on page 71.

DEMBCZY, K. Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. *Proceedings of the 27th International Conference on Machine Learning*, p. 279–286, 2010. Cited on page 72.

DHULEKAR, N.; NAMBIRAJAN, S.; OZTAN, B.; YENER, B. Seizure Prediction by Graph Mining, Transfer Learning, and Transformation Learning. In: *Proceedings of the 11th International Conference on Machine Learning and Data Mining in Pattern Recognition - Volume 9166*. [S.l.]: Springer-Verlag New York, Inc., 2015. p. 32–52. Cited 2 times on pages 94 and 95.

DOS SANTOS, E. M.; SABOURIN, R.; MAUPIN, P. Pareto analysis for the selection of classifier ensembles. In: *Conference on Genetic and Evolutionary Computation*. [S.l.: s.n.], 2008. p. 681–688. ISBN 9781605581309. Cited on page 66.

EKBAL, A.; SAHA, S. Classifier ensemble using multiobjective optimization for named entity recognition. *Frontiers in Artificial Intelligence and Applications*, v. 215, p. 783–788, 2010. Cited on page 66.

EKBAL, A.; SAHA, S. Multiobjective optimization for classifier ensemble and feature selection: An application to named entity recognition. *International Journal on Document Analysis and Recognition*, v. 15, n. 2, p. 143–166, 2012. Cited on page 66.

EKBAL, A.; SAHA, S. Simultaneous feature and parameter selection using multiobjective optimization: application to named entity recognition. *International Journal of Machine Learning and Cybernetics*, Springer Berlin Heidelberg, v. 7, n. 4, p. 597–611, 2016. Cited on page 66.

ENGEN, V.; VINCENT, J.; SCHIERZ, A. C.; PHALP, K. Multi-objective evolution of the Pareto optimal set of neural network classifier ensembles. In: *International Conference on Machine Learning and Cybernetics*. [S.l.: s.n.], 2009. v. 1, p. 74–79. Cited 3 times on pages 66, 67, and 74.

EVGENIOU, T.; PONTIL, M. Regularized multi-task learning. In: *International conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2004. p. 109–117. Cited on page 75.

FADDOUL, J. B.; CHIDLOVSKII, B.; GILLERON, R.; TORRE, F. Learning multiple tasks with boosted decision trees. In: *Machine Learning and Knowledge Discovery in Databases*. [S.l.]: Springer Berlin Heidelberg, 2012. v. 7523, p. 681–696. Cited on page 75.

FARQUHAR, J.; HARDOON, D.; MENG, H.; SHAWE-TAYLOR, J. S.; SZEDMAK, S. Two view learning: SVM-2K, theory and practice. *Advances in neural information processing systems*, p. 355–362, 2005. Cited on page 76.

FERNÁNDEZ CABALLERO, J. C.; MARTINEZ, F. J.; HERVAS, C.; GUTIERREZ, P. A. Sensitivity versus accuracy in multiclass problems using memetic Pareto evolutionary neural networks. *IEEE Transactions on Neural Networks*, v. 21, n. 5, p. 750–770, 2010. Cited 2 times on pages 65 and 66.

FERNÁNDEZ-DELGADO, M.; CERNADAS, E.; BARRO, S.; AMORIM, D. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, v. 15, p. 3133–3181, 2014. Cited 3 times on pages 78, 79, and 87.

FINNER, H. On a Monotonicity Problem in Step-Down Multiple Test Procedures. *Journal of the American Statistical Association*, v. 88, n. 423, p. 920–923, 1993. Cited 3 times on pages 80, 84, and 91.

FRATELLO, M.; CAIAZZO, G.; TROJSI, F.; RUSSO, A.; TEDESCHI, G.; TAGLIAFERRI, R.; ESPOSITO, F. Multi-View Ensemble Classification of Brain Connectivity Images for Neurodegeneration Type Discrimination. *Neuroinformatics*, Neuroinformatics, v. 15, n. 2, p. 199–213, 2017. Cited on page 77.

FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, v. 32, n. 200, p. 675–701, 1937. Cited 3 times on pages 80, 84, and 91.

GARCÍA, S.; ALER, R.; GALVÁN, I. M. Using evolutionary multiobjective techniques for imbalanced classification data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6352 LNCS, n. PART 1, p. 422–427, 2010. Cited 2 times on pages 65 and 66.

GEOFFRION, A. M. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, v. 22, n. 3, p. 618–630, 1968. Cited on page 34.

GIANNAKAKIS, G.; SAKKALIS, V.; PEDIADITIS, M.; TSIKNAKIS, M. Methods for Seizure Detection and Prediction: An Overview. In: *Modern Electroencephalographic Assessment Techniques: Theory and Applications*. [S.l.]: Springer New York, 2015. p. 131–157. ISBN 978-1-4939-1298-8. Cited on page 94.

GOLDBERGER, A. L.; AMARAL, L. A. N.; GLASS, L.; HAUSDORFF, J. M.; IVANOV, P. C.; MARK, R. G.; MIETUS, J. E.; MOODY, G. B.; PENG, C.; STANLEY, H. E. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, American Heart Association, Inc., v. 101, n. 23, p. e215—-e220, 2000. Cited on page 94.

GONÇALVES, A. R.; SIVAKUMAR, V.; ZUBEN, F. J. V.; BANERJEE, A. Multi-task Sparse Structure Learning. In: *International Conference on Conference on Information and Knowledge Management*. New York, New York, USA: ACM Press, 2014. p. 451–460. Cited on page 75.

GONÇALVES, A. R.; ZUBEN, F. J. V.; BANERJEE, A. Multi-Label Structure Learning with Ising Model Selection. In: *Proccedings of 24th International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. p. 3525–3531. Cited 3 times on pages 54, 72, and 107.

GONG, P.; YE, J.; ZHANG, C. Robust Multi-Task Feature Learning. In: *International Conference on Knowledge Discovery and Data Mining*. New York, New York, USA: ACM Press, 2012. p. 895–903. Cited 4 times on pages 54, 75, 111, and 113.

GONG, P.; YE, J.; ZHANG, C.; ZOU, H. Multi-Stage Multi-Task Feature Learning. *Journal of Machine Learning Research*, v. 14, p. 2979–3010, 2013. Cited on page 74.

GROSS, R. *Psychology: The Science of Mind and Behavior*. [S.l.: s.n.], 2010. 912 p. ISBN 9781444108316. Cited on page 22.

GUO, H.; VIKTOR, H. L. Learning from Imbalanced Data Sets with Boosting and Data Generation : The DataBoost-IM Approach. *ACM SIGKD Explorations Newsletter - Special issue on learning from imbalanced datasets*, v. 6, n. 1, p. 30–39, 2004. Cited on page 71.

HAN, H.; WANG, W.-y.; MAO, B.-h. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: . [S.l.: s.n.], 2005. p. 878–887. Cited on page 71.

HAN, L.; ZHANG, Y. Learning Multi-Level Task Groups in Multi-Task Learning. In: *AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. p. 2638–2644. Cited on page 75.

HAN, L.; ZHANG, Y. Learning Tree Structure in Multi-Task Learning. In: *International Conference on Knowledge Discovery and Data Mining*. New York, New York, USA: ACM Press, 2015. p. 397–406. Cited on page 75.

HASTIE, T.; TIBSHARANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. [S.l.: s.n.], 2009. Cited on page 22.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. Adaptive Synthetic Sampling Approach for Imbalanced Learning. n. 3, p. 1322–1328, 2008. Cited on page 71.

HUANG, G.; ZHOU, H.; DING, X.; ZHANG, R. Extreme learning machine for regression and multiclass classification. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics*, v. 42, n. 2, p. 513–529, 2012. Cited 3 times on pages 68, 80, and 90.

IGEL, C. Multi-objective model selection for support vector machines. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2005. v. 3410, p. 534–546. Cited on page 65.

ISHIBUCHI, H.; NOJIMA, Y. Difficulties in choosing a single final classifier from non-dominated solutions in multiobjective fuzzy genetics-based machine learning. In: *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. [S.l.: s.n.], 2013. p. 1203–1208. Cited on page 67.

JALALI, A.; RAVIKUMAR, P.; SANGHAVI, S.; RUAN, C. A dirty model for multi-task learning. In: *Conference on Neural Information Processing Systems*. [S.l.: s.n.], 2010. p. 964–972. Cited 3 times on pages 75, 111, and 113.

JAPKOWICZ, N.; MATWIN, S. Multi-label Classification via Multi-target Regression on Data Streams. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2015. v. 9356, n. 1, p. 170–185. Cited on page 73.

JIN, Y.; SENDHOFF, B. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, v. 38, n. 3, p. 397–415, may 2008. Cited 3 times on pages 25, 65, and 74.

JIN, Y. Y.; GRUNA, R. R.; SENDHOFF, B. B. Pareto analysis of evolutionary and learning systems. *Frontiers of Computer Science in China*, v. 3, n. 1, p. 4–17, 2009. Cited 2 times on pages 65 and 74.

JUBRIL, A. A nonlinear weights selection in weighted sum for convex multiobjective optimization. *Facta universitatis-series: Mathematics and Informatics*, v. 27, n. 3, p. 357–372, 2012. Cited on page 35.

KIM, H.; PAIK, J. Low-Rank Representation-Based Object Tracking Using Multitask Feature Learning with Joint Sparsity. *Abstract and Applied Analysis*, Hindawi, v. 2014, p. 1–12, nov 2014. Cited on page 74.

KOSKI, J. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods*, v. 1, n. May, p. 333–337, 1985. Cited on page 34.

KRAMER, M. A.; KOLACZYK, E. D.; KIRSCH, H. E. Emergent network topology at seizure onset in humans. *Epilepsy Research*, v. 79, p. 173–186, 2008. Cited on page 95.

KRAUS, J. M.; MÜSSEL, C.; PALM, G.; KESTLER, H. A. Multi-objective selection for collecting cluster alternatives. *Computational Statistics*, v. 26, n. 2, p. 341–353, 2011. Cited 2 times on pages 62 and 66.

KRAWCZYK, B.; WOZNIAK, M. Accuracy and diversity in classifier selection for one-class classification ensembles. In: *IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL)*. [S.l.: s.n.], 2013. p. 46–51. Cited 2 times on pages 66 and 67.

KRAWCZYK, B.; WOŹNIAK, M. Optimization Algorithms for One-Class Classification Ensemble Pruning. In: *Asian Conference on Intelligent Information and Database Systems (ACIIDS)*. [S.l.: s.n.], 2014. p. 127–136. Cited 2 times on pages 66 and 67.

KRSTAJIC, D.; BUTUROVIC, L. J.; LEAHY, D. E.; THOMAS, S. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, Journal of Cheminformatics, v. 6, n. 1, p. 1–15, 2014. Cited on page 68.

KULAIF, A. C. P.; ZUBEN, F. J. V. Improved Regularization in Extreme Learning Machines. *Anais do Congresso Brasileiro de Inteligência Computacional (CBIC)*, 2013. Cited on page 69.

LAROCHELLE, H.; ERHAN, D.; COURVILLE, A.; BERGSTRA, J.; BENGIO, Y. An empirical evaluation of deep architectures on problems with many factors of variation. *International Conference on Machine Learning*, n. 2006, p. 473–480, 2007. Cited on page 68.

LATKA, M.; WAS, Z.; KOZIK, A.; WEST, B. J. Wavelet analysis of epileptic spikes. *Physical Review E*, American Physical Society, v. 67, n. 5, p. 52902, 2003. Cited 2 times on pages 94 and 95.

LI, C.; GEORGIOPOULOS, M.; ANAGNOSTOPOULOS, G. C. Pareto-path multitask multiple kernel learning. *IEEE transactions on neural networks and learning systems*, Institute of Electrical and Electronics Engineers Inc., v. 26, n. 1, p. 51–61, jan 2015. Cited on page 58.

LI, C.; LI, H. Network-constrained Regularization and Variable Selection for Analysis of Genomic Data. *Bioinformatics*, v. 24, n. 9, p. 1175–1182, 2007. Cited on page 75.

LIN, Y.; LEE, Y.; WAHBA, G. Support vector machines for classification in nonstandard situations. *Machine Learning*, v. 46, n. 1-3, p. 191–202, 2002. Cited on page 71.

LIU, J.; JI, S.; YE, J. Multi-Task Feature Learning Via Efficient L2,1-Norm Minimization. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. [S.l.: s.n.], 2009. p. 339–348. Cited on page 74.

LIU, X.; WU, J.; ZHOU, Z. Exploratory Undersampling for Class Imbalance Learning. *IEEE Transactions on Systems, Man and Cybernetics*, v. 39, n. 2, p. 539–550, 2009. Cited on page 71.

LÖFSTRÖM, T.; JOHANSSON, U.; BOSTRÖM, H. Ensemble member selection using multi-objective optimization. In: *IEEE Symposium on Computational Intelligence and Data Mining*. [S.l.: s.n.], 2009. p. 245–251. Cited 2 times on pages 66 and 67.

MADJAROV, G.; KOCEV, D.; GJORGJEVIKJ, D.; DŽEROSKI, S. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, v. 45, n. 9, p. 3084–3104, 2012. Cited on page 107.

MAO, W.; TIAN, M.; CAO, X.; XU, J. Model selection of extreme learning machine based on multi-objective optimization. *Neural Computing and Applications*, Springer-Verlag, v. 22, n. 3-4, p. 521–529, mar 2013. Cited on page 65.

MARATEA, A.; PETROSINO, A.; MANZO, M. Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, Elsevier Inc., v. 257, p. 331–341, 2014. Cited on page 71.

MARLER, R. T.; ARORA, J. S. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, v. 26, n. 6, p. 369–395, 2004. Cited 2 times on pages 31 and 32.

MEIER, L.; Van De Geer, S.; BÜHLMANN, P. The group lasso for logistic regression. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, v. 70, n. 1, p. 53–71, 2008. Cited on page 54.

MEMISEVIC, R.; SIGAL, L.; FLEET, D. J.; MEMBER, S. for Discriminative Inference. *Pattern Analysis and Machine Intelligence*, v. 34, n. 4, p. 778–790, 2012. Cited on page 77.

MIETTINEN, K. *Nonlinear Multiobjective Optimization*. [S.l.]: Springer, 1999. Cited 3 times on pages 31, 32, and 34.

MIRANDA, P. B. C.; PRUDÊNCIO, R. B. C.; CARVALHO, A. C. P. L. F. de; SOARES, C. Combining a multi-objective optimization approach with meta-learning for SVM parameter selection. In: *IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.: s.n.], 2012. p. 2909–2914. Cited 2 times on pages 65 and 67.

MIRANDA, P. B. C.; PRUDÊNCIO, R. B. C.; CARVALHO, A. P. L. F.; SOARES, C. A hybrid meta-learning architecture for multi-objective optimization of SVM parameters. *Neurocomputing*, Elsevier, v. 143, p. 27–43, 2014. Cited 2 times on pages 65 and 67.

MITCHELL, T. M. *Machine learning*. [S.l.: s.n.], 1997. 1–414 p. Cited on page 22.

MUKHOPADHYAY, A. A.; MAULIK, U. U.; BANDYOPADHYAY, S. S. Multiobjective genetic clustering with ensemble among pareto front solutions: Application to MRI brain image segmentation. In: *International Conference on Advances in Pattern Recognition*. [S.l.: s.n.], 2009. p. 236–239. Cited on page 66.

MÜSSEL, C.; LAUSSER, L.; MAUCHER, M.; KESTLER, H. a. Multi-Objective Parameter Selection for Classifiers. *Journal of Statistical Software*, v. 46, n. 5, p. 1–27, 2012. Cited on page 65.

NAG, K.; PAL, N. R. A Multiobjective Genetic Programming-Based Ensemble for Simultaneous Feature Selection and Classification. *IEEE Transactions on Cybernetics*, Institute of Electrical and Electronics Engineers Inc., v. 46, n. 2, p. 499–510, feb 2016. Cited on page 66.

NETO, A. A. F.; CANUTO, A. M. P.; LUDERMIR, T. B. Using good and bad diversity measures in the design of ensemble systems: A genetic algorithm approach. In: *IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2013. p. 789–796. Cited on page 66.

NIGAM, K.; GHANI, R. Analyzing the effectiveness and applicability of co-training. *Proceedings of the ninth international conference on Information and knowledge management - CIKM '00*, p. 86–93, 2000. Cited on page 77.

OBOZINSKI, G.; WAINWRIGHT, M. J.; JORDAN, M. L. High-dimensional support union recovery in multivariate regression. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2008. p. 1217–1224. Cited 2 times on pages 74 and 113.

OJHA, V. K.; ABRAHAM, A.; SNÁŠEL, V. Ensemble of heterogeneous flexible neural trees using multiobjective genetic programming. *Applied Soft Computing Journal*, v. 52, p. 909–924, 2017. Cited on page 66.

OLIVEIRA, L. S.; MORITA, M.; SABOURIN, R.; BORTOLOZZI, F. Multi-objective genetic algorithms to create ensemble of classifiers. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2005. v. 3410, p. 592–606. Cited 2 times on pages 66 and 67.

OPITZ, D.; MACLIN, R. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, v. 11, p. 169–198, 1999. Cited on page 70.

ORTIGOSA-HERNÁNDEZ, J.; INZA, I.; LOZANO, J. A. Measuring the class-imbalance extent of multi-class problems. *Pattern Recognition Letters*, v. 98, p. 32–38, 2017. Cited 4 times on pages , 91, 92, and 93.

PAN, S. J.; YANG, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 10, p. 1345–1359, oct 2010. Cited on page 76.

PILAT, M.; NERUDA, R. Multiobjectivization for classifier parameter tuning. In: *Conference companion on Genetic and evolutionary computation conference companion*. [S.l.: s.n.], 2013. p. 97. Cited 2 times on pages 65 and 66.

QIANG, Y.; MUNRO, P. W. Improving a Neural Network Classifier Ensemble with Multi-Task Learning. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. [S.l.]: IEEE, 2006. p. 5164–5170. Cited on page 75.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Cited on page 75.

RAIMUNDO, M. M.; DRUMMOND, T. F.; MARQUES, A. C. R.; LYRA, C.; ROCHA, A.; VON ZUBEN, F. J. Exploring multiobjective training in multiclass classification. Manuscript submitted for publication. *IEEE Transactions on Neural Networks and Learning Systems*, 2018. Cited 7 times on pages 31, 57, 61, 65, 68, 69, and 78.

RAIMUNDO, M. M.; VON ZUBEN, F. J. MONISE - Many Objective Non-Inferior Set Estimation. *arXiv*, v. 1709.00797, p. 1–39, 2017. Cited 8 times on pages 27, 28, 31, 32, 43, 44, 67, and 76.

RAIMUNDO, M. M.; VON ZUBEN, F. J. Investigating multiobjective methods in multitask classification. In: *International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. Cited 5 times on pages , 59, 61, 73, and 109.

RAIMUNDO, M. M.; VON ZUBEN, F. J. Many-Objective Ensemble-Based Multilabel Classification. In: MENDOZA, M.; VELASTÍN, S. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Cham: Springer International Publishing, 2018. p. 365–373. ISBN 978-3-319-75193-1. Cited 3 times on pages 58, 72, and 104.

RAMÍREZ-CORONA, M.; SUCAR, L. E.; MORALES, E. F. Hierarchical multilabel classification based on path evaluation. *International Journal of Approximate Reasoning*, Elsevier Inc., v. 68, p. 179–193, 2016. Cited on page 72.

RAMÓN QUEVEDO, J.; LUACES, O.; BAHAMONDE, A. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, v. 45, n. 2, p. 876–883, 2012. Cited on page 73.

READ, J.; PFAHRINGER, B.; HOLMES, G. Multi-label classification using ensembles of pruned sets. *Proceedings - IEEE International Conference on Data Mining, ICDM*, p. 995–1000, 2008. Cited 2 times on pages 72 and 73.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. *Machine Learning*, v. 85, n. 3, p. 333–359, 2011. Cited 3 times on pages 72, 73, and 106.

ROKACH, L. Ensemble-based classifiers. *Artificial Intelligence Review*, v. 33, n. 1-2, p. 1–39, 2010. Cited 3 times on pages 101, 106, and 112.

ROMERO, C.; REHMAN, T. Multiobjective programming. In: ROMERO, C.; REHMAN, T. (Ed.). *Multiple Criteria Analysis for Agricultural Decisions*. [S.l.]: Elsevier, 2003, (Developments in Agricultural Economics, v. 11). cap. 4, p. 47–61. Cited on page 36.

ROSALES-PÉREZ, A.; GONZALEZ, J. A.; Coello Coello, C. A.; ESCALANTE, H. J.; REYES-GARCIA, C. A. Surrogate-assisted multi-objective model selection for support vector machines. *Neurocomputing*, v. 150, p. 163–172, 2015. Cited 2 times on pages 65 and 66.

SÁEZ, J. A.; LUENGO, J.; STEFANOWSKI, J.; HERRERA, F. SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling

method with filtering. *Information Sciences*, v. 291, n. C, p. 184–203, 2015. Cited on page 90.

SAHA, S.; MITRA, S.; YADAV, R. K. A multiobjective based automatic framework for classifying cancer-microRNA biomarkers. *Gene Reports*, Elsevier B.V., v. 4, p. 91–103, 2016. Cited on page 66.

SATAPATHY, S. C.; GOVARDHAN, A.; RAJU, K. S.; MANDAL, J. K. Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 2. *Advances in Intelligent Systems and Computing*, v. 338, p. I–IV, 2015. Cited on page 73.

SCHAPIRE, R. E. Measures of Diversity in Classifier Ensembles. *Machine Learning*, v. 51, n. 2, p. 181–207, 2003. Cited on page 62.

SCHAPIRE, R. E. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, v. 14, n. 5, p. 771–780, 2009. Cited on page 70.

SEIFFERT, C.; KHOSHGOFTAAR, T. M.; Van Hulse, J.; NAPOLITANO, A. RUSBoost: Improving classification performance when training data is skewed. *2008 19th International Conference on Pattern Recognition*, p. 1–4, 2008. Cited on page 71.

SHI, C.; KONG, X.; FU, D.; YU, P. S.; WU, B. Multi-Label Classification Based on Multi-Objective Optimization. *ACM Transactions on Intelligent Systems and Technology*, v. 5, n. 2, p. 1–22, 2014. Cited on page 73.

SHI, C.; KONG, X.; YU, P.; WANG, B. Multi-label ensemble learning. *Lecture Notes in Computer Science*, v. 6913 LNAI, n. PART 3, p. 223–239, 2011. Cited on page 73.

SHI, C.; KONG, X.; YU, P. S.; WANG, B. Multi-objective multi-label classification. In: *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*. [S.l.: s.n.], 2012. p. 355–366. Cited on page 73.

SHOEB, A. H. *Application of machine learning to epileptic seizure onset detection and treatment*. 157–162 p. Tese (Doutorado) — Harvard-MIT Division of Health Sciences and Technology, 2009. Cited 2 times on pages 94 and 95.

SHOEB, A. H.; GUTTAG, J. V. Application of Machine Learning To Epileptic Seizure Detection. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, p. 975–982, 2010. Cited on page 94.

SHON, A.; GROCHOW, K.; HERTZMANN, A.; RAO, R. Learning shared latent structure for image synthesis and robotic imitation. *Adv. Neural Inf. Process. Syst.*, v. 18, p. 1233, 2006. Cited on page 77.

SIMM, J.; MAGRANS, I.; ABRIL, D. E.; SUGIYAMA, M. Tree-Based Ensemble Multi-Task Learning Method for Classification and Regression. In: *IEICE Transactions on Information and Systems*. [S.l.: s.n.], 2014. p. 1677–1681. Cited on page 75.

SKILLINGS, J. H. On the Use of a Friedman-Type in Balanced Statistic Block Designs and Unbalanced. *Technometrics*, v. 23, n. 2, p. 171–177, 1981. Cited on page 87.

SMITH, C.; JIN, Y. Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. *Neurocomputing*, v. 143, p. 302–311, 2014. Cited on page 66.

SUN, Y.; WONG, A.; WANG, Y. Parameter inference of cost-sensitive boosting algorithms. *Machine Learning and Data Mining in Pattern Recognition*, v. 3587, n. July, p. 21–30, 2005. Cited on page 71.

SUN, Y.; WONG, A. K. C.; KAMEL, M. S. CLASSIFICATION OF IMBALANCED DATA: A REVIEW. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 23, n. 04, p. 687–719, 2009. Cited on page 70.

TAHIR, M. A.; KITTLER, J.; BOURIDANE, A. Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recognition Letters*, Elsevier B.V., v. 33, n. 5, p. 513–523, 2012. Cited on page 73.

TANG, L.; RAJAN, S.; NARAYANAN, V. K. Large scale multi-label classification via metalabeler. *Proceedings of the 18th International Conference on World Wide Web*, p. 211, 2009. Cited on page 73.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, v. 23, n. 7, p. 1079–1089, 2011. Cited 3 times on pages 72, 73, and 106.

Van Esbroeck, A.; SMITH, L.; SYED, Z.; SINGH, S.; KARAM, Z. Multi-task seizure detection: addressing intra-patient variation in seizure morphologies. *Machine Learning*, Springer US, v. 102, n. 3, p. 309–321, 2016. Cited on page 96.

WAINBERG, M.; ALIPANAHI, B.; FREY, B. J. Are Random Forests Truly the Best Classifiers? *Journal of Machine Learning Research*, v. 17, n. 110, p. 1–5, 2016. Cited 4 times on pages 79, 87, 88, and 89.

WANG, Q.; ZHANG, L. Ensemble learning based on multi-task class labels. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6119, p. 464–475, 2010. Cited on page 75.

WANG, S.; WANG, J.; WANG, Z.; JI, Q. Enhancing multi-label classification by modeling dependencies among labels. *Pattern Recognition*, Elsevier, v. 47, n. 10, p. 3405–3413, 2014. Cited 2 times on pages 66 and 74.

WANG, W.; ZHOU, Z.-H. Analyzing co-training style algorithms. *ECML 07 Proceedings of the 18th European conference on Machine Learning*, p. 454–465, 2007. Cited on page 77.

WEIHS, C.; LUEBKE, K.; CZOGIEL, I. Response Surface Methodology for Optimizing Hyper Parameters. 2005. Cited on page 69.

WIECEK, M. M.; EHRGOTT, M.; ENGAU, A. Continuous Multiobjective Programming. In: *Multiple Criteria Decision Analysis*. [S.l.: s.n.], 2016. p. 739–814. Cited 2 times on pages 31 and 32.

XIA, T.; TAO, D.; MEI, T.; ZHANG, Y. Multiview spectral embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, v. 40, n. 6, p. 1438–1446, 2010. Cited on page 77.

XU, C.; TAO, D.; XU, C. A Survey on Multi-view Learning. p. 1–59, 2013. Cited 2 times on pages 76 and 77.

YANG, P.; ZHANG, X.-Y.; HUANG, K.; LIU, C.-L. Manifold Regularized Multi-Task Learning. In: *IEEE Transactions on Image Processing*. [S.l.: s.n.], 2012. p. 528—-536. Cited on page 75.

YIN, J.; TAO, T.; XU, J. A Multi-label Feature Selection Algorithm Based on Multi-objective Optimization. 2015. Cited on page 73.

YUAN, M.; LIN, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, v. 68, n. 1, p. 49–67, 2006. Cited 2 times on pages 54 and 59.

ZHANG, G. G. b.; YIN, J.; ZHANG, S.; CHENG, L. L. Regularization based ordering for ensemble pruning. In: *International Conference on Fuzzy Systems and Knowledge Discovery*. [S.l.: s.n.], 2011. v. 2, p. 1325–1329. Cited on page 66.

ZHANG, M. L.; ZHOU, Z. H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, v. 40, n. 7, p. 2038–2048, 2007. Cited on page 72.

ZHANG, Y.; YEUNG, D.-y. A Convex Formulation for Learning Task Relationships in Multi-Task Learning. In: *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence*. [S.l.: s.n.], 2010. p. 733–442. Cited on page 75.

ZHANG, Y.; YEUNG, D.-Y. A Regularization Approach to Learning Task Relationships in Multitask Learning. *ACM Transactions on Knowledge Discovery from Data*, ACM, v. 8, n. 3, p. 1–31, jun 2014. Cited on page 75.

ZHENG, A. X.; BILENKO, M. Lazy Paired Hyper-Parameter Tuning. *International joint conference on Artificial Intelligence*, p. 1924–1931, 2013. Cited on page 69.

ZHONG, L. W.; KWOK, J. T.; HK, J. U. Convex multitask learning with flexible task clusters. In: *Proceedings of the 29th International Conference on Machine Learning*. [S.l.: s.n.], 2012. p. 49–56. Cited 3 times on pages 75, 110, and 111.

ZHONG, S.; PU, J.; JIANG, Y.-G.; FENG, R.; XUE, X. Flexible multi-task learning with latent task grouping. *Neurocomputing*, v. 189, n. Supplement C, p. 179–188, 2016. Cited on page 75.

ZHOU, J.; YUAN, L.; LIU, J.; YE, J. A multi-task learning formulation for predicting disease progression. In: *International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2011. p. 814. Cited 3 times on pages 74, 75, and 113.

ZHOU, Q.; ZHAO, Q. Flexible Clustered Multi-Task Learning by Learning Representative Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 2, p. 266–278, 2016. Cited on page 75.

ZHOU, Z. *Ensemble Methods: Foundations and Algorithms*. [S.l.]: Chapman & Hall, 2012. (CRC Machine Learning & Pattern Recognition). Cited 5 times on pages 61, 62, 63, 65, and 70.