Noise detection in classification problems

Luís Paulo Faina Garcia

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Luís Paulo Faina Garcia

Noise detection in classification problems

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

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

USP – São Carlos
August 2016

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

G216n Garcia, Luís Paulo Faina
 Noise detection in classification problems / Luís
 Paulo Faina Garcia; orientador André Carlos Ponce de
 Leon Ferreira de Carvalho. - São Carlos - SP, 2016.
 108 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional)
 - Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2016.

 1. Aprendizado de Máquina. 2. Problemas
 de Classificação. 3. Detecção de Ruídos.
 4. Meta-aprendizado. I. Carvalho, André Carlos Ponce
 de Leon Ferreira de, orient. II. Título.

Luís Paulo Faina Garcia

Detecção de ruídos em problemas de classificação

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

Área de Concentração: Ciências de Computação e Matemática Computacional

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

USP – São Carlos
Agosto de 2016

*There are things known and there
are things unknown, and in between
are the doors of perception.*

Aldous Huxley

Acknowledgements

Firstly, I would like to express my deep gratitude to Prof. André de Carvalho and Ana Lorena, my research supervisors. Prof. André de Carvalho is one of the few fascinating people who we have the pleasure to meet in life. An exceptional professional and a humble human being. Prof. Ana Lorena is responsible for one of the most important achievements of my life, which was the finishing of this work. She enlightened every step of this journey with her personal and professional advices. I thank both for granting me the opportunity to grow as a researcher.

Besides my advisors, I would like to thank Francisco Herrera and Stan Matwin for sharing their valuable knowledge and advice during the internships. I am also thankful to Prof. João Rosa, Prof. Rodrigo Mello and Prof. Gustavo Batista for being my professors in the first half of the doctorate. With them I had the pleasure to learn the meaning of being a good professor.

I thank my friends and labmates who supported me in so many different ways. To Jáder Breda, Carlos Breda, Luiz Trondoli e Alexandre Vaz for being my brothers since 2005 and expend so many coffee with me. To Davi Santos for the opportunity to know a bit of your thoughts. To Henrique Marques for all kilometers running and all breathless talks. To André Rossi, Daniel Cestari, Everlândio Fernandes, Victor Barella, Adriano Rivolli, Kemilly Garcia, Murilo Batista, Fernando Cavalcante, Fausto Costa, Victor Padilha e Luiz Coletta for the moments in the Biocom, talking, discussing and laughing.

My gratitude also goes to my girlfriend Thalita Liporini, for all her love and support. You made the happy moments much more sweet. I also would like to thank my parents Prof. Paulo Garcia and Tânia Maria and my sisters Gabriella Garcia and Laleska Garcia. You are my huge treasure. This work is yours.

Finally, I would like to thank FAPESP for the financial support which made possible the development of this work (process 2011/14602 – 7).

Abstract

Garcia, L. P. F.. Noise detection in classification problems. 2016. 108 f. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP.

In many areas of knowledge, considerable amounts of time have been spent to comprehend and to treat noisy data, one of the most common problems regarding information collection, transmission and storage. These noisy data, when used for training Machine Learning techniques, lead to increased complexity in the induced classification models, higher processing time and reduced predictive power. Treating them in a preprocessing step may improve the data quality and the comprehension of the problem. This Thesis aims to investigate the use of data complexity measures capable to characterize the presence of noise in datasets, to develop new efficient noise filtering techniques in such subsamples of problems of noise identification compared to the state of art and to recommend the most properly suited techniques or ensembles for a specific dataset by meta-learning. Both artificial and real problem datasets were used in the experimental part of this work. They were obtained from public data repositories and a cooperation project. The evaluation was made through the analysis of the effect of artificially generated noise and also by the feedback of a domain expert. The reported experimental results show that the investigated proposals are promising.

Key-words: Machine Learning, Classification Problems, Noise Detection, Meta-learning.

Resumo

Garcia, L. P. F.. Noise detection in classification problems. 2016. 108 f. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP.

Em diversas áreas do conhecimento, um tempo considerável tem sido gasto na compreensão e tratamento de dados ruidosos. Trata-se de uma ocorrência comum quando nos referimos à coleta, à transmissão e ao armazenamento de informações. Esses dados ruidosos, quando utilizados na indução de classificadores por técnicas de Aprendizado de Máquina, aumentam a complexidade da hipótese obtida, bem como o aumento do seu tempo de indução, além de prejudicar sua acurácia preditiva. Tratá-los na etapa de pré-processamento pode significar uma melhora da qualidade dos dados e um aumento na compreensão do problema estudado. Esta Tese investiga medidas de complexidade capazes de caracterizar a presença de ruídos em um conjunto de dados, desenvolve novos filtros que sejam mais eficientes em determinados nichos do problema de detecção e remoção de ruídos que as técnicas consideradas estado da arte e recomenda as mais apropriadas técnicas ou comitês de técnicas para um determinado conjunto de dados por meio de meta-aprendizado. As bases de dados utilizadas nos experimentos realizados neste trabalho são tanto artificiais quanto reais, coletadas de repositórios públicos e fornecidas por projetos de cooperação. A avaliação consiste tanto da adição de ruídos artificiais quanto da validação de um especialista. Experimentos realizados mostraram o potencial das propostas investigadas.

Palavras-chave: Aprendizado de Máquina, Problemas de Classificação, Detecção de Ruídos, Meta-aprendizado.

Contents

Contents	xv
List of Figures	xix
List of Tables	xxi
List of Algorithms	xxiii
List of Abbreviations	xxv
1 Introduction	1
1.1 Motivations	2
1.2 Objectives and Proposals	4
1.3 Hypothesis	6
1.4 Outline	7
2 Noise in Classification Problems	9
2.1 Types of Noise	11
2.2 Describing Noisy Datasets: Complexity Measures	12
2.2.1 Measures of Overlapping in Feature Values	14
2.2.2 Measures of Class Separability	16
2.2.3 Measures of Geometry and Topology	17
2.2.4 Measures of Structural Representation	18
2.2.5 Summary of Measures	22
2.3 Evaluating the Complexity of Noisy Datasets	23
2.3.1 Datasets	23
2.3.2 Methodology	24
2.4 Results obtained in the Correlation Analysis	27
2.4.1 Correlation of Measures with the Noise Level	28
2.4.2 Correlation of Measures with the Predictive Performance	29
2.4.3 Correlation Between Measures	30
2.5 Chapter Remarks	32

3	Noise Identification	33
3.1	Noise Filters	35
3.1.1	Ensemble Based Noise Filters	35
3.1.2	Noise Filters Based on Data Descriptors	37
3.1.3	Distance Based Noise Filters	40
3.1.4	Other Noise Filters	42
3.2	Noise Filters: a Soft Decision	43
3.3	Evaluation Measures for Noise Filters	44
3.4	Evaluating the Noise Filters	47
3.4.1	Datasets	47
3.4.2	Methodology	47
3.5	Experimental Evaluation of Crisp Filters	49
3.5.1	Rank analysis	49
3.5.2	F_1 per noise level	50
3.6	Experimental Evaluation of Soft Filters	54
3.6.1	Similarity and Rank analysis	54
3.6.2	$p@n$ per noise level	56
3.6.3	NR-AUC per noise level	61
3.7	Chapter Remarks	66
4	Meta-learning	67
4.1	Modelling the Algorithm Selection Problem	69
4.1.1	Instance Features	70
4.1.2	Problem Instances	71
4.1.3	Algorithms	72
4.1.4	Evaluation measures	73
4.1.5	Learning using the meta-dataset	73
4.2	Evaluating MTL for NF prediction	74
4.2.1	Datasets	75
4.2.2	Methodology	75
4.3	Experimental Evaluation to Predict the Filter Performance	76
4.3.1	Experimental Analysis of the Meta-dataset	76
4.3.2	Performance of the Meta-regressors	77
4.4	Experimental Evaluation of the Filter Recommendation	81
4.4.1	Experimental analysis of the meta-dataset	81
4.4.2	Performance of the Meta-classifiers	82
4.5	Case Study: Ecology Data	85
4.5.1	Ecological Dataset	86
4.5.2	Filtering Recommendation	87

4.5.3	Experimental Results	87
4.6	Chapter Remarks	88
5	Conclusion	91
5.1	Main Contributions	92
5.2	Limitations	93
5.3	Prospective work	95
5.4	Publications	96
	References	99

List of Figures

2.1	Types of noise in classification problems.	11
2.2	Building a graph using ϵ -Nearest Neighbor (NN)	19
2.3	Flowchart of the experiments.	26
2.4	Histogram of each measure for distinct noise levels.	28
2.5	Correlation of each measure to the noise levels.	29
2.6	Correlation of each measure to the predictive performance of classifiers. . .	30
2.7	Heatmap of correlation between measures.	31
3.1	Building the graph for an artificial dataset.	39
3.2	Noise detection by GNN filter.	41
3.3	Example of NR-AUC calculation	46
3.4	Ranking of crisp NF techniques according to F_1 performance.	49
3.5	F_1 values of the crisp NF techniques per dataset and noise level.	51
3.6	F_1 values of the crisp NF techniques per dataset and noise level.	52
3.7	Ranking of crisp NF techniques according to F_1 performance per noise level.	53
3.8	Ranking of soft NF techniques according to $p@n$ performance.	55
3.9	Dissimilarity of filters predictions.	56
3.10	$p@n$ values of the best soft NF techniques per dataset and noise level. . . .	57
3.11	$p@n$ values of the best soft NF techniques per dataset and noise level. . . .	58
3.12	Ranking of best soft NF techniques according to $p@n$ performance per noise level.	60
3.13	NR-AUC values of the best soft NF techniques per dataset and noise level.	62
3.14	NR-AUC values of the best soft NF techniques per dataset and noise level.	63
3.15	Ranking of best soft NF techniques according to NR-AUC performance per noise level.	64
4.1	Smith-Miles (2008) algorithm selection diagram. (Adapted from Smith- Miles (2008))	70
4.2	Performance of the six crisp NF techniques.	78
4.3	MSE of each meta-regressor for each NF technique in the meta-dataset. . .	79
4.4	Performance of the six crisp NF techniques.	80
4.5	Frequency with which each meta-feature was selected by CFS technique. .	81

4.6	Distribution of highest $p@n$	82
4.7	Accuracy of each meta-classifier in the meta-dataset.	83
4.8	Performance of meta-models in the base-level.	83
4.9	Meta DT model for NF recommendation.	85
5.1	IR achieved by the best crisp NF techniques in datasets with the higher IR.	94
5.2	Increase of performance by the Best meta-regressor in the base-level when using DF as baseline.	96

List of Tables

2.1	Summary of Measures.	22
2.2	Summary of datasets characteristics: name, number of examples, number of features, number of classes and the percentage of the majority class. . .	25
3.1	Confusion matrix for noise detection.	45
3.2	Possible ensembles of NF techniques considered in this work	48
3.3	Percentage of best performance for each noise level.	61
4.1	Summary of the characterization measures.	72
4.2	Summary the predictive features of the <i>species</i> dataset.	86

List of Algorithms

1	SEF	36
2	Selecting m classifiers to compose the DEF ensemble	37
3	Saturation Test	38
4	Saturation Filter	38
5	AENN	42

List of Abbreviations

AENN	<i>All-k</i> -Nearest Neighbor
ANN	Artificial Neural Network
AUC	Area Under the ROC Curve
CFS	Correlation-based Feature Selection
CLCH	Complexity of the Least Correct Hypothesis
CVCF	Cross-validated Committees Filter
DCoL	Data Complexity Library
DEF	Dynamic Ensemble Filter
DF	Default Technique
DM	Data Mining
DT	Decision Tree
DWNN	Distance-weighted <i>k</i> -NN
ENN	Edited Nearest Neighbor
GNN	Graph Nearest Neighbor
HARF	High Agreement Random Forest Filter
INFFC	Iterative Noise Filter based on the Fusion of Classifiers
IPF	Iterative-Partitioning Filter
IR	Imbalance Ratio
ML	Machine Learning
MSE	Mean Squared Error

MST	Minimum Spanning Tree
MTL	Meta-learning
NB	Naive Bayes
NDP	Noisy Degree Prediction
NF	Noise Filtering
NN	Nearest Neighbor
NR-AUC	Noise Ranking Area Under the ROC Curve
RENN	Repeated Edited Nearest Neighbor
RD	Random Technique
RF	Random Forest
SEF	Static Ensemble Filter
SF	Saturation Filter
ST	Saturation Test
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine

Chapter 1

Introduction

This Thesis investigates new alternatives for the use of Noise Filtering (NF) techniques to improve the predictive performance of classification models induced by Machine Learning (ML) algorithms.

Classification models are induced by supervised ML techniques when these techniques are applied to a labeled dataset. This Thesis will assume that a labeled dataset is composed by n pairs $(\mathbf{x}_i, y_i)$, where each $\mathbf{x}_i$ is a tuple of predictive features describing a certain object and y_i is target feature, which value corresponds to the object class. The predictive performance of the induced model for new data depends on various factors, such as the training data quality and the inductive bias of the ML algorithm. Nonetheless, despite of the algorithm bias, when data quality is low, the performance of the predictive model is harmed.

In real world applications, there are many inconsistencies that affect data quality, such as missing data or unknown values, noise and faults in the data acquisition process (Wang et al., 1995; Fayyad et al., 1996). Data acquisition is inherently leaned to errors, even though extreme efforts are made to avoid them. It is also a resource-consuming step, since at least 60% of the efforts in a Data Mining (DM) task is spent on data preparation, which includes data preprocessing and data transformation (Pyle, 1999). Some studies estimate that, even in controlled environments, there are at least 5% of errors in a dataset (Wu, 1995; Maletic & Marcus, 2000).

Although many ML techniques have internal mechanisms to deal with noise, such as the pruning mechanism in Decision Trees (DTs) (Quinlan, 1986b,a), the presence of noise in data may lead to difficulties in the induction of ML models. These difficulties include an increase in processing time, a higher complexity of the induced model and a possible deterioration of its predictive performance for new data (Lorena & de Carvalho, 2004). When these models are used in critical environments, they may also have security and reliability problems (Strong et al., 1997).

To reduce the data modeling problems due to the presence of noise, the two usual approaches are: to employ a noise-tolerant classifier (Smith et al., 2014); or, to adopt

a preprocessing step, also known as data cleansing (Zhu & Wu, 2004) to identify and remove noisy data. The use of noise-tolerant classifiers aims to construct robust models by using some information related to the presence of noise. The preprocessing step, on the other hand, normally involves the application of one or more NF techniques to identify the noisy data. Afterwards, the identified inconsistencies can be corrected or, more often, eliminated (Gamberger et al., 2000). The research carried out in this Thesis follows the second approach.

Even using more than one NF technique, each with a different bias, it is usually not possible to guarantee whether a given example is really a noisy example without the support of a data domain expert (Wu & Zhu, 2008; Sáez et al., 2013). Just filtering out potentially noisy data can also remove correct examples containing valuable information, which could be useful for the learning process. Thus, an extraction of noisy patterns might be needed to perform a proper filtering process. It could be done through the use of characterization measures, leading to the recommendation of the best NF using Meta-learning (MTL) for a new dataset and improves the noise detection accuracy.

The study presented in this Thesis investigates how noise affects the complexity of classification datasets identifying problem characteristics that are more sensitive to the presence of noise. This work also seeks to improve the robustness in noise detection and to recommend the best NF technique for the identification of potential noisy examples in new datasets with support of MTL. The validation of the filtering process in a real dataset is also investigated.

This chapter is structured as follows. Section 1.1 presents the main problems and gaps related to noise detection in classification tasks. Section 1.2 presents the objectives of this work and Section 1.3 defines the hypothesis investigated in this research. Finally, Section 1.4 presents the outline of this Thesis.

1.1 Motivations

The manual search for inconsistencies in a dataset by an expert is usually an unfeasible task. In the 1990s, some organizations, which used information collected from dynamic environments, spent annually, millions of dollars on training, standardization and error detection tools (Redman, 1997). In the last decades, even with the automation of the collecting processes, this cost has increased, as a consequence of the growing use of data monitoring tools (Shearer, 2000). As a result, there was an increase in data cleansing costs to avoid security and reliability problems (Strong et al., 1997).

Data cleansing processes provide techniques to automatically treat data inconsistencies. Some of them are general (Wang et al., 1995; Redman, 1998; Maletic & Marcus, 2000; Shanab et al., 2012), while other techniques target specific issues, such as:

- missing values (Batista & Monard, 2003);
- outlier detection (Hodge & Austin, 2004);
- imbalanced data (Hulse et al., 2011; López et al., 2013);
- noise detection (Brodley & Friedl, 1999; Verbaeten & Assche, 2003).

The noise detection is a critical component of the preprocessing step. The techniques which deal with noise in a preprocessing step are known as Noise Filtering (NF) techniques (Zhu et al., 2003). The noise detection literature commonly divides noise detection in two main approaches: noise detection in the predictive features and noise detection in the target feature.

The presence of noise is more common in the predictive features than in the target feature. Predictive feature noise is found in large quantities in many real problems (Teng, 1999; Yang et al., 2004; Hulse et al., 2007; Sahu et al., 2014). An alternative to deal with the predictive noise is the elimination of the examples where noise was detected. However, the elimination of examples with noise in predictive features could cause more harm than good (Zhu & Wu, 2004), since other predictive features from these examples may be useful to build the classifier.

Noise in the target feature is usually investigated in classification tasks, where the noise changes the true class label to another class label. A common approach to overcome the problems due to the presence of noise in the target feature is the use of NF techniques which remove potentially noisy examples. Most of the existing NF techniques focus on the elimination of examples with class label. Such approach has been shown to be advantageous (Miranda et al., 2009; Sluban et al., 2010; Garcia et al., 2012; Sáez et al., 2013; Sluban et al., 2014). Noise in the class label, from now on named class noise, can be treated as an incorrect class label value.

Several studies show that the use of these techniques can improve the classification performance and reduce the complexity of the induced predictive models (Brodley & Friedl, 1999; Sluban et al., 2014; Garcia et al., 2012; Sáez et al., 2016). NF techniques can rely on different types of information to detect noise, such as those employing neighborhood or density information (Wilson, 1972; Tomek, 1976; Garcia et al., 2015), descriptors extracted from the data (Gamberger et al., 1999; Sluban et al., 2014) and noise identification models induced by classifiers (Sluban et al., 2014) or ensembles of classifiers (Brodley & Friedl, 1999; Verbaeten & Assche, 2003; Sluban et al., 2010; Garcia et al., 2012). Since each NF has a bias, they can present a distinct predictive performance for different datasets (Wu & Zhu, 2008; Sáez et al., 2013). Consequently, the proper management of NF bias is expected to lead to an improvement on the noise detection accuracy.

Despite the technique employed to deal with noise, it is important to understand the effect of noise in the classification task. Characterization measures extracted from a

classification dataset can be used to detect the presence or absence of noise in the dataset. These measures can be used to assess the complexity of the classification task (Ho & Basu, 2002; Orriols-Puig et al., 2010; Kolaczyk, 2009). For such, they take into account the overlap between classes imposed by feature values, the separability and distribution of the data points and the value of structural measures based on the representation of the dataset as a graph structure. Accordingly, experimental results show that the addition of noise in a dataset affects the geometry of the classes separation, which can be captured by these measures (Sáez et al., 2013).

Another open research issue is the definition of how suitable a NF technique is for each dataset. MTL has been largely used in the last years to support the recommendation of the most suitable ML algorithm(s) for a new dataset (Brazdil et al., 2009). Given a set of widely used NF techniques and a set of complexity measures able to characterize datasets, an automatic system could be employed to support the choice of the most suitable NF technique by non-experts. In this Thesis, we investigate the support provided by the proposed MTL-based recommendation system. The experiments were based on a meta-dataset consisting of complexity measures extracted from a collection of several artificially corrupted datasets along with information about the performance of widely used NF techniques.

1.2 Objectives and Proposals

The main goal of this study is the investigation of class label noise detection in a preprocessing step, providing new approaches able to improve the noise detection predictive performance. The proposed approaches include the study of the use of complexity measures to identify noisy patterns, the development of new techniques to fill gaps in existing techniques regarding predictive performance in noise detection and the use of MTL to recommend the most suitable NF technique(s). Another contribution of this study is the validation of the proposed approaches on a real dataset with an application domain expert.

The complexity measures were initially proposed in Ho & Basu (2002) to understand the complications associated to the induction of classification models from datasets. These measures extract characteristics related to the overlapping in the feature values, class separability and geometry and topology of the data. These characteristics can be associated with inconsistencies or presence of noisy data, justifying investigations involving their use in noise detection. This research also proposes the use of complexity structural measures, captured by representing the dataset through a graph structure (Kolaczyk, 2009). These measures extract topological and structural properties from the graphs. The use of a subset of measures capable to characterize the presence or absence of noise in a dataset can improve noise detection and support the decision of whether a NF technique should

be applied whether a new dataset should be cleaned by a NF technique.

Even for the well-known NF techniques that use different types of information to detect noise, such as neighborhood or density information, descriptors extracted from the data and noise identification models induced by classifiers or ensembles of classifiers, there is usually a margin of improvement on the noise detection accuracy. Two NF techniques are proposed, one of them based on a subset of complexity measures capable to detect noisy patterns and the other based on a committee of classifiers - both can increase the robustness in the noise identification.

Most NF techniques adopt a crisp decision for noise identification, classifying each training example as either noisy or safe. Soft decision strategies, on the other hand, assign a Noisy Degree Prediction (NDP) to each example. In practice, this allows not only to identify, but also to rank the potential noisy cases, evidencing the most unreliable instances. These examples could then be further examined by a domain expert. The adaptation of the original NF techniques for soft decision and the aggregation of different individual techniques can improve noise detection accuracy. These issues are also investigated in this Thesis.

The bias of each NF technique influences its predictive performance on a particular dataset. Therefore, there is no single technique that can be considered the best for all domains or data distributions and choosing a particular filter for a new dataset is not straightforward. An alternative to deal with this problem is to have a model able to recommend the best NF technique(s) for a new dataset. MTL has been successfully used for the recommendation of the most suitable technique for each one of several tasks, like classification, clustering, time series analysis and optimization. Thus, MTL would be a promising approach to induce a model able to predict the performance and recommend the best NF techniques for a new dataset. Its use could reduce the uncertainty in the selection of NF technique(s) and improve the label noise identification.

The predictive accuracy of MTL depends on how a dataset is characterized by meta-features. Thus, the first step to use MTL is to create a meta-dataset, with one meta-example representing each dataset. In this meta-dataset, for each meta-example, the predictive features are the meta-features extracted from a dataset and the target feature is the technique(s) with the best performance in the dataset.

The set of meta-features used in this Thesis describes various characteristics for each dataset, including its expected complexity level (Ho & Basu, 2002). Examples in this meta-dataset are labeled with the performance achieved by the NF technique in the noise identification. ML techniques from different paradigms are applied to the meta-dataset to induce a meta-model, which is used in a recommendation system to predict the best NF technique(s) for a new dataset.

To validate the proposed approaches, the results of the cleansing in a real dataset from the ecological niche modeling domain by a NF technique recommended using MTL

is analyzed by a domain expert. The dataset used for this validation shows the presence or absence of species in georeferenced points. Both classes present label noise: the absence of the species can be a misclassification if the point analyzed does not represent the protected area or even the false presence if the point analyzed does not have environmental compatibility in a long-term window.

All experiments use a large set of artificial and public domain datasets like UCI¹ with different levels of artificial imputed noise (Lichman, 2013). The NF evaluation is performed by standard measures, which are able to quantify the quality of the preprocessed datasets. The quality is related to the noisy cases correctly identified among those examples identified as noisy by the filter and noisy cases correctly identified among the noisy cases present in the dataset.

1.3 Hypothesis

Considering the current limitations and the existence of margins for improvement in noise detection in classification datasets, this work investigated four main hypotheses aiming to make inferences about the impact of label noise in classification problems and the possibility to performing data cleansing effectively. The hypotheses are:

1. **The characterization of datasets by complexity and structural measures can help to better detect noisy patterns.** Noise presence may affect the complexity of the classification problem, making it more difficult. Thereby, monitoring several measures in the presence of different label noise levels can indicate the measures that are more sensitive to the presence of label noise, and can thereby be used to support noise identification. Geometric, statistical and structural measures are extracted to characterize the complexity of a classification dataset.
2. **New techniques can improve the state of the art in noise detection.** Even with a high number of NF techniques, there is no single technique that has satisfactory results for all different niches and different noise levels. Thus, new techniques for NF can be investigated. The proposed NF techniques are based on a subset of complexity measures able to detect noisy patterns and based on an ensemble of classifiers.
3. **Noise filters techniques can be adapted to provide a NDP, which can increase the data understanding and the noise detection accuracy.** In order to highlight the most unreliable instances to be further examined, the rank of the potential noisy cases can increase the data understanding and it even makes easier to combine multiple filters in ensembles. While the expert can use the rank

¹<https://archive.ics.uci.edu/ml/datasets.html>

of unreliable instances to understand the noisy patterns, the ensembles can combine the NF techniques to increase the noise detection accuracy for a larger number of datasets than the individual techniques used alone.

4. **A model induced using meta-learning can predict the performance or even recommend the best NF technique(s) for a new dataset.** The bias of each NF technique influences its predictive performance on a particular dataset. Therefore, there is no single technique that can be considered the best for all datasets. A MTL system able to predict the expected performance of NF techniques in noisy data identification tasks could recommend the most suitable NF technique(s) for a new dataset.

1.4 Outline

The remainder of this Thesis is organized as follows:

Chapter 2 presents an overview of noisy data and complexity measures that can be used to characterize the complexity of noisy classification datasets. Preliminary experiments are performed to analyse the measures and, based on the experimental results, a subset of measures is suggested as more sensitive to the addition of noise in a dataset.

Chapter 3 addresses the preprocessing step, describing the main NF techniques. This chapter also proposed two new NF, one of them based in the experimental results presented in the previous chapter and the other based on the use of an ensemble of classifiers. In this chapter the NF techniques are also adapted to rank the potential noisy cases to increase the data understanding. Experiments are performed to analyse the predictive performance of the NF techniques for different noise levels with different evaluation measures.

Chapter 4 focuses on MTL, explaining the main meta-features and the algorithm selection problem adopted in this research. Experiments using MTL for NF technique recommendation are carried out, to predict the NF technique predictive performance and to recommend the best NF technique. In this chapter, a validation of the recommendation system approach on a real dataset with support of a domain expert is also presented.

Finally, Chapter 5 summarizes the main observations extracted from the experimental results from the previous chapters. It also points out some limitations of this study, raising questions that could be further investigated and discuss prospective research on the topic of noise detection.

Chapter 2

Noise in Classification Problems

The characterization of a dataset by the amount of information present in the data is a difficult task (Hickey, 1996). In many cases, only an expert can analyze the dataset and provide an overview about the dispersion concepts and the quality of the information present in the data (Pyle, 1999). Dispersion concepts are those associated with the process of identifying, understanding and planning the information to be collected, while quality of the information is related with the addition of inconsistencies in the collection process. Since the analysis of dispersion concepts is very difficult, it is natural to consider only the aspects associated with inconsistencies.

These inconsistencies can be absent of information (missing or unknown values), noise or errors (Wang et al., 1995; Fayyad et al., 1996). Even with extreme efforts to avoid noise, it is very difficult to ensure a data acquisition process without errors. Whereas the noise data needs to be identified and treated, secure data must be preserved in the dataset (Sluban et al., 2014). The term secure data usually refers to instances that are the core of the knowledge necessary to build accurate learning models (Quinlan, 1986b). This study deals with the problem of identifying noise in labeled datasets.

Various strategies and techniques have been proposed in the literature to reduce the problems derived from the presence of noisy data (Tomek, 1976; Brodley & Friedl, 1996; Verbaeten & Assche, 2003; Sluban et al., 2010; Garcia et al., 2012; Sluban et al., 2014; Smith et al., 2014). Some recent proposals include designing classification techniques more tolerant and robust to noise, as surveyed in Frenay & Verleysen (2014). Generally, the data identified as noisy are first filtered and removed from the datasets. Nonetheless, it is usually difficult to determine if a given instance is indeed noisy or not.

Despite the strategy employed to deal with noisy data, either by data cleansing or by the design of noise-tolerant learning algorithms, it is important to understand the effects that the presence of noise in a dataset cause in classification tasks. The use of measures capable to characterize the presence or absence of noise in a dataset could assist the noise detection or even the decision of whether a new dataset needs to be cleaned by a NF technique. Complexity measures may play an important role in this issue. A

recent work that uses complexity measures in the NF scenario is Sáez et al. (2013). The authors employ these measures to predict whether a NF technique is effective for cleaning a dataset that will be used for the induction of k -NN classifiers.

The approach presented in Sáez et al. (2013) differs from the approach proposed in this Thesis in several aspects. One of the main differences is that, while the approach proposed by Sáez et al. (2013) is restricted to k -NN classifiers, the proposed approach investigates how noise affects the complexity of the decision border that separates the classes. For such, it employs a series of statistic and geometric measures originally described in Ho & Basu (2002). These measures evaluate the difficulty of a classification task of a given dataset by analyzing some characteristics of the dataset and the predictive performance of some simple classification models induced from this dataset. Furthermore, the proposed approach uses new measures able to represent a dataset through a graph structure, named here structural measures (Kolaczyk, 2009; Morais & Prati, 2013).

The studies presented in this Thesis allow a better understanding of the effects of noise in the predictive performance of predictive models in classification tasks. Besides, they allow the identification of problem characteristics that are more sensitive to the presence of noise and that can be further explored in the design of new noise handling techniques. To make the reading of this text more direct, from now on, this Thesis will refer to complexity of datasets associated with classification tasks as complexity of classification tasks.

The main contributions from this chapter can be summarized as:

- Proposal of a methodology for the empirical evaluation of the effects of different levels of label noise in the complexity of classification datasets;
- Analysis of the sensibility of various measures associated with the geometrical complexity of classification datasets to detect the presence of label noise;
- Proposal of new measures able to evaluate the structural complexity of a classification dataset;
- Highlight complexity measures that can be further explored in the proposal of new noise handling techniques.

This chapter is structured as follows. Section 2.1 presents an overview of noisy data. Section 2.2 describes the complexity measures employed in this study to characterize the complexity of noisy classification datasets. A subset of these same measures is employed in Chapters 3 and 4 to characterize noisy datasets. Section 2.3 presents the experimental methodology followed in this Thesis to evaluate the sensitivity of the complexity measures to label noise imputation, while Section 2.4 presents and discusses the experimental results obtained in this analysis. Finally, Section 2.5 concludes this chapter.

2.1 Types of Noise

Noisy data also can be regarded as objects that present inconsistencies in their predictive and/or target feature values (Quinlan, 1986a). For supervised learning datasets, Zhu & Wu (2004) distinguish two types of noise: (i) in the predictive features and (ii) in the target feature. Noise in predictive features is introduced in one or more predictive features as consequence of incorrect, absent or unknown values. On the other hand, noise in target features occurs in the class labels. They can be caused by errors or subjectivity in data labeling, as well as by the use of inadequate information in the labeling process. Lately, noise in predictive features can lead to a wrong labeling of the data points, since they can be moved to the wrong side of the decision border.

The artificial binary dataset shown in Figure 2.1 illustrates these cases. The original dataset has 2 classes ($\bullet$ and $\blacktriangle$) that are linearly separable. Figure 2.1(a) shows the same artificial dataset with two potential predictive noisy examples, while Figure 2.1(b) has two potential label noisy examples. Although the noise identification for this artificial dataset is rather simplistic, for instance when the degree of noise in the predictive features is lower, the noise detection capability can dramatically decrease.

Figure 2.1: Types of noise in classification problems.

According to Zhu & Wu (2004), the removal of examples with noise in the predictive features is not as useful as label noise identification, since the values of other predictive features from the same examples can be helpful in the classifier induction process. Therefore, most of the NF techniques focus on the elimination of examples with label noise, which has shown to be more advantageous (Gamberger et al., 1999). For this reason, this work will concentrate in the identification of noise in label features. Hereafter, the term

noise will refer to label noise.

Ideally, noise identification should involve a validation step, where the objects highlighted as noisy are confirmed as such, before they can be further processed. Since the most common approach is to eliminate noisy data, it is important to properly distinguish these data from the safe data. Safe data need to be preserved, once they have features that represent part of the knowledge necessary for the induction of an adequate model.

In a real application, evaluating whether a given example is noisy or not usually has to rely on the judgment of a domain specialist, which is not always available. Furthermore, the need to consult a specialist tends to increase the cost and duration of the preprocessing step. This problem is reduced when artificial datasets are used, or when simulated noise is added to a dataset in a controlled way. The systematic addition of noise simplifies the validation of the noise detection techniques and the study of noise influence in the learning process.

There are two main methods to add noise to the class feature: (i) random, in which each example has the same probability of having its label corrupted (exchanged by another label) (Teng, 1999); and (ii) pairwise, in which a percentage $x\%$ of the majority class examples have their labels modified to the same label of the second majority class (Zhu et al., 2003). Whatever the strategy employed to add noise to a dataset, it is necessary to corrupt the examples within a given rate. In most of the related studies, noise is added according to rates that range from 5% to 40%, with intervals of 5% (Zhu & Wu, 2004), although other papers opt for fixed rates (as 2%, 5% and 10%) (Sluban et al., 2014). Besides, due to its stochastic nature, this addition is normally repeated a number of times for each noise level.

2.2 Describing Noisy Datasets: Complexity Measures

Each noise-tolerant technique and cleansing filter has a distinct bias when dealing with noise. To better understand their particularities, it is important to know how noisy data affects a classification problem. According to Li & Abu-Mostafa (2006), noisy data tends to increase the complexity of the classification problem. Therefore, the identification and removal of noise can simplify the geometry of the separation border between the problem classes (Ho, 2008).

Singh (2003) recommends a technique that estimates the complexity of the classification problem using neighborhood information for the identification of outliers. Sáez et al. (2013) use measures able to characterize the complexity of the classification problem to predict when a NF technique can be effectively applied to a dataset. Smith et al. (2014) propose a measure to capture instance hardness, considering an instance as hard if it is misclassified by a diverse set of classification algorithms. The instance hardness measure proposed is afterwards included into the learning process in two ways. They first propose

a modification of the error function minimized during neural networks training, so that hard instances have a lower weight on the error function update. The second proposal is a NF technique that removes hard instances, which correspond to potential noisy data. All previous work confirm the effect of noise in the complexity of the classification problem.

This work evaluates deeply the effects of different noise levels in the complexity of the classification problems, by extracting different measures from the datasets and monitoring their sensitivity to noise imputation. According to Ho & Basu (2002), the difficulty of a classification problem can be attributed to three main aspects: the ambiguity among the classes, the complexity of the separation between the classes, and the data sparsity and dimensionality. Usually, there is a combination of these aspects. They propose a set of geometrical and statistical descriptors able to characterize the complexity of the classification problem associated with a dataset. Originally proposed for binary classification problems (Ho & Basu, 2002), some of these measures were later extended to multiclass classification in Mollineda et al. (2005); Lorena & de Souto (2015) and Orriols-Puig et al. (2010). For measures only suitable for binary classification problems, we first transform the multiclass problem into a set of binary classification subproblems by using the *one-vs-all* approach. The mean of the complexity values obtained in such subproblems is then used as an overall measure for the multiclass dataset.

The descriptors of Ho & Basu (2002) can be divided into three categories:

Measures of overlapping in the feature values. Assess the separability of the classes in a dataset according to its predictive features. The discriminant power of each feature reflects its ambiguity level compared to the other features.

Measures of class separability. Quantify the complexity of the decision boundaries separating the classes. They are usually based on linearity assumptions and on the distance between examples.

Measures of geometry and topology. They extract features from the local (geometry) and global (topology) structure of the data to describe the separation between classes and data distribution.

Additionally, a classification dataset can be characterized as a graph, allowing the extraction of some structural measures from the data. Modeling a classification dataset through a graph allows capturing additional topological and structural information from a dataset. In fact, graphs are powerful tools for representing the information of relations between data (Ganguly et al., 2009). Therefore, this work includes an additional class of complexity measures in the experiments related to noise understanding:

Measures of structural representation. They are extracted from a structural representation of the dataset using graphs, which are built taking into account the relationship among the examples.

The recent work of Smith et al. (2014) also proposes a new set of measures, which are intended to understand why some instances are hard to classify. Since this type of analysis is not within the scope of this thesis, these measures were not included in the experiments.

2.2.1 Measures of Overlapping in Feature Values

Fisher's discriminant ratio (F1): Selects the feature that best discriminates the classes. It can be calculated by Equation 2.1, for binary classification problems, and by Equation 2.2 for problems with more than two classes (C classes). In these equations, m is the number of input features and f_i is the i -th predictive feature.

$$F1 = \max_{i=1}^m \frac{(\mu_{c_1}^{f_i} - \mu_{c_2}^{f_i})^2}{(\sigma_{c_1}^{f_i})^2 + (\sigma_{c_2}^{f_i})^2} \quad (2.1)$$

$$F1 = \max_{i=1}^m \frac{\sum_{c_j=1}^C \sum_{c_k=c_j+1}^C p_{c_j} p_{c_k} (\mu_{c_j}^{f_i} - \mu_{c_k}^{f_i})^2}{\sum_{c_j=1}^C p_{c_j} \sigma_{c_j}^2} \quad (2.2)$$

For continuous features, $\mu_{c_j}^{f_i}$ and $(\sigma_{c_j}^{f_i})^2$ are, respectively, the average and standard deviation of the feature f_i within the class c_j . Nominal features are first mapped into numerical values and $\mu_{c_j}^{f_i}$ is their median value, while $(\sigma_{c_j}^{f_i})^2$ is the variance of a binomial distribution, as presented in Equation 2.3, where $p_{\mu_{c_j}^{f_i}}$ is the median frequency and n_{c_j} is the number of examples in the class c_j .

$$\sigma_{c_j}^{f_i} = \sqrt{p_{\mu_{c_j}^{f_i}}(1 - p_{\mu_{c_j}^{f_i}}) * n_{c_j}} \quad (2.3)$$

High values of F1 indicate that at least one of the features in the dataset is able to linearly separate data from different classes. Low values, on the other hand, do not indicate that the problem is non-linear, but that there is not an hyperplane orthogonal to one of the data axis that separates the classes.

Directional-vector maximum Fisher's discriminant ratio (F1v): this measure complements F1, modifying the orthogonal axis in order to improve data projection. Equation 2.4 illustrates this modification.

$$R(d) = \frac{\mathbf{d}^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \mathbf{d}}{\mathbf{d}^T \bar{\Sigma} \mathbf{d}} \quad (2.4)$$

Where:

- $\mathbf{d}$ is the directional vector where data are projected, calculated as $\mathbf{d} = \bar{\Sigma}^{-1}(\mu_1 - \mu_2)$;

- μ_i is the mean feature vector for the class c_i ;
- $\bar{\Sigma} = \alpha\Sigma_1 + (1 - \alpha)\Sigma_2, 0 \leq \alpha \leq 1$;
- Σ_i is the covariance matrix for the examples from the class c_i .

This measure can be calculated only for binary classification problems. A high F1v value indicates that there is a vector that separates the examples from distinct classes, after they are projected into a transformed space.

Overlapping of the per-class bounding boxes (F2): This measure calculates the volume of the overlapping region on the feature values for a pair of classes. This overlapping considers the minimum and maximum values of each feature per class in the dataset. A product of the calculated values for each feature is generated. Equation 2.5 illustrates F2 as it is defined in (Orriols-Puig et al., 2010), where f_i is the feature i and c_1 and c_2 are two classes.

$$F2 = \prod_{i=1}^m \left| \frac{\min(\max(f_i, c_1), \max(f_i, c_2)) - \max(\min(f_i, c_1), \min(f_i, c_2))}{\max(\max(f_i, c_1), \max(f_i, c_2)) - \min(\min(f_i, c_1), \min(f_i, c_2))} \right| \quad (2.5)$$

In multiclass problems, the final result is the sum of the values calculated for the underlying binary subproblems. A low F2 value indicates that the features can discriminate the examples of distinct classes and have low overlapping.

Maximum individual feature efficiency (F3): Evaluates the individual efficacy of each feature by considering how much each feature contributes to the classes separation. This measure uses examples that are not in overlapping ranges and outputs an efficiency ratio of linear separability. Equation 2.6 shows how F3 is calculated, where n is the number of examples in the training set and *overlap* is a function that returns the number of overlapping examples between two classes. High values of F3 indicate the presence of features whose values do not overlap between classes.

$$F3 = \max_{i=1}^m \frac{n - \text{overlap}(\mathbf{x}_{c_1}^{f_i}, \mathbf{x}_{c_2}^{f_i})}{n} \quad (2.6)$$

Collective feature efficiency (F4): based on F3, this measure evaluates the collective power of discrimination of the features. It uses an iterative procedure selecting the feature with the highest discrimination power and removing these examples from the dataset. The procedure is repeated until all examples are discriminated or all features are analysed, returning the proportion of instances that have been discriminated. Equation 2.7 shows how F4 is calculated, where $\text{overlap}(\mathbf{x}_{c_1}^{f_i}, \mathbf{x}_{c_2}^{f_i})_{T_i}$

measure the overlap in a subset of the data T_i generated by removing the examples already discriminated in T_{i-1} .

$$F4 = \sum_{i=1}^m \frac{\text{overlap}(\mathbf{x}_{c_1}^{f_i}, \mathbf{x}_{c_2}^{f_i})_{T_i}}{n} \quad (2.7)$$

Higher values indicate that more examples can be discriminated by using a combination of the available features.

2.2.2 Measures of Class Separability

Distance of erroneous instances to a linear classifier (L1): This measure quantifies the linearity of data, since the classification of linear separable data is considered a simpler classification task. L1 computes the sum of the distances of erroneous data to an hyperplane separating two classes. Support Vector Machine (SVM) with a linear kernel function (Vapnik, 1995) are used to induce the hyperplane. This measure is used only for binary classification problems. In Equation 2.8, $f(\cdot)$ is the linear function, $h(\cdot)$ is the prediction and y_i is the class of $\mathbf{x}_i$. Values equal to 0 indicate a linearly separable problem.

$$L1 = \sum_{h(\mathbf{x}_i) \neq y_i} f(\mathbf{x}_i) \quad (2.8)$$

Training error of a linear classifier (L2): Measures the predictive performance of a linear classifier for the training data. It also uses a SVM with linear kernel. Equation 2.9 shows how L2 is calculated. The $h(\mathbf{x}_i)$ is the prediction of the linear classifier obtained and $I(\cdot)$ is the evaluation measure which returns 1 if $\mathbf{x}_i$ is true and 0 otherwise. A lower training error indicate the linearity of the problem.

$$L2 = \frac{\sum_{i=1}^n I(h(\mathbf{x}_i) \neq y_i)}{n} \quad (2.9)$$

Fraction of points lying on the class boundary (N1): Estimates the complexity of the correct hypothesis underlying the data. Initially, a Minimum Spanning Tree (MST) is generated from the data, connecting the data points by their distances. The fraction of points from different classes that are connected in the MST is returned. Equation 2.10 defines how N1 is calculated. The $\mathbf{x}_j \in NN(\mathbf{x}_i)$ verify if x_j is the NN example and $y_i \neq y_j$ verify if they are examples of different class. High values of N1 indicate the need for more complex boundaries for separating the data.

$$N1 = \frac{\sum_{i=1}^n I(\mathbf{x}_j \in NN(\mathbf{x}_i) \text{ and } y_i \neq y_j)}{n} \quad (2.10)$$

Average intra/inter class nearest neighbor distances (N2): The mean intra-class and inter-class distances use the k -Nearest Neighbor (k -NN) (Mitchell, 1997) algorithm to analyse the spread of the examples from distinct classes. The intra-class distance considers the distance from each example to its nearest example in the same class, while the inter-class distance computes the distance of this example to its nearest example in other class. Equation 2.11 illustrates N2, where *intra* and *inter* are distance function.

$$N2 = \frac{\sum_{i=1}^n \text{intra}(\mathbf{x}_i)}{\sum_{i=1}^n \text{inter}(\mathbf{x}_i)} \quad (2.11)$$

Low N2 values indicate that examples of the same class are next to each other, while far from the examples of the other classes.

Leave-one-out error rate of the 1-NN algorithm (N3): Evaluates how distinct the examples from different classes are by considering the error rate of the 1-NN (Mitchell, 1997) classifier, with the *leave-one-out* strategy. Equation 2.12 shows the N3 measure. Low values indicate a high separation of the classes.

$$N3 = \frac{\sum_{i=1}^n I(1NN(\mathbf{x}_i) \neq y_i)}{n} \quad (2.12)$$

2.2.3 Measures of Geometry and Topology

Nonlinearity of a linear classifier (L3): Creates a new dataset by the interpolation of training data. New examples are created by linear interpolation with random coefficients of points chosen from a same class. Next, a SVM (Vapnik, 1995) classifier with linear kernel function is induced and its error rate for the original data is recorded. It is sensitive to the spread and overlapping of the data points and is used for binary classification problems only. Equation 2.13 illustrate the L3 measure, where l is the number of points and the examples generated by the interpolation. Low values indicate a high linearity.

$$L3 = \frac{\sum_{i=1}^l I(h(\mathbf{x}_i) \neq y_i)}{l} \quad (2.13)$$

Nonlinearity of the 1-NN classifier (N4): Has the same reasoning of L3, but using the 1-NN (Mitchell, 1997) classifier instead of the linear SVM (Vapnik, 1995). Equation 2.14 illustrate the N4 measure.

$$N4 = \frac{\sum_{i=1}^l I(1NN(\mathbf{x}_i) \neq y_i)}{l} \quad (2.14)$$

Fraction of maximum covering spheres on data (T1): Builds hyperspheres centered on the data points. The radius of these hyperspheres are increased until touching any example of different classes. Smaller hyperspheres inside larger ones are eliminated. It outputs the ratio of the number of hyperspheres formed to the total number of data points. Equation 2.15 shows T1, where $hyperpheres(D)$ returns the number of hyperspheres which can be built from the dataset. Low values indicate a low number of hyperspheres due to a low complexity of the data representation.

$$T1 = \frac{hyperpheres(D)}{n} \quad (2.15)$$

There are other measures presented in Ho & Basu (2002) and Orriols-Puig et al. (2010) that were not employed in this work because, by definition, they do not vary when the label noise level is increased. One of them is the dimensionality of the dataset and another is the ratio of the number of features to the number of data points (data sparsity).

2.2.4 Measures of Structural Representation

Before using these measures, it is necessary to transform the classification dataset into a graph. This graph must preserve the similarities and distances between examples, so that the data relationships are captured. Each data point will correspond to a node or vertex of the graph. Edges are added connecting all pairs of nodes or some of the pairs.

Several techniques can be used to build a graph for a dataset. The most common are the k -NN and the ϵ -NN (Zhu et al., 2005). While k -NN connects a pair of vertices i and j whenever i is one of the k -NN of j , ϵ -NN connects a pair of nodes i and j only if $d(i, j) < \epsilon$, where d is a distance function. We employed the ϵ -NN variant, since many edge and degree based measures could be fixed for k -NN, despite the level of noise inserted in a dataset. Afterwards, all edges between examples from different classes are pruned from the graph (Zhu et al., 2005). This is a postprocessing step that can be employed for labeled datasets, which takes into account the class information.

Figure 2.2 illustrates the graph build for the artificial binary dataset shown in Figure 2.1(b) which has two potential label noisy examples. The technique used to build the graph was the ϵ -NN with $\epsilon = 15\%$ of NN examples. Figure 2.2(a) shows the first step when the pairs of vertices with $d(i, j) < \epsilon$ are connected. Figure 2.2(b) shows the pruning process applied to the examples from different classes. With this kind of postprocessing the noise examples can be identified and measures about the level of noise can be extracted.

There are various measures able to characterize the topological and structural properties of a graph. Some of them come from the statistical characterization of complex networks (Kolaczyk, 2009). We used some of these graph-based measures in this work, which are referred by their original nomenclature, as follows:

Figure 2.2: Building a graph using ϵ -NN

Number of edges (Edges): Total number of edges contained in the graph. High values for edge-related measures indicate that many of the vertices are connected and, therefore, that there are many regions of high densities from a same class. This is true because of the postprocessing of edges connecting examples from different classes applied in this work. Equation 2.16 illustrate the measure, where v_{ij} is equal to 1 if i and j are connected, and 0 otherwise. Thus, the dataset is regarded as having low complexity if it shows a high number of edges.

$$edges = \sum_{i,j} v_{ij} \quad (2.16)$$

Average degree of the network (Degree): The degree of a vertex i is the number of edges connected to i . The average degree of a network is the average degree of all vertices in the graph. For undirected networks, it can be computed by Equation 2.17.

$$degree = \frac{1}{n} \sum_{i,j} v_{ij} \quad (2.17)$$

The same reasoning of edge-related measures applies to degree based measures, since the degree of a vertex corresponds to the number of edges incident to it. Therefore, high values for the degree indicates the presence of many regions of high densities from a same class, and the dataset can be regarded as having low complexity.

Average density of network (Density): The density of a graph is the fraction of the

number of edges it contains by the number of possible edges that could be formed. The average density also allows capturing whether there are dense regions from the same class in the dataset. Equation 2.18 illustrates the measure, where n is the number of vertices and $\frac{n(n-1)}{2}$ is the number of possible edges. High values indicate the presence of such regions and a simpler dataset.

$$density = \frac{2}{n(n-1)} \sum_{i,j} v_{ij} \quad (2.18)$$

Maximum number of components (MaxComp): Corresponds to the maximal number of connected components of a graph. In an undirected graph, a component is a subgraph with paths between all of its nodes. When a dataset shows a high overlapping between classes, the graph will probably present a large number of disconnected components, since connections between different classes are pruned from the graph. The minimal component will tend to be smaller in this case. Thus, we will assume that smaller values of the MaxComp measures represent more complex datasets.

Closeness centrality (Closeness): Average number of steps required to access every other vertex from a given vertex, which is the number of edges traversed in the shortest path between them. It can be computed by the inverse of the distance between the nodes, as shown in Equation 2.19:

$$closeness = \frac{1}{\sum_{i \neq j} d(v_{ij})} \quad (2.19)$$

Since the closeness measure uses the inverse of the shortest distance between vertices, larger values are expected for simpler datasets that will show low distances between examples from the same class.

Betweenness centrality (Betweenness): The vertex and edge betweenness are defined by the average number of shortest paths that traverses them. We employed the vertex variant. Equation 2.20 represents the betweenness value of a vertex v_j , where $d(v_{il})$ is the total number of the shortest paths from node i to node l and $d_j(v_{il})$ is the number of those paths that pass through j :

$$betweenness(v_j) = \sum_{i \neq j \neq l} \frac{d_j(v_{il})}{d(v_{il})} \quad (2.20)$$

The value of Betweenness will be small for simpler datasets, since the distance between the shortest paths and the paths which pass through j will be close.

Clustering Coefficient (ClsCoef): Measures the probability that adjacent vertices of a graph are connected. The clustering coefficient of a vertex v_i is given by the

ratio of the number of edges between its neighbors (k_i) and the maximum number of edges that could possibly exist between these neighbors. Equation 2.21 illustrates this measure. Measure *ClsCoef* will be higher for simpler datasets, which will produce large connected components joining vertices from the same class.

$$ClsCoef(v_i) = \frac{2}{k_i(k_i - 1)} \sum_{i,j \in k} v_{ij} \quad (2.21)$$

Hub score (Hubs): Measures the score of each node by the number of connections it has to other nodes, weighted by the number of connections these neighbors have. That is, more connected vertices, which are also connected to highly connected vertices, have higher hub score. The hub score is expected to have a low mean for high complexity datasets, since strong vertices will become less connected to strong neighbors. For instance, hubs are expected at regions of high density from a given class. Therefore, simpler datasets with high density will show larger values for this measure.

Average Path Length (AvgPath): Average size of all shortest paths in the graph. It measures the efficiency of information spread in the network. It is illustrated by Equation 2.22, where n represents the number of vertices of the graph and $d(v_{ij})$ is the shortest distance between vertices i and j .

$$AvgPath = \frac{2}{n(n-1)} \sum_{i \neq j} d(v_{ij}); \quad (2.22)$$

For the *AvgPath* measure, high values are expected for low density graphs, indicating an increase in complexity.

For those measures that are calculated for each vertex individually, we computed an average for all vertices in the graph. The graph measures used in this study mainly evaluate the overlapping of the classes and their density.

A previous paper also investigated the use of complex-network measures to characterize supervised datasets (Morais & Prati, 2013). It used part of the measures presented here to design meta-learning models able to predict the best performing model between a pair of classifiers for a given dataset. They also compared these measures to those from Ho & Basu (2002), but in a distinct scenario from the one adopted here. It is not clear whether they employ a postprocessing of the graph for removing edges between nodes of different classes, as done in this work. Also, some of the measures employed in that work are not suitable for our scenario and are not used here. One example is the number of nodes of the graph, which will not vary for a given dataset despite of its noise level. The only measures in common to those used in Morais & Prati (2013) are the number of edges, the

average clustering coefficient and the average degree. Besides introducing new measures, we also describe the behavior of all measures for simpler or complex problems. Moreover, we try to identify the best suited measures for detecting the presence of label noise in a dataset.

2.2.5 Summary of Measures

Table 2.1 summarizes the measures employed to characterize the complexity of the datasets used in this study. For each measure, we present upper (Maximum value) and lower bounds (Minimum value) achievable and how they are associated with the increase or decrease of complexity of the classification problems (Complexity column). For a given measure, the value in column “Complexity” is “+” if higher values of the measure are observed for high complexity datasets, that is, when the measure value correlates positively to the complexity level. On the other hand, the “-” sign denotes the opposite, so that low values of the measure are obtained for high complexity datasets, denoting a negative correlation.

Table 2.1: Summary of Measures.

Type of Measure	Measure	Minimum Value	Maximum Value	Complexity
Overlapping in feature values	F1	0	$+\infty$	-
	F1v	0	$+\infty$	-
	F2	0	$+\infty$	+
	F3	0	1	-
	F4	0	$+\infty$	-
Class separability	L1	0	$+\infty$	+
	L2	0	1	+
	N1	0	1	+
	N2	0	$+\infty$	+
	N3	0	1	+
Geometry and topology	L3	0	1	+
	N4	0	1	+
	T1	0	1	+
Structural representation	Edges	0	$n * (n - 1)/2$	-
	Degree	0	$n - 1$	-
	MaxComp	1	n	-
	Closeness	0	$1/(n - 1)$	-
	Betweenness	0	$(n - 1) * (n - 2)/2$	+
	Hubs	0	1	-
	Density	0	1	-
	ClsCoef	0	1	-
	AvgPath	$1/n * (n - 1)$	0.5	+

Most of the bounds were obtained considering the equations directly, while some of the graph-based bounds were experimentally defined. For instance, for the F1 measure, if the means of the feature values are always equal, meaning that the classes overlap for all features (an extreme case), the nominator of Equation 2.2 will be 0. Similarly, a maximum value cannot be determined for F1, as it is dependent on the feature values of each dataset.

We denote that by the “ ∞ ” value in the Table 2.1. In the case of graph-based measures, we generated graphs representing simple and complex relations between the same number of data points and observed the achieved measure values. A simple graph would correspond to a case where the classes are well separated and there is a high number of connections between examples from the same class, while a complex dataset would correspond to a graph where examples of different classes are always next to each other and ultimately the connections between them are pruned according to our graph construction method.

2.3 Evaluating the Complexity of Noisy Datasets

This section presents the experiments performed to evaluate how the different data complexity measures from Section 2.2 behave in the presence of label noise for several benchmark public datasets. First, a set of classification benchmark datasets were chosen for the experiments. Different levels of label noise were later added to each dataset. The experiments also monitor how the complexity level of the datasets are affected by noise imputation. This is accomplished by:

1. Verifying the Spearman correlation between the measure values with the noise rates artificially imputed and the predictive performance of a group of classifiers. This analysis allows the identification of a set of measures that are more sensitive to the presence of noise in a dataset.
2. Evaluating the correlation between the measure values in order to identify those measures that (i) capture different concepts regarding noisy environments and (ii) can be jointly used to support the development of new noise-handling techniques.

The next sections present in detail the experimental protocol previously outlined.

2.3.1 Datasets

Two groups of datasets, artificial and real datasets, were selected for the experiments. The artificial datasets were introduced and generously provided by Amancio et al. (2013). The authors generated artificial classification datasets based on multivariate Gaussians, with different levels of overlapping between the classes. For the study carried out in this Thesis, 180 balanced datasets (with the same number of examples per class) with 2 classes, containing 2, 10 and 50 predictive features and with different overlapping rates for each of the number of features were selected. The datasets were selected according to observations made in a recent work (Smith et al., 2014), which points out that class overlap seems to be a principal contributor to instance hardness and that noisy data can ultimately be considered hard instances.

Regarding the real datasets, 90 benchmarks were selected from the UCI¹ repository (Lichman, 2013). Because they are real, it is not possible to assert that they are noise-free, although some of them are artificial and show no label inconsistencies. Nonetheless, a recent study showed that most of the datasets from UCI can be considered easy problems, once many classification techniques are able to obtain high predictive accuracies when applied to them (Maciá & Bernadó-Mansilla, 2014). Table 2.2 summarizes the main characteristics of the datasets used in the experiments of this Thesis: number of examples (#EX), number of features (#FT), number of classes (#CL) and percentage of the examples in the majority class (%MC).

In order to corrupt the datasets with noise, the uniform random addition method, which is the most common type of artificial noise imputation method for classification tasks (Zhu & Wu, 2004), was used. For each dataset, noise was inserted at different levels, namely 5%, 10%, 20% and 40%. Thus, making possible to investigate the influence of increasing noise levels in the results. Besides, all datasets were partitioned according to 10-fold-cross-validation, but noise was inserted only in the training folds. Once the selection of examples was random, 10 different noisy versions of the training data for each noise level were generated.

2.3.2 Methodology

Figure 2.3 shows the flow chart of the experimental methodology. First, noisy versions of the original datasets from Section 2.3.1 were created by using the previously described systematic model of noise imputation. The complexity measures and the predictive performance of classifiers were extracted from the original training datasets and from their noisy versions.

To calculate the complexity measures described from Section 2.2.1 to Section 2.2.3, the Data Complexity Library (DCoL) (Orriols-Puig et al., 2010) was used. All distance-based measures employed the normalized euclidean distance for continuous features and the *overlap* distance for nominal features (this distance is 0 for equal categorical values and 1 otherwise) (Giraud-Carrier & Martinez, 1995). To build the graph for the graph-based measures, the ϵ -NN algorithm, with the ϵ threshold value equal to 15%, was used, like in Morais & Prati (2013). The measures described in Section 2.2.4 were calculated using the Igraph library (Csardi & Nepusz, 2006). Measures like the directional-vector Fisher’s discriminant ratio (F1v) and collective feature efficiency (F4) from Orriols-Puig et al. (2010) were disregarded in this particular analysis, since they have a concept similar to other measures already employed.

The application of these measures result in one meta-dataset, which will be employed in the subsequent experiments. This meta-dataset contains 20 meta-features (# complexity

¹<https://archive.ics.uci.edu/ml/datasets.html>

Table 2.2: Summary of datasets characteristics: name, number of examples, number of features, number of classes and the percentage of the majority class.

Dataset	#EX	#FT	#CL	%MC	Dataset	#EX	#FT	#CL	%MC
abalone	4153	9	19	17	meta-data	528	22	24	4
acute-nephritis	120	7	2	58	mines-vs-rocks	208	61	2	53
acute-urinary	120	7	2	51	molecular-promoters	106	58	2	50
appendicitis	106	8	2	80	molecular-promotor	106	58	2	50
australian	690	15	2	56	monks1	556	7	2	50
backache	180	32	2	86	monks2	601	7	2	66
balance	625	5	3	46	monks3	554	7	2	52
banana	5300	3	2	55	movement-libras	360	91	15	7
banknote-authentication	1372	5	2	56	newthyroid	215	6	3	70
blogger	100	6	2	68	page-blocks	5473	11	5	90
blood-transfusion-service	748	5	2	76	parkinsons	195	23	2	75
breast-cancer-wisconsin	699	10	2	66	phoneme	5404	6	2	71
breast-tissue-4class	106	10	4	46	pima	768	9	2	65
breast-tissue-6class	106	10	6	21	planning-relax	182	13	2	71
bupa	345	7	2	58	qualitative-bankruptcy	250	7	2	57
car	1728	7	4	70	ringnorm	7400	21	2	50
cardiotocography	2126	21	10	27	saheart	462	10	2	65
climate-simulation	540	21	2	91	seeds	210	8	3	33
cmc	1473	10	3	43	segmentation	2310	19	7	14
collins	485	22	13	16	spectf	349	45	2	73
colon32	62	33	2	65	spectf-heart	349	45	2	73
crabs	200	6	2	50	spect-heart	267	23	2	59
dbworld-subjects	64	243	2	55	statlog-australian-credit	690	15	2	56
dermatology	366	35	6	31	statlog-german-credit	1000	21	2	70
expgen	207	80	5	58	statlog-heart	270	14	2	56
fertility-diagnosis	100	10	2	88	tae	151	6	3	34
flags	178	29	5	34	thoracic-surgery	470	17	2	85
flare	1066	12	6	31	thyroid-newthyroid	215	6	3	70
glass	205	10	5	37	tic-tac-toe	958	10	2	65
glioma16	50	17	2	56	titanic	2201	4	2	68
habermans-survival	306	4	2	74	user-knowledge	403	6	5	32
hayes-roth	160	5	3	41	vehicle	846	19	4	26
heart-cleveland	303	14	5	54	vertebra-column-2c	310	7	2	68
heart-hungarian	294	14	2	64	vertebra-column-3c	310	7	3	48
heart-repro-hungarian	294	14	5	64	voting	435	17	2	61
heart-va	200	14	5	28	vowel	990	11	11	9
hepatitis	155	20	2	79	vowel-reduced	528	11	11	9
horse-colic-surgical	300	28	2	64	waveform-5000	5000	41	3	34
indian-liver-patient	583	11	2	71	wdbc	569	31	2	63
ionosphere	351	34	2	64	wholesale-channel	440	8	2	68
iris	150	5	3	33	wholesale-region	440	8	3	72
kr-vs-kp	3196	37	2	52	wine	178	14	3	40
led7digit	500	8	10	11	wine-quality-red	1599	12	6	43
leukemia-haslinger	100	51	2	51	yeast	1479	9	9	31
mammographic-mass	961	6	2	54	zoo	84	17	4	49

and graph-based measures) and 4 predictive performance obtained from the application of 4 classifiers to the benchmark datasets and their noisy versions. This meta-dataset has therefore 3690 examples: 90 (# original datasets) + 90 (# datasets) * 4 (# noise levels) * 10 (# random versions).

Three types of analysis were performed using the meta-dataset: (i) correlation between the measure values and the noise level of the datasets; (ii) correlation between measure values and predictive performance of classifiers and (iii) correlation within the measure

Figure 2.3: Flowchart of the experiments.

values. The first and second analysis will consider all measures. The results obtained in these analyzes will then refine a subset of measures more sensitive to noise imputation, which will be further analyzed in the third correlation study.

The first analysis verifies if there is a direct relation between the noise level of a dataset and the values of the measures extracted from the dataset. This allows the identification of the measures that are more sensitive to the presence of noise. For such, the Spearman's rank correlation between the measure values and the different noise levels was calculated for all datasets. Those measures that present a significant correlation according to the Spearman's statistical test (at 95% of confidence value) were selected for further analysis.

It is important to observe that the real datasets have intrinsic noise. Therefore, the noise rates artificially added could not match to the rate of noise present in the data. The predictive performance of a classifier for a particular dataset is often associated with the difficulty of the classification problem represented by this dataset (Lorena et al., 2012; Maciá & Bernadó-Mansilla, 2014). It is intuitive that for easy classification problems it is also easy to obtain a plausible and highly accurate classification hypothesis, while the opposite is verified for difficult problems. It is also true that a classification task tends to become more difficult as noise is added to its data (Zhu & Wu, 2004).

The second analysis verifies if there is a direct relation between the accuracy rates

obtained by the classifiers induced by each algorithm and the measured values extracted from the datasets. Algorithms from different paradigms were induced using the original and corrupted training datasets: C4.5 (Quinlan, 1986b), 3-NN (Mitchell, 1997), Random Forest (RF) (Breiman, 2001) and SVM (Vapnik, 1995) with a radial kernel function. Spearman’s statistical test (at 95% of confidence value) were selected for additional analysis.

The third analysis evaluates the Spearman correlation between the measures with the highest sensitivity to the presence of noise according to the previous experimental results. It looks for overlapping in the complexity concepts extracted by these measures. Similar analyses are carried out in Smith et al. (2014) for accessing the relationship between some instance hardness measures proposed by the authors. While a high correlation could indicate that the measures are capturing the same complexity concepts, a low correlation could indicate that the measures complement each other, an issue that can be further explored.

2.4 Results obtained in the Correlation Analysis

This section presents the experimental results for the correlation analysis previously described. We also have evaluated the results for some artificial datasets as described in Section 2.3.1. These results were quite similar to those observed for the benchmark datasets, with the difference that the absolute correlation values calculated were higher for the artificial datasets. Therefore, they are omitted here.

Figure 2.4 presents histograms of the values of the complexity measures for all benchmark datasets when random noise is added. The bars are colored according to the amount of noise inserted, from 0% (original datasets) to 40%. The measure values were normalized considering all datasets to allow their direct comparison. It is possible to notice that some of the measures are more sensitive to noise imputation and present clear limits on their values for different noise levels. They are: N1, N3, Edges, Degree and Density. On the other hand, other measures like Betweenness do not present a clear contrast in their values for different noise levels.

Furthermore, it is also possible to notice from Figure 2.4 that, as more noise is added to the datasets, the complexity of the classification problem tends to increase. This is reflected in the values of the majority of the complexity measures, that either increased or decreased when noise is added, in accordance to their positive or negative correlation to the complexity level, as shown in Table 2.1 (column “Complexity”). For instance, higher N1 values are expected for more complex datasets and the N1 values indeed increased for higher levels of noise. On the other hand, lower F1 values are expected for more complex datasets and we can observe that as more noise is added to the datasets, the F1 values tend to reduce.

Figure 2.4: Histogram of each measure for distinct noise levels.

2.4.1 Correlation of Measures with the Noise Level

Figure 2.5 shows the correlation between the values of the measures for the different noise levels in the datasets. Positive and negative values are plotted in order to show clearly which measures are directly or indirectly correlated to the noise levels. It is noticeable that, as the noise level increases, the values of the complexity measures either increase or reduce accordingly, indicating increases in the complexity level of the noisy datasets. The closer to 1 or -1 , the higher is the relation between the measure and the noise level.

According to the statistical test employed, 19 measures presented significant correlation to the noise levels, at 95% of confidence. Among the measures with direct correlation

Figure 2.5: Correlation of each measure to the noise levels.

to the noise level, nine are basic complexity measures from the literature (N3, N1, N2, L2, N4, L1, T1, F2, and L3). These measures mainly capture: classes separability (N3, N1, N2, L2 and L1), data topology according to a NN (Mitchell, 1997) classifier (N4, T1 and L3) and individual feature overlapping (F2). Regarding those measures indirectly related to the noise levels, two are basic complexity measures based on feature overlapping (F1 and F3), while six are based on structural representation (Density, Hub, Degree, ClsCoef, Edges and MaxComp). Only the Betweenness measure did not present significant correlation to the noise levels. As expected, the most prominent measures are the same that showed more distinct values for different noise levels in the histograms from Figure 2.4.

Despite the statistical difference, it is possible to notice some low correlation values in Figure 2.5. Only the measures N3, N1 and N2 presented correlation values higher than 0.5. These correlations were higher in the experiments with artificial datasets. This can be a result of the fact that, for real datasets, the amount of noise added is potential rather than actual.

2.4.2 Correlation of Measures with the Predictive Performance

Figure 2.6 relates the values of the measures with the predictive performance of four classification techniques: C4.5 (Quinlan, 1986b), k -NN (Mitchell, 1997), RF (Breiman, 2001) and SVM (Vapnik, 1995). The values are plotted in order to show clearly which measures are directly or indirectly correlated to the accuracy of the classifiers. The closer to 1 or -1 , the higher is the relation between the measure and accuracy of the classifiers.

Again, using the Spearman's rank correlation coefficient, 16 measures show statistical difference regarding the RF correlation results (technique with the best overall predictive performance in this study). Besides, the MaxComp, Edges, Betweenness, L1 and ClsCoef

Figure 2.6: Correlation of each measure to the predictive performance of classifiers.

measures presented low correlation values. Although there are differences in the rankings of the measures for distinct classification techniques, they are mostly similar. Measures like N1, N3, N2, N4 and Density have high correlation. The importance assigned to these measures coincide with those from the previous analysis, reinforcing their relevance in capturing effects of data alterations that arise from the presence of noise.

2.4.3 Correlation Between Measures

In order to verify whether the measures capture similar or distinct information from data, we calculated pairwise correlations between their values. Only those measures considered more relevant in the previous analysis were considered. These measures were highlighted as more sensitive to noise imputation and can therefore be successfully employed for noise identification.

Figure 2.7 shows a heatmap of the correlation between these pairs of measures. Each column and row corresponds to a measure. Each box is colored according to the correlation values calculated, from gray (highest correlation, despite positive or negative) to white (lowest correlation). The absolute values of all correlations are also shown inside the heatmap cells. We highlight in bold the correlation values that are not significant according to the Spearman's correlation test (at 95% of confidence level). These pairs of measures correspond to those that can potentially complement each other.

According to the heatmap, various measures are weakly correlated to each other. Therefore, they capture distinct aspects from the data. As expected, the measures N1, N2, N3 and N4 from Ho & Basu (2002) are highly correlated. They are all based on NN information. Despite the fact that all structural representation measures are extracted from a NN graph, their correlation to N1, N2, N3 and N4 is low in several cases. Among the

Figure 2.7: Heatmap of correlation between measures.

graph-based measures, high correlations are observed between Edges, Degree, Closeness and MaxComp. Since the degree of a graph is calculated considering the number of its edges and number of connected components, this correlation is expected by definition.

It is interesting to notice that many of the measures highlighted as distinguishing the noise levels have low correlation between them. This is particularly true for class separability measures (e.g., N3) when paired to the structural representation measures (e.g., Closeness, Degree and Edges). Therefore, they could be combined to improve noise identification and handling. This issue is preliminarily investigated in the proposal of a new NF technique, which will be described in the next chapter.

2.5 Chapter Remarks

This chapter defined label noise and investigated how its presence affects the complexity of classification tasks, by monitoring the values of simple measures extracted from datasets with increasing noise levels. Part of these measures were already used in the literature for understanding and analyzing the complexity of classification tasks. Some other measures that are based on the modeling of datasets by graphs were introduced in this study.

Experimentally, measures able to capture characteristics like separability of the classes, alterations in the class boundary and densities within the classes were the most affected by the introduction of label noise in the data. Therefore, they are good candidates for further exploitation and to support the design of new noise identification techniques and noise-tolerant classification algorithms. Moreover, experimental results showed a low correlation between the basic complexity measures and the graph-based measures, stressing the relevance of exploring different views and representations of the data structure.

The graph-based measures Closeness, Hub, Edges, Degree and Density were highlighted in all analysis carried out. This may have occurred because, when label noise is introduced, examples from distinct classes become closer to each other and are not connected in the graph. The standard data complexity measures, those that rely on NN information, as N1 and N3, were also able to better capture the effects of noise imputation. This is also due to the fact that label noise tends to affect the spatial proximity of data from different classes. Thus, the idea that data from the same class tend to be next from each other in the feature space, while far from examples from different classes, is reinforced.

The results presented in this chapter are part of the journal paper:

- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2015). “Effect of label noise in the complexity of classification problems”. *Neurocomputing*, 160:108 - 119.

Chapter 3

Noise Identification

The identification of noise in classification datasets has been the subject of several studies. They follow two main approaches: (i) designing classification techniques that are more tolerant and robust to noise (Frenay & Verleysen, 2014) and, (ii) data cleaning in a previous preprocessing step (Sluban et al., 2014).

The pruning process in DT induction algorithms is an early initiative to increase the robustness of classification models to noisy data (Quinlan, 1986b). Nonetheless, if the noise level is high, the definition of the pruning degree can be challenging and can ultimately remove branches that are based in safe information too. Another example is the use of slack variables in the SVM training (Vapnik, 1995), which allow some examples to be misclassified or to lie within the margins of separation between the classes. This introduces an additional parameter to be tuned during the SVM training: the regularization constant, which accounts for the amount of training examples that can be misclassified or to be placed near the decision boundary.

Recent work addresses noise-tolerant classifiers, where a label noise model is learnt jointly to the classification model itself (Smith et al., 2014). For such, typically, some information must be available about the label noise or its effects (Eskin, 2000; Frenay & Verleysen, 2014). The learning algorithm can also be modified to embed data cleansing (Ganapathiraju & Picone, 2000). Other authors prefer to treat noise previously, in a preprocessing step. Filters are developed for such, which scan the dataset for unreliable data (Sluban et al., 2010; Garcia et al., 2012; Sluban et al., 2014). The preprocessed dataset can then be used as input to any classification algorithm.

Several studies show the benefits from using class Noise Filtering (NF) techniques regarding improvements in the classification predictive performance and the reduction in the complexity of the classifiers built (Brodley & Friedl, 1999; Sluban et al., 2014; Garcia et al., 2012; Sáez et al., 2016). NF techniques can use different information to detect noise, such as those employing neighborhood or density information (Wilson, 1972; Tomek, 1976; Garcia et al., 2015), descriptors extracted from the data (Gamberger et al., 1999; Sluban et al., 2014) and noise identification models induced by classifiers (Sluban et al., 2014) or

ensembles of classifiers (Brodley & Friedl, 1999; Verbaeten & Assche, 2003; Sluban et al., 2010; Garcia et al., 2012).

The majority of the existent NF techniques only point examples as noisy or not, in a crisp decision. In this chapter some of these techniques are adapted to provide a soft decision. Thereby, each example is assigned a probability of being noisy and the examples from the dataset can be ranked according to their (un)reliability value. The main advantage of this approach is the possible identification of noisy examples which are more difficult, which can be further analyzed by a domain expert. For evaluating the efficacy of these noise rankers, this chapter also presents new evaluation measures which take into account the orderings produced.

Another investigation performed was to combine individual NF techniques into ensembles. This approach can provide more robustness in noise identification, since it is usually not possible to guarantee that a given example is truly noisy without relying on an expert judgment. Besides, each NF technique has a distinct bias and can present the best performance for some specific datasets (García et al., 2015; Wu & Zhu, 2008). By aggregating the bias of different individual techniques, the ensembles can present a high noise detection accuracy for a larger number of datasets than the individual techniques used alone.

The contributions introduced in this chapter can be summarized as:

- Proposal of two new NF techniques. One of them based in the experimental results presented in the previous chapter, which considers measures of data complexity. The other one is an adaptation of an ensemble of classifiers for noise identification;
- Adaptation of various NF techniques to provide a soft decision, that is, a degree of confidence in noise prediction;
- Proposal of a new evaluation measure for the soft decision filters: the Area Under the ROC Curve (AUC) obtained in NF analysis;
- Investigation of the effects of combining multiple soft NF techniques into ensembles.

The rest of this chapter is organized as follows. Section 3.1 has an overview of the crisp NF techniques investigated in this study. The adaptations to obtain soft predictions are described in Section 3.2. Section 3.3 describes the measures used to evaluate the NF. Section 3.4 describes the experiments carried out to evaluate these techniques, while Sections 3.5 and 3.6 report and analyze the experimental results obtained. Finally, Section 3.7 summarizes the main conclusions from this chapter.

3.1 Noise Filters

NF techniques (Brodley & Friedl, 1999; Garcia et al., 2012; Sluban et al., 2010, 2014; Garcia et al., 2015; Tomek, 1976) are preprocessing methods that can be applied to any given dataset, outputting the potential noisy examples (Frenay & Verleysen, 2014). Some filters try to relabel the potential noisy examples, instead of removing them (Garcia et al., 2012). Nonetheless, the most common approach is to remove the unreliable examples, producing a new reduced dataset.

Most of the existing filters also focus on the elimination of examples with class noise, which has shown to be advantageous (Gamberger et al., 1999). In contrast, the elimination of examples with feature noise is not as beneficial (Zhu & Wu, 2004), since other features from these examples may be useful to build the classifier. Next, the NF techniques considered in this study are presented.

3.1.1 Ensemble Based Noise Filters

The NF techniques (Brodley & Friedl, 1999; Garcia et al., 2012; Sluban et al., 2010) based on ensembles use a set of classifiers in order to improve the noise detection. The motivation for using ensembles is that if distinct classifiers disagree on their predictions for an instance, the instance is probably incorrectly labeled. The main possible disadvantage of using ensembles for noise detection is the increase of complexity by the generated model and the increase of the computational cost of the filter.

There are also many aggregation strategies to combine the predictions of the classifiers in noise identification when ensembles are employed (Brodley & Friedl, 1996). The most common are consensus and majority voting strategies. In the first, an example is considered noisy if all classifiers in the ensemble misclassify it. In the second, an example is considered noisy if the majority of the classifiers in the ensemble misclassify it.

In Brodley & Friedl (1999), for instance, the authors describe strategies and a set of NF techniques based on combination of predictions of distinct classifiers in noise identification. According to the authors, the predictions made by k -NN (Mitchell, 1997), C4.5 (Quinlan, 1986b) and linear SVM (Vapnik, 1995) using majority vote with 10-fold cross-validation presented the best predictive performance. This filter will be referred as Static Ensemble Filter (SEF), because the set of classifiers composing the ensemble is fixed. Algorithm 1 describes this filter. The input of the algorithm are the training data (E), testing data (T) and the testing data label (Y). The output is the noisy examples subset (A). For a given sample i and the evaluation of a given classifier j , the prediction is saved in the prediction vector $P_{i,j}$. After the evaluation for all classifiers, the *majority* voting strategy function compare the label (Y) with the prediction. If the majority of the models classified the sample as noisy, it is added in the critical subset.

In Dynamic Ensemble Filter (DEF) (Garcia et al., 2012), the authors tried to increase

Algorithm 1 SEF**Input:** E (training data), T (testing data), Y (testing data class) C (classifiers)**Output:** A (critical example set)

```

 $A \leftarrow \emptyset$ 
for  $i \leftarrow 1, \dots, |T|$  do
  for  $j \leftarrow 1, \dots, |C|$  do
     $P_{i,j} \leftarrow C_j(E, T_i)$ 
  end for
  if  $\text{majority}(P_i) \neq Y_i$  then
     $A \leftarrow A \cup T_i$ 
  end if
end for
return  $A$ 

```

the robustness of the SEF filter choosing the classifiers to be combined in noise identification based on a criterion that considers the agreements in the predictions. Thus, the set of classifiers combined is dynamically adapted for each dataset. Algorithm 2 describes the function to select the m classifiers with best agreement to compose the ensemble. The input of the algorithm are the training (E) and testing (T) data as well the classifiers (C) available and the number (m) of classifiers to be selected. The output is a vector (V) with the classifiers chosen to compose the ensemble. First, all C classifiers are applied to the training and testing data. The prediction $P_{i,j}$ correspond to the prediction of each C_j classifier to each example i of the testing data. The next step is generate all the m -combination of the predictions P with the *combination* function. With all the G combinations, we can evaluate the *agreement* function on each G_i and return an agreement (V_i) index. In Garcia et al. (2012) the *agreement* function is the amount of concordances in the predictions made by the pairs of classifiers. Finally, the algorithm return the m classifiers which together have the maximum agreement. The next step is the application of the algorithm 1 with the selected classifiers. As in SEF, afterwards a consensus or majority voting aggregation strategy can be used to combine the predictions of the classifiers chosen in the previews step and assess whether an example is noisy or not.

The main disadvantage of DEF filter is the exponential increase of the number of possible combinations for large numbers of classifiers. Based on that, in this work we first listed a set of classification techniques that can be chosen to compose the DEF ensemble coming from different learning paradigms, so that they can complement each other: SVM (Vapnik, 1995) with linear and radial kernel functions, RF (Breiman, 2001), k -NN (Mitchell, 1997), DTs induced with C4.5 (Quinlan, 1986b) and Naive Bayes (NB) (Lewis, 1998). Next, for choosing the set of classifiers composing the ensemble in DEF, their individual 10-fold-cross-validation predictive performance on training data is considered, so that the $m = 3$ classifiers with best performance are selected.

Another recent ensemble is the High Agreement Random Forest Filter (HARF) method

Algorithm 2 Selecting m classifiers to compose the DEF ensemble

Input: E (training data), T (testing data), C (classifiers)

Output: V (classification techniques to be combined)

```

 $V \leftarrow \emptyset$ 
for  $i \leftarrow 1, \dots, |T|$  do
  for  $j \leftarrow 1, \dots, |C|$  do
     $P_{i,j} \leftarrow C_j(E, T_i)$ 
  end for
end for
 $G \leftarrow \text{combination}(P, m)$ 
for  $i \leftarrow 1, \dots, |G|$  do
   $V_i \leftarrow \text{agreement}(G_i)$ 
end for
return  $\max(V)$ 

```

(Sluban et al., 2010, 2014), which uses RF classifiers in noise identification. The algorithm considers the rate of disagreement in the predictions made by the individual trees of the forest using 10-fold-cross-validation to detect the noisy examples: if the rate is relatively high (70% up to 90%), the example is probably noisy; otherwise, it is considered to be clean.

3.1.2 Noise Filters Based on Data Descriptors

The Saturation Filter was initially proposed by Gamberger & Lavrač (1997) to explore the notion of training data saturation and the Occam's Razor theory. A saturated set can be defined as a dataset that allows the induction of a correct and simple hypothesis, capturing all relevant information required to represent the data. Thereby, the algorithm search those examples which if removed could transform an unsaturated dataset into a saturated dataset.

The identification of the noise examples are made by the reduction of a measure named Complexity of the Least Correct Hypothesis (CLCH), associated to each training data. To estimate the CLCH value, the problem is first represented in first order language. Next, this formalized dataset is fed into the filter, which removes (α) example(s) per iteration, generating all possible combinations of saturated data. If the CLCH value decreases when a subset of examples is removed, the subset is considered noisy. This step is called Saturation Test (ST) and is represented in Algorithm 3. The input of the algorithm is the training data (E) and the output is the subset of noisy examples (A) whose elimination may lead to a saturated training data. The S represent all possible subset of examples when one ($\alpha = 1$) example is removed from the training data. This algorithm generate all possible subset (S_i) and compare the CLCH of S_i with the CLCH of the training data. If the CLCH decrease for the subset S_i , the example i is included in the critical example subset. This condition is tested for all examples.

Algorithm 3 Saturation Test

Input: E (training data)**Output:** A (critical example set)

```

 $A \leftarrow \emptyset$ 
for  $i \leftarrow 1, \dots, |E|$  do
   $S_i \leftarrow E \setminus E_i$ 
  if  $CLCH(S_i) < CLCH(E)$  then
     $A \leftarrow A \cup E_i$ 
  end if
end for
return  $A$ 

```

The iterative noise elimination algorithm that results in the reduced training set with eliminated noisy examples is presented in Algorithm 4. This procedure continues until no example is marked as noisy or until a stop criterion is reached. The input of the algorithm is the training data (E) and the output is the subset of noisy examples (A). The critical subset A starts the process with no element. While the *SaturationTest* algorithm return noisy example (S), it is removed from the training data (E) and included in the subset of critical examples (A). If no example is returned from the *SaturationTest* the process is stopped.

Algorithm 4 Saturation Filter

Input: E (training data)**Output:** A (critical example set)

```

 $A \leftarrow \emptyset$ 
while TRUE do
   $S \leftarrow \text{SaturationTest}(E)$ 
  if  $S \neq \emptyset$  then
     $E \leftarrow E \setminus S$ 
     $A \leftarrow A \cup S$ 
  else
    break
  end if
end while
return  $A$ 

```

Some effort has been made in Gamberger et al. (1999) to decrease the computational cost of Saturation Filter (SF), once the exhaustive search prevents its execution for large datasets. In this new approach, the examples are marked with weights that represent a probability *a priori* to be noise. However, this algorithm is still exhaustive and depends on a parameter related to a sensitivity value. In Sluban et al. (2014), new efforts were made to reduce the computational burden of SF. The proposed modifications were to use a DT to prune the examples that are most probably noisy before applying the SF iterations. The size of a DT without pruning is used to estimate the CLCH value (Sluban

et al., 2014).

The Graph Nearest Neighbor (GNN) was proposed in Garcia et al. (2015) based on the results presented in Chapter 2. The GNN filter identify noisy examples by first constructing a graph from the dataset, as described in Section 2.2.4. Afterwards, it uses the degree of each vertex for pointing an example as a potential noise. The degree measure has demonstrated a high correlation to the noise levels in the experiments carried out in Section 2.4. In fact, when an example is misclassified, it will be probably close to examples from another class(es). In this case, its edges to close examples will be pruned and the example will tend to have a low degree value. Safe examples, on the other hand, will be connected to a high number of examples from the same class and show a high degree value. For this reason, the degree of each vertex in the graph will be initially examined to point an example as potential noise. Next it is necessary to stipulate a threshold on the node degree so as its mapped example can be really considered as noisy. Figure 3.1 illustrates this with a graph building with an artificial binary dataset. Figure 3.1(a) shows an artificial dataset with two classes ($\bullet$ and $\blacktriangle$) that are non linearly separable and that contains four potential label noisy examples in red. Figure 3.1(b) shows the graph of the same dataset build by ϵ -NN with an ϵ value of 15%. The noisy examples still colored in red and present a low degree value.

Figure 3.1: Building the graph for an artificial dataset.

When a dataset has a large amount of noise, a larger number of examples will have a low degree value and the threshold value can be higher. On the other hand, for datasets with a lower noise level, a lower threshold value can be required. Otherwise, many safe examples will be regarded as noisy. Due to the difficulty in selecting a specific threshold value, we used the N3 measure to estimate the percentage of noise in the dataset. This

was the most correlated measure to the noise levels in our experiments and for which clearer limits on the values obtained for distinct noise levels can be observed (Figure 2.5).

Therefore, in GNN we first order all examples according to their degree values. Afterwards, the N3 value delimits how much of the examples of lower degree can be regarded as noisy. Furthermore, among the examples with a degree lower than the threshold, only those that are misclassified by the NN classifier used in N3 are considered noisy. This polling allows more robustness for maintaining safe examples.

Figure 3.2 illustrate the GNN filter for the artificial dataset described in Figure 3.1. Figure 3.2(a) shows the original graph with the noisy examples signalized as potentially noisy by the N3 measure. The N3 measure was more lanky, pointing six safe examples as noisy, but misclassifying only one noisy example. Sorting the graph degree of the vertices, Figure 3.2(b) shows the nine examples with lowest degree. The difference between the original noisy examples and those with lowest degree is of five examples. All of them are safe examples pointed as noise. When we combine the prediction of both degree and N3 measures by a consensus voting we have the results of Figure 3.2(c), which corresponds to the output of the GNN filter. In this case only two noisy examples are misclassified as safe.

3.1.3 Distance Based Noise Filters

Some popular NF techniques are based on the distance between examples and employ the k -NN algorithm (Wilson, 1972; Wilson & Martinez, 2000; Tomek, 1976). They consider an example to be consistent if it is close to other examples from its class. Otherwise, it is either probably incorrectly labeled or in the decision border. In the later case, the example is also considered unsafe, since small perturbations in a borderline example can move it to the wrong side of the decision border. Therefore, the filters based on distance usually remove both noisy and borderline examples. This tends to increase the margin of separation between different classes.

The Edited Nearest Neighbor (ENN) (Wilson, 1972) technique removes an example if the majority label of its k -NN differs from its own label. Repeated Edited Nearest Neighbor (RENN) is a variation of ENN, applying ENN repeatedly until all objects have the majority of their neighbors of the same class. The *All- k -Nearest Neighbor* (AENN) technique applies the k -NN classifier with several increasing values of k (Tomek, 1976). At each iteration, examples that have the majority of their neighbors from other classes are marked as noisy. Algorithm 5 shows the AENN filter. The input are the training data (E), the testing data (T), the label of the testing data (Y) and the maximum k number of NNs. The output is the noisy examples subset (A). For a given sample i and a given j value with interval from 1 to k , the NN classifier is evaluated and the prediction saved in the prediction vector $P_{i,j}$. After the evaluation for all values of k , the *majority* voting

(a) The 9 examples signalized by N3 as noisy.

(b) The 9 examples with lowest graph degree signalized as noisy.

(c) Intersection of the N3 and the graph degree predictions.

Figure 3.2: Noise detection by GNN filter.

strategy function compare the label (Y) with the prediction. If the majority of the models classified the sample as noisy, it is added in the critical subset.

Algorithm 5 AENN

Input: E (training data) T (testing data) Y (testing data class) k (number of nearest neighbors)
Output: A (critical example set)

```

 $A \leftarrow \emptyset$ 
for  $i \leftarrow 1, \dots, |T|$  do
  for  $j \leftarrow 1, \dots, k$  do
     $P_{i,j} \leftarrow NN(E, T_i, j)$ 
  end for
  if  $\text{majority}(P_i) \neq Y_i$  then
     $A \leftarrow A \cup T_i$ 
  end if
end for
return  $A$ 

```

3.1.4 Other Noise Filters

There are many other NF techniques in the literature (Verbaeten & Assche, 2003; Khoshgoftaar & Rebours, 2004; Sáez et al., 2015; Garcia et al., 2015; Sáez et al., 2016). The Cross-validated Committees Filter (CVCF) algorithm, proposed in Verbaeten & Assche (2003), induces a classification model using 10-fold-cross-validation. Examples from the training fold wrongly classified by this model are considered as potential noise. The number of times an example is marked as noisy is used to assess its reliability. If the example was marked as noisy most of the times, CVCF will consider the example to be noisy.

Khoshgoftaar & Rebours (2004) proposed the Iterative-Partitioning Filter (IPF), which induces DTs models in an iterative process using the training data divided according to cross-validation. The iterative process finishes when less than 1% of the data is not misclassified by the DTs after the third iteration. Sáez et al. (2015) combined the Synthetic Minority Over-sampling Technique (SMOTE) and the IPF filter to propose the SMOTE-IPF filter. This filter focuses on searching noisy examples in imbalanced datasets.

In Sáez et al. (2016) a framework for noise detection called Iterative Noise Filter based on the Fusion of Classifiers (INFFC) is used to detect noisy examples. The idea is very similar to that presented in Sluban et al. (2014), where the information gathered from different classifiers are combined. The main difference between the papers is the iterative process with multiple classifiers. First, a preliminary classifier is performed and noisy examples are filtered. Then, another classifier is built from the examples that are not identified as noisy in the preliminary filtering. Finally, a noise sensitivity process is applied in order to select the noise examples.

All previous filters adopt a *crisp decision* in noise identification, classifying each training example either as noisy or safe. Next section deals with the slightly modified problem of noise ranking, where the examples of the dataset are ordered according to an estimate of their unreliability level (Lorena et al., 2015).

3.2 Noise Filters: a Soft Decision

When standard filters are employed in noise detection, a hard decision is obtained of whether an example is noisy or not. In soft decision filters, the objective is to order a dataset according to the (un)reliability level of its examples. This reliability, called as Noisy Degree Prediction (NDP), can be estimated by different strategies. An example that contains core knowledge for pattern discovery should be evaluated as highly reliable, while those examples that do not follow the general patterns of the dataset should be considered unsafe. Obtaining such NDP value can be considered interesting for various reasons. One of them is to evidence the most problematic examples in a dataset. These instances can then be further examined by a domain specialist and increase data understanding.

Knowing which are the most problematic examples can also support the development of new noise tolerant ML techniques. In Smith et al. (2014), for example, an estimate of instance hardness is used to adapt the training algorithm of an Artificial Neural Network (ANN), so that hard instances have a lower weight on the back-propagation error function update. The same authors consider noisy instances as hard and design a new filter based on their instance hardness measure. This measure considers an instance hard if it is misclassified by a diverse set of classification algorithms. This is also the assumption of most ensemble-based filters in noise identification.

A noticeable related work in noise ranking is Sluban et al. (2014), where an ensemble of noise detection algorithms included in a tool called NOISERANK was applied to a medical coronary heart disease dataset. Interestingly, the top-ranked instances were either incorrectly diagnosed patients or worth noting *outlier* cases. NOISERANK takes into account the agreement level of different filters in pointing an example as noisy. In this work we employ a different approach and adapt the output of each individual filter for obtaining a NDP value.

The NF techniques whose outputs are adapted for a soft decision are HARF, SEF, DEF, PruneSF and AENN. Although there are many other filters in the literature, those chosen here are well-known representatives of different NF categories and have different bias. They were adapted to provide an estimate of the NDP of an example being noisy. These NDP values can then be employed for ranking the examples in a dataset, such that top-ranked instances will be those most unreliable and probably noisy.

For the ensemble based techniques SEF and DEF, we estimate the NDP as the percentage of disagreement between the predictions of the classifiers combined. Given an example,

each classifier outputs a confidence regarding its noise presence prediction. These values are averaged to obtain the final NDP value for the example. For HARF the NDP of an example is given by the percentage of base trees that disagree on their predictions for that particular instance. This is equivalent to eliminate the threshold level of HARF.

In the case of PruneSF, we have two steps. Firstly all examples pruned by the initial DT induced are equally ranked first, that is, they are assigned a probability of 1 of being noisy. Next, the examples are ranked according to their CLCH values, that give the confidence estimate. The CLCH values are also normalized to give a probability estimate.

In the case of AENN, first a gaussian kernel function based on the k -NN of an example is used to estimate its NDP at each iteration of the k -NN (from $i = 1$ to k). The final NDP value of an example is the average of the probability values obtained during the AENN iterations.

The GNN technique was an experimental filter proposed to show the ability of the measures investigated in Chapter 2 and was not adapted to soft decision. The main motivation for this was the high computational cost, which could compromise the execution for a range of datasets since the N3 measures used in the technique involves inducing multiple k -NN classifiers with leave-one-out. Even this, a possible soft GNN version could be the use of a gaussian kernel function based on the k -NN to calculate the N3 measure. The average of the graph degree and the probability values obtained by the 1-NN odds would be the NDP of the examples. It is important to reinforce that this adaptation still is highly costly.

Like in classification tasks, more robust decisions in noise identification can be obtained by combining outputs from diverse NF techniques (Brown, 2010). Committees of filters with different bias, can increase the noise detection accuracy for a larger number of datasets. Thus, this work also combined the previous filters into ensembles. A simple approach was adopted, where these ensembles combine the NDP values estimated by the individual techniques, taking their average.

3.3 Evaluation Measures for Noise Filters

In order to properly evaluate the performance of NF techniques in noise detection, it is necessary to know in advance which are the noisy instances. Using this knowledge, Sluban et al. (2014) proposed a methodology to evaluate the efficacy of the filters. In this methodology, the well-known precision, recall and F_β -score metrics can be used to assess the filters performance. These metrics use a confusion matrix, as that illustrated in Table 3.1. This table contains the number of examples correctly and incorrectly identified as noisy or clean by a given filter, where: TP is the number of noisy examples correctly identified, TN is the number of correct clean examples, FP is the number of clean examples incorrectly identified as noisy and FN is the number of noisy examples disregarded by

the filter.

Table 3.1: Confusion matrix for noise detection.

Predicted/Real	Noisy	Clean
Noisy	TP	FP
Clean	FN	TN

From the confusion matrix, precision and recall can be calculated. Precision (Equation 3.1) is the percentage of noisy cases correctly identified among those examples identified as noisy by the filter. Recall (Equation 3.2) is the percentage of noisy cases correctly identified among the noisy cases present in the dataset.

$$precision = \frac{TP}{TP + FP} \quad (3.1)$$

$$recall = \frac{TP}{TP + FN} \quad (3.2)$$

The F_β -score metric combines precision and recall values, as presented in Equation 3.3. Considering $\beta = 1$ we have a harmonic mean where precision and recall have the same importance. Sluban et al. (2014) used $\beta = 0.5$, giving more importance to precision than to recall. The authors state that precision should be preferred in noise identification such that the noisy cases identified are indeed noise. All measures range from 0 to 1 and higher values indicate a better performance in noise detection by a filter.

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall} \quad (3.3)$$

The previous measures can be used when there is a hard decision of classifying an example as noisy. For rankers, where a soft decision is obtained, other strategies should be employed instead. They should take into account the ordering produced, such that better values are obtained if noisy instances are top-ranked, while clean examples are bottom-ranked.

A simple adaptation of the previous evaluation measures is the application of a threshold to the number of top-ranked examples that will be regarded as noisy (Schubert et al., 2012). Afterwards, the precision, recall and F_β values are recorded. These measures are named here $p@n$, $r@n$ and $F_\beta@n$, where n is the number of top-ranked examples that are considered noisy (Schubert et al., 2012; Craswell, 2009). For setting the n value to be employed, which is the number of top-ranked examples to be considered noisy, we use the same approach as Schubert et al. (2012), where n is set as the known number of noisy instances in a dataset. In this case, we have $p@n = r@n = F_\beta@n$, since a noisy example misclassified will be replaced by a clean example, increasing both false positive and false negative rates by one unit. Therefore, the precision for the top-ranked instances (Equation 3.4) in noise detection is then defined as the number of correctly identified noisy cases

($\#correct_noisy$) divided by the number of examples identified by the filter as noisy (the threshold n):

$$p@n = \frac{\#correct_noisy}{n} \quad (3.4)$$

Based on an evaluation measure proposed for feature ranking in Spolaôr et al. (2013), we presented an evaluation measure named Noise Ranking Area Under the ROC Curve (NR-AUC), which is independent of a particular threshold value (Lorena et al., 2015). Given an ordering of the examples, first a ROC-type graph is built, which considers the true positive rate (TPR) and false positive rate (FPR) in noise prediction. Next, the area under the plotted curve is calculated. NR-AUC values range from 0 to 1, where higher values indicate a better performance, while values close to 0.5 are associated to a random noise identification performance.

As an example, consider an artificial dataset where there are five known noisy cases and 15 clean examples. A given noise ranker produces the ordering: $N_1, N_2, C_1, C_2, N_3, N_4, C_3, C_4, N_5, C_5, \dots, C_{15}$, where N stands for a noisy example and C for a clean example. It is possible to observe that the third example in the list is clean but it is between examples that are top-ranked as noisy. The adapted ROC graph obtained for this example is shown in Figure 3.3. Each time a noisy case is observed, a TP value is accounted and the curve grows one unit at the TPR axis. When a clean example is found, a FP value is accounted and the curve grows one unit at the FPR axis. NR-AUC can then be calculated as the number of unit squares bellow the curve, normalized by the total number of squares.

Figure 3.3: Example of NR-AUC calculation

The NR-AUC of Figure 3.3 is equal to $67/(5 * 15) = 0.8933$. The $p@n$ of the same noise ranker is 0.6, once $\#correct_noisy = 3$ and $n = 5$. The main advantage of using the NR-AUC is to avoid the bias of the selection of a specific n threshold value in $p@n$.

3.4 Evaluating the Noise Filters

This section presents the experiments performed to evaluate the previous NF techniques in the presence of label noise for several benchmark public datasets. First different levels of label noise were added to each dataset. We then monitor the performance of each filter. This is accomplished by:

1. Evaluating the overall performance of the crisp and soft NF techniques in noise identification, as well as their behavior per noise level. The first analysis considers the average of the performance of each filter over all noise levels in a dataset. The objective in this case is to identify the filters which are more robust in noise identification. The second analysis considers the performance of the filters for each specific noise level.
2. Comparing the performance of individual soft filters and several ensemble of these filters. This analysis allows identifying a subset of ensembles which increase the noise detection accuracy for a larger number of datasets than the individual techniques used alone. For evaluating the efficacy of the filters, measures which take into account the noise orderings produced are used.

For the sake of generality, the proposal will be evaluated using five different up-to-date NF techniques, which are well-known representatives of the field and present different biases (Frenay & Verleysen, 2014). They are HARF, SEF, DEF, AENN and PruneSF. The GNN filter was also used in the crisp NF analysis. This algorithm was omitted from the soft decision analysis as its adaptation to provide a NDP value can be considered costly.

Next, we detail the experimental protocol previously outlined.

3.4.1 Datasets

All techniques are evaluated in noisy versions of the datasets from Table 2.2 and created by using the random noise imputation as described in Chapter 2, Section 2.1. For each dataset, random noise was added at rates of 5%, 10%, 20% and 40%. For each dataset and noise level, 10 different noisy versions were generated, resulting in 3600 datasets with class noise. Noise injection was thus controlled to allow the recognition of the noise cases and the assessment of their identification by the NF techniques.

3.4.2 Methodology

The crisp NF techniques were evaluated using the F_β -score with $\beta = 1$, which gives the same importance to precision and to recall performance values in the identification

A correlation analysis of the predictions of pairs of soft NF techniques allows identifying their similarities. The partitions produced by complete linkage over the predictions illustrates the similarity between all filters. A dendrogram can be obtained and used to identify the similarity between NF techniques regarding their predictive performances. Our objective with this analysis is to support the selection of the filters that should be further investigated, which are namely those NF with the best predictive performance and higher performance diversity.

3.5 Experimental Evaluation of Crisp Filters

This section presents the experimental results obtained for the crisp NF techniques when evaluated by the F_1 measure. Section 3.5.1 reports the overall ranking of the performance obtained by each filter in all the datasets, despite the noise level introduced. Section 3.5.2 presents the average predictive performance obtained by each filter for each specific noise level.

3.5.1 Rank analysis

Figure 3.4 summarizes the F_1 predictive performance for all NF techniques. It shows the average ranking of each filter, regarding its predictive performance for all datasets, independently of the noise level introduced. Each value in the x -axis represents one filter. The y -axis shows the average and the standard deviation for the ranking of each NF technique. The filter with the best predictive performance will have the lowest average (and standard deviation spread) ranking values.

Figure 3.4: Ranking of crisp NF techniques according to F_1 performance.

According to Figure 3.4, the DEF was the best performing filter. HARF comes next, followed by SEF. PruneSF, GNN and AENN were the worst performing filters. The best filter had an F_1 average predictive performance of 0.5823. It is also interesting to notice that all filters showed a high standard deviation. Since the graph joins the results obtained for various datasets and noise levels, this can be expected. For instance, some filters may be better for some noise levels or datasets with specific characteristics.

3.5.2 F_1 per noise level

The previous analysis on the average F_1 performance hinders the behavior of the techniques for specific noise levels. Figures 3.5 and 3.6 show the F_1 predictive performance achieved by the NF techniques in each dataset, for each noise level. The x -axis represents the noise levels while the y -axis shows the F_1 for each noise level. HARF is shown by black dots, SEF by red triangles, DEF by blue squares, AENN by green crosses, GNN by purple hollow squares with crosses inside and PruneSF by orange asterisk.

For the datasets *abalone*, *blood-transfusion-service*, *breast-tissue-4class*, *breast-tissue-6class*, *bupa*, *cmc*, *dbworld-subjects*, *glioma16*, *habermans-survival*, *heart-cleveland*, *heart-repro-hungarian*, *heart-va*, *indian-liver-patient*, *meta-data*, *monks2*, *pima*, *planning-relax*, *saheart*, *spect-heart*, *statlog-german-credit*, *tae*, *wholesale-region* and *yeast*, the average predictive performance for the best filter is lower than 0.5. This represents a poor predictive performance. For the datasets *acute-nephritis*, *acute-urinary*, *banknote-authentication*, *car*, *dermatology*, *page-blocks*, *qualitative-bankruptcy*, *segmentation*, *wine* and *zoo*, on the other hand, the average performance for almost all noise levels for the best filter is higher than 0.9, which is a high accuracy rate.

Looking at the other datasets with low noise rates, like 5% and 10%, the best filter is HARF with a F_1 average predictive performance of 0.6210. PruneSF comes next with 0.5294, followed by DEF with 0.5225. AENN, GNN and SEF were the worst performing filters. For high noise rates, like 20% and 40%, the best filter is DEF with average 0.69 of F_1 value. SEF comes next with average of 0.6430. The other filters had the worst average performance. For low noise rates, the few datasets where HARF, PruneSF and DEF filters did not achieve a good predictive performance were *crabs*, *expgen*, *mines-vs-rocks*, *movement-libras*, *parkinsons*, *vowel* and *vowel-reduced*. For high noise levels, the datasets where DEF and SEF did not show a good performance were *cardiotocography*, *collins*, *flags*, *flare*, *glass*, *hayes-roth*, *led7digit*, *mammographic-mass*, *monks3*, *movement-libras*, *titanic*, *user-knowledge*, *vehicle*, *vowel*, *vowel-reduced*, *waveform-5000* and *wine-quality-red*.

Figure 3.7 summarizes the rank of the NF techniques for all datasets for each noise level. The HARF filter was the best performing filter for 5% and 10% of noise levels and the DEF filter was the best filter for 20% and 40% of noise levels. While the filters DEF and SEF increased the ranking performance, HARF, PruneSF and AENN decreased the

Figure 3.5: F_1 values of the crisp NF techniques per dataset and noise level.

Figure 3.6: F_1 values of the crisp NF techniques per dataset and noise level.

ranking performance for higher noise levels. Therefore, the later techniques tend to be less robust to high levels of label noise.

Figure 3.7: Ranking of crisp NF techniques according to F_1 performance per noise level.

Using the Friedman statistical test with the Nemenyi post-test at 95% of confidence level (Demšar, 2006), the following results can be reported for each noise level:

- 5% of noise level: HARF was better than SEF, DEF, AENN, GNN and PruneSF. The DEF and PruneSF techniques were better than SEF and GNN. The AENN was better than the SEF.
- 10% of noise level: HARF and DEF were better than SEF, AENN, GNN and PruneSF. PruneSF was better than GNN.
- 20% of noise level: DEF was better than HARF, SEF, AENN, GNN and PruneSF. The HARF, SEF and PruneSF techniques were better than AENN and GNN.
- 40% of noise level: DEF and SEF were better than HARF, AENN, GNN and PruneSF. The HARF, GNN and PruneSF techniques were better than AENN.

Considering the combined results of the F_1 performance illustrated in Figure 3.7 and of the statistical tests performed, the HARF filter was able to improve the F_1 values for low noise rates, while DEF was able to improve performance for high noise rates. The SEF technique was the worst NF technique for low noise rates, while AENN was the worst technique for high noise rates. The GNN technique showed worst results for intermediate noise rates.

Therefore, the choice of a particular filter can be dependent on the expected noise level of a particular dataset. Based on this information, DEF should be preferred when a high noise level is expected, while HARF should be employed when the noise level is low. But the characteristics of the datasets can also influence in the results obtained, since each filtering technique has a bias that can fit specific cases more properly. This motivates the use of MTL in the domain of label noise identification, as we describe in Chapter 4.

3.6 Experimental Evaluation of Soft Filters

This section presents the experimental results obtained for the soft NF techniques. As in the analysis of crisp filters, Section 3.6.1 reports the overall ranking of the techniques regarding $p@n$ performance for all noise levels. A similarity analysis of the NF techniques is also performed. It allows to identify the most diverse soft filters among those tested here. This analysis was performed because of the high number of soft filters being compared. Section 3.6.2 presents the average predictive performance obtained by each chosen NF technique for each noise level using $p@n$, while Section 3.6.3 presents the NR-AUC performance per noise level.

3.6.1 Similarity and Rank analysis

Figure 3.8 summarizes the $p@n$ predictive performance for all soft NF techniques (individual and ensembles). It shows the average ranking of each filter, regarding its predictive performance for all datasets, independently of the noise level introduced. Each value in the x -axis represents one filter. The y -axis shows the average and the standard deviation for the ranking of each filter. The individual NF techniques have their names highlighted in bold in the figure, while ensembles are not highlighted.

It is possible to observe in Figure 3.8 that only some ensembles improved the performance compared to the individual NF techniques. The best ensembles were E2, E11, E13, E21, E22 and E26. Some of them also decreased the standard deviation of the results across different datasets. This is the case of E26, for example. The best individual filter was DEF, while HARF presented an intermediate ranking, but both showed a high standard deviation. The AENN and PruneSF filters were the worst ranked techniques.

It must be observed that, although the best $p@n$ performance was obtained by the ensembles, they have a higher computational cost than the individual NF techniques. Moreover, the best technique, E2, had an average $p@n$ predictive performance of 0.67. Thus, there is still room for improvements. An alternative to improve the predictive performance would be to look for filters that are among those techniques with the best predictive performance, and make different misclassifications.

Figure 3.9 shows a dendrogram presenting the similarity of the predictions made by the

Figure 3.8: Ranking of soft NF techniques according to $p@n$ performance.

NF techniques. The dendrogram was obtained by running a complete-linkage clustering algorithm. The algorithm used a Euclidean distance of the correlation vectors of the filter predictions. In this dendrogram, lower branches in the hierarchy (y -axis) represent low similarity and higher branches represent high similarity. The proximity of NF techniques in the x -axis is related with their similarity degree. The names of the individual filters are highlighted in bold.

It is possible to observe in Figure 3.9 that the predictive performance of the individual NF techniques are more dissimilar than that of the ensembles. The least similar filters are AENN and PruneSF, followed by HARF and the two filters based on ensembles of classifiers (DEF and SEF). The NF ensembles with the highest similarity are those combining four or five filters, like ensembles E21 to E26. In intermediate branches, pairs of ensembles like E2 and E10, E5 and E7, E1 and E20, do not share any individual filter as base component. Ensembles E16 and E19 do not share any individual filters, except for PruneSF. The most promising pairs of NF alternatives are those that present the lowest similarity and are contained in most different branches, since they present good predictive performance and identify diverse noisy examples.

The combination of the results from Figures 3.9 and 3.8 makes it easier to select ensembles that showed good predictive performance in noise identification and a low similarity among each other. This is done to increase the diversity of the ensembles while maintaining a good performance in noise detection. According to these combined results, the ensembles E1, E2, E11 and E21 were selected for further analysis. E2 was selected because it is the best filter regarding predictive performance and it also shows to be more diverse. Ensembles E11 and E13 had a high similarity between each other, so ensemble E11 was selected as a representative. The same happened to ensembles E21, E22 and E26.

Figure 3.9: Dissimilarity of filters predictions.

In this case, ensemble E21 was selected. E1 was preferred to E3 since it achieved best predictive performance. Regarding the individual filters, HARF and DEF were selected because they are the more accurate and diverse among the individual filters.

3.6.2 $p@n$ per noise level

This analysis will consider the average predictive performance achieved by each previously chosen NF techniques, for specific noise levels. Figures 3.10 and 3.11 show the $p@n$ predictive performance of the best filters for all datasets and for each noise level. Each value in the x -axis represents one filter. The y -axis shows the $p@n$ for each noise level. HARF is shown in black with solid circles, DEF in red with solid triangles, E1 in blue with solid squares, E2 in green with crosses, E11 in purple with hollow squares and crosses inside, E21 in orange with asterisk. The last plot summarizes the ranking of the $p@n$ values for each noise level considering all datasets.

For the datasets *blogger*, *blood-transfusion-service*, *breast-tissue-4class*, *breast-tissue-6class*, *bupa*, *cmc*, *habermans-survival*, *heart-va*, *indian-liver-patient*, *meta-data*, *monks2*, *pima*, *planning-relax*, *saheart*, *spect-heart*, *statlog-german-credit*, *tae*, *titanic* and *wholesale-region*, the average predictive performance for the best filter is lower than 0.5. This represents a poor predictive performance. For the datasets *acute-nephritis*, *acute-urinary*, *banknote-authentication*, *car*, *collins*, *dermatology*, *expgen*, *newthyroid*, *page-blocks*, *qualitative-*

Figure 3.10: $p@n$ values of the best soft NF techniques per dataset and noise level.

Figure 3.11: $p@n$ values of the best soft NF techniques per dataset and noise level.

bankruptcy, *segmentation*, *thyroid-newthyroid*, *vowel*, *vowel-reduced*, *wine* and *zoo*, on the other hand, the average performance for almost all noise levels is higher than 0.9, which is a high accuracy rate. The datasets pointed as having performance lower than 0.5 and higher than 0.9 are mostly the same from section 3.5.2. This indicates a high correlation between the evaluation measures.

Looking at the other datasets with low noise rates, like 5% and 10%, the best filter is E11 with a $p@n$ average predictive performance of 0.65. E2 comes next with 0.6482 and followed by E1 with 0.64. The best original filter is HARF with a $p@n$ of 0.64. The worst performing filter is DEF with $p@n = 0.63$. For high noise rates, like 20% and 40%, the best filters are E2 with average $p@n = 0.70$ and E11 with $p@n = 0.69$. DEF comes next with $p@n = 0.68$. The worst performing filter is E1. For low noise rates, the individual NF techniques perform better in some datasets, like *appendicitis*, *banana*, *horse-colic-surgical*, *ionosphere*, *mammographic-mass*, *molecular-promotor*, *waveform-5000* and *wine-quality-red*. For high noise rates, like 20% and 40%, the ensembles achieved the best predictive performance, except for the datasets *appendicitis*, *backache*, *banana*, *breast-cancer-wisconsin*, *climate-simulation*, *colon32*, *flags*, *heart-cleveland*, *led7digit*, *mines-vs-rocks*, *ringnorm*, *spectf-heart*, *waveform-5000*, *wine-quality-red* and *yeast*. In four of these datasets, the original filters presented a better predictive performance than the ensemble filters for all noise levels.

A similar analysis is summarized in the Figure 3.12, which presents the average rank of the NF techniques in all datasets by noise level. The ensembles E11 and E2 were the best for all noise rates. The original filters had the worst ranking for low noise rates and an intermediate ranking for high noise rates. While the filters E2, DEF and HARF increased their performance for higher noise rates, E1 and E21 decreased their ranking performance. Once the E11 and E2 are composed by HARF and DEF, it is possible that the ensembles took advantage of the good performance of HARF for low noise levels and of DEF for high noise levels to increase the performance in all noise levels.

Using the Friedman statistical test with the Nemenyi post-test at 95% of confidence level (Demšar, 2006), the following results can be reported for each noise level:

- 5% of noise level: E2 and E11 were better than HARF and DEF. Ensemble E21 was better than DEF. The best ensemble was better than the best individual NF technique.
- 10% of noise level: E2 was better than DEF, E1 and E21. E11 was better than HARF, DEF, E1 and E21. There was no difference between the best ensemble and the best individual NF technique.
- 20% of noise level: E2 and E11 were better than HARF, DEF, E1 and E21. The best ensemble was better than the best individual NF technique.

Figure 3.12: Ranking of best soft NF techniques according to $p@n$ performance per noise level.

- 40% of noise level: ensemble E2 was better than HARF, DEF, ensembles E1 and E21. The filter HARF and the ensemble E11 were better than E1 and E21. The filter DEF was better than ensemble E1. There was no difference between the best ensemble and the best individual NF technique.

Considering the results illustrated in Figure 3.12 and of the statistical tests performed, the ensembles E2 and E11 were able to improve the $p@N$ values for almost all noise levels, when compared to the individual filters HARF and DEF. An interesting point to be considered is the committee difference between E2 and E11. While E2 is composed by the two best original filters HARF and DEF, E11 also use the SEF filter.

Table 3.3 compares the best individual NF technique with the best ensemble NF technique for all datasets. It shows how often each technique won and when a tie occurred. For all noise levels, in a large number of datasets, the best individual filter presented a predictive performance similar or better than the best ensemble. It is interesting to notice that when the difference between the individual filter and the ensemble is the largest, the number of ties is also the largest. These results show that none of these two alternatives alone would be a good choice. The ideal situation would be to recommend, for each dataset, the best of these two alternatives.

Taking into account that the computational cost of the individual filter is lower than that of the ensembles, when the predictive performance of an individual NF technique filter is better than or similar to the performance of an ensemble, the individual filter

Table 3.3: Percentage of best performance for each noise level.

Noise level	Ensemble	Individual	Tie
5%	61%	25%	14%
10%	50%	41%	9%
20%	62%	37%	1%
40%	51%	48%	1%

should be preferred. The ideal situation would be to recommend, for each dataset, the best of these two alternatives. The use of a recommendation system based on MTL to choose, for a new dataset, between the best ensemble and the best individual filter could not only improve the noise detection predictive performance for the cases where the individual filter already has a good performance, but also decrease the overall filtering computational cost.

3.6.3 NR-AUC per noise level

This analysis will consider the average predictive performance achieved by each of the previously chosen soft NF techniques using the NR-AUC measure, for each noise level. This measure allows a ranking analysis independent of a specific threshold in the number of examples regarded as noisy. Figures 3.13 and 3.14 show the NR-AUC values obtained by the soft filters for each noise level. Each value in the x -axis represents one filter. The y -axis shows the NR-AUC for each noise level. The filters are shown using the same labels from Figure 3.10.

For almost all cases, the performance degrades for higher levels of label noise. Therefore, ranking results were highly affected by the noise level present in the datasets. The *meta-data* dataset is the only one with predictive performance for almost all noise levels lower than 0.5. This represents a random predictive performance. For the datasets *acute-nephritis*, *acute-urinary*, *balance*, *banana*, *banknote-authentication*, *breast-cancer-wisconsin*, *cardiotocography*, *car*, *climate-simulation*, *collins*, *dermatology*, *expgen*, *flare*, *glass*, *hayes-roth*, *ionosphere*, *iris*, *kr-vs-kp*, *led7digit*, *monks1*, *monks3*, *movement-libras*, *newthyroid*, *page-blocks*, *parkinsons*, *phoneme*, *qualitative-bankruptcy*, *ringnorm*, *seeds*, *segmentation*, *thyroid-newthyroid*, *tic-tac-toe*, *user-knowledge*, *vehicle*, *vertebra-column-3c*, *voting*, *vowel*, *vowel-reduced*, *waveform-5000*, *wdbc*, *wholesale-channel*, *wine*, *wine-quality-red*, *yeast* and *zoo*, on the other hand, the performance for the best filter for almost all noise levels is higher than 0.9, which is a very high NR-AUC rate. The dataset pointed as having performance lower than 0.5 is also signed by the $p@n$ evaluation measure. The main difference between the results is the number of datasets considered with low performance. For performance higher than 0.9, the number of datasets signed by the NR-AUC is higher than but also including the datasets pointed in Section 3.6.2.

Looking at the other datasets with low noise rates, like 5% and 10%, the best filters

Figure 3.13: NR-AUC values of the best soft NF techniques per dataset and noise level.

Figure 3.14: NR-AUC values of the best soft NF techniques per dataset and noise level.

are E2, E11 and E21 with a NR-AUC average predictive performance of 0.86. The best original filter is HARF with NR-AUC = 0.86. The worst performing filter is DEF with NR-AUC = 0.85. For high noise rates, like 20% and 40%, the best filters are E2, E11 and HARF with average NR-AUC = 0.76. The worst performing filter is DEF with NR-AUC = 0.75. For low noise rates, like 5% and 10%, the individual NF techniques perform better in some datasets, like *abalone*, *blood-transfusion-service*, *breast-tissue-6class*, *bupa*, *dbworld-subjects*, *heart-hungarian*, *heart-va*, *hepatitis*, *mammographic-mass*, *molecular-promoters*, *molecular-promotor*, *planning-relax*, *saheart*, *spectf-heart*, *thoracic-surgery* and *wholesale-region*. For high noise rates, like 20% and 40%, the ensembles achieved the best predictive performance, except for the datasets *abalone*, *appendicitis*, *backache*, *blogger*, *blood-transfusion-service*, *breast-tissue-6class*, *flags*, *heart-repro-hungarian*, *heart-va*, *horse-colic-surgical*, *mines-vs-rocks*, *planning-relax*, *spectf* and *spectf-heart*. Considering all noise levels, for six of these datasets the original filters presented a better predictive performance than the ensemble filters.

Figure 3.15 summarizes the rank of the NF techniques for all datasets for each noise level. The ensembles E11 and E2 were the best for all noise rates with a better performance for E11 in low noise levels and a better performance for E2 in high noise rates. The original filters had the worst ranking results for low noise rates and an intermediate ranking for high noise rates. While the filters E2, DEF and HARF increased their performance for higher noise levels, E1 and E21 decreased their ranking performance.

Figure 3.15: Ranking of best soft NF techniques according to NR-AUC performance per noise level.

Using the Friedman statistical test with the Nemenyi post-test at 95% of confidence level (Demšar, 2006), the following results can be reported regarding each noise level:

- 5% of noise level: E11 was better than HARF, DEF and Ens1. The filters HARF, E1, E2 and E21 were better than DEF. The best ensemble was better than the best individual NF technique.
- 10% of noise level: E2 and E11 were better than HARF, DEF, E1 and E21. HARF, E1 and Ens21 were better than DEF. The best ensemble was better than the best individual NF technique.
- 20% of noise level: E2 and E11 were better than HARF, DEF, E1 and E21. The best ensemble was better than the best individual NF technique.
- 40% of noise level: ensemble E2 was better than HARF, DEF, E1 and E21. The ensemble E11 was better than DEF, E1 and E21. The filter HARF was better than E1 and E21. The filter DEF was better than ensemble E21. There was no difference between the best ensemble and the best individual NF technique.

Considering the results illustrated in Figure 3.15 and of the statistical tests performed, the ensembles E2 and E11 were able to improve the NR-AUC values for almost all noise levels, when compared with HARF, DEF and E1. When the best ensemble is compared with the best individual filter, the ensembles are better for all noise levels, except for 40%.

Therefore, when the results from Section 3.6 are combined to those from the NR-AUC analysis, some main differences can be signalized. While for 19 datasets the $p@n$ average performance was lower than 0.5, only one of these are signalized as bad by NR-AUC. The same happens with the datasets with intermediate $p@n$ performance, which are mostly classified as presenting a high NR-AUC performance. Although, when the performance of the filters are compared, the results are similar. The main difference from Figures 3.12 and 3.15 are the small improvements of the rankings of the ensembles E11 and E2. For low noise rates, the difference between their rankings also increased.

All these facts are related with the main characteristics of the NR-AUC measure, which is interested not only in the top-ranked noisy examples, but also in the correct prediction of the safe examples. For a real problem, if the percentage of potential noisy examples is low and the removal of noise is the goal, the use of the $p@n$ could be a better choice to evaluate the noise performance. If the analysis are also interested in the safe examples and the NDP rates are fuzzy, NR-AUC can be a better performance measure for the NF techniques.

3.7 Chapter Remarks

This chapter presented and analyzed the performance of well-known crisp NF techniques. We also adapted most of these filters for soft decision and investigated how noise detection could be improved by using an ensemble of NF techniques. The techniques were evaluated using a large set of public datasets from UCI repository (Lichman, 2013) with different levels of artificial imputed noise.

The experimental results related with the evaluation of crisp NF techniques showed a good performance of HARF and DEF techniques in certain cases. While HARF had a higher performance for low noise rates, DEF had a increased of performance for high noise levels. Other filters like PruneSF and SEF also presented good performance. Therefore, the choice of a particular filter can be dependent on the expected noise level of a particular dataset.

The experimental results related with the evaluation of soft NF techniques improved the identification of noisy examples in a set of datasets. The use of ensembles of NF was also another contribution which increased the performance. The ensembles E11 (composed by HARF, DEF and SEF) and E2 (composed by HARF and DEF) were the best for all noise rates. They were also evaluated with different metrics, including a measure based on ROC-type analysis (NR-AUC) which allows a ranking analysis independent of one specific threshold for noise identification.

This chapter was based on the following papers produced in this work:

- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2015). “Effect of label noise in the complexity of classification problems”. *Neurocomputing*, 160:108 - 119.
- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2012). “A study on class noise detection and elimination”. *Brazilian Symposium on Neural Networks (SBRN)*, 13 - 18.
- Lorena, A. C., Garcia, L. P. F., & de Carvalho, A. C. P. L. F. (2015). “Adapting Noise Filters for Ranking”. *Brazilian Conference on Intelligent Systems (BRACIS)*, 299 - 304.

Chapter 4

Meta-learning

In ML, bias has been defined as the choice of a specific generalization hypothesis over several other possible generalizations, restricting the searching space (Wolpert, 1992; Mitchell, 1997). Due to the lack of exact knowledge about the real data distribution, when deciding which technique has the most adequate bias for a new dataset, several algorithms need to be tried. This process, known as trial and error, is laborious and subjective. An alternative to support the automatic selection of technique is the use of Meta-learning (MTL) (Brazdil et al., 2009). By using knowledge from the previous application of the available algorithms to several datasets, it is possible to induce a meta-model able to recommend the most suitable technique for a new dataset.

Brazdil et al. (2009) define MTL as the study of methods that explore metaknowledge in order to improve or to obtain more efficient ML solutions. It is worth noting that MTL has been applied not only for the recommendation of ML algorithms. MTL has also been used for the recommendation of techniques and approaches for: data classification (Brazdil et al., 2009), optimization (Kanda et al., 2011), time series analysis (Rossi et al., 2014), gene expression tissue classification (de Souza et al., 2010), regression (Soares et al., 2004), SVM parameter tuning (Miranda et al., 2014; Mantovani et al., 2015), among others (Smith-Miles, 2008; Giraud-Carrier et al., 2004).

Before using MTL, a meta-dataset must be constructed. Typically, each meta-example is associated with a dataset, from which a set of characteristics is extracted. These characteristics are named meta-features and can be either descriptors extracted from dataset (Brazdil et al., 2009), landmarks representing the performance of simple algorithms applied to the dataset (Pfahringer et al., 2000), internal features of models induced by a ML technique for a dataset (Brazdil et al., 2009), or measures of the underlying complexity of the dataset (Ho & Basu, 2002). Each meta-example is labeled with the accuracy value obtained when a set of ML algorithms is applied to the dataset. These knowledge extracted from the data are recorded for a large number of datasets, in order to avoid bias.

This process results in a meta-dataset, where each meta-example represents one of the

datasets. The predictive feature values of a meta-example are the meta-feature values extracted from the dataset associated with the meta-example. Suppose that n classifiers are investigated in the MTL process. The target feature value of the meta-example can be: the algorithm that presented the best performance for the dataset (the meta-dataset will be a conventional n classes multiclass classification task); the performance of each algorithm when applied to the dataset (the meta-dataset will contain n regression tasks); or the ranking position regarding the predictive performance for all the n investigated algorithms (the meta-dataset will be a ranking classification task).

The next step is the induction of a meta-model from the meta-dataset. The meta-model can be induced by ML techniques and can be used in a recommendation system to select the most suitable algorithm(s) for a new dataset. It is important to notice that a theoretical support and a preprocessing step are needed in most of the cases, to provide a refinement of the recommendation framework (Smith-Miles, 2008). The theoretical perspective provides a validation of the meta-models by an expert. This information can be used to generate insights into algorithm behavior or even about preprocessing steps that can be used to refine the entire process (Rossi et al., 2014).

This chapter investigates the use of meta-models to recommend NF techniques, among those described in Chapter 3, for the identification of noisy examples. Therefore, the meta-dataset contains datasets with different noisy levels as meta-examples. These corrupted datasets are produced by the controlled injection of different noise levels in benchmark ML datasets. The recorded performance of the NF technique is used to label the meta-examples. To characterize the datasets, we employed a set of standard measures from the MTL literature and also measures able to describe the complexity of a classification problem (Soares et al., 2001; Castiello et al., 2005; Ho & Basu, 2002; Orriols-Puig et al., 2010).

This study will investigate two alternatives to recommend NF techniques: one based on the prediction of the performance of the crisp NF techniques; the other one is based on the recommendation of the best soft NF technique for a specific problem, among the individual NF or ensemble of NF. We believe that a good predictive performance in the estimation of the crisp filter performance will lead to better label noise identification in new datasets. And the recommendation of one of two best soft NF techniques could decrease the computational cost of filtering once, for a particular dataset, since an individual technique can have the same predictive performance as an ensemble, as shown in Chapter 3. In this case the individual technique should be preferred.

Finally, some of the techniques are further validated using a real dataset from the ecological niche modeling domain with support of a domain expert, who evaluated the quality of the noise predictions. This study allows evaluating the effectiveness of the recommendation system and of the quality of the noise predictions obtained.

The contributions from this chapter can be summarized as:

- Proposal of a new MTL approach based on the induction of meta-regressors able to predict the expected performance of crisp NF techniques in the identification of noisy data.
- Proposal of a new MTL approach based on the induction of meta-classifiers able to predict the best soft NF technique for a new dataset.
- Show the relevance of MTL as a decision support tool for the recommendation of a suitable NF technique for a new classification dataset.
- Validation of proposed approach on a real dataset with support of a domain expert.

In the next sections, we present the background information necessary to describe the proposed approach: Section 4.1 explains the framework used to model the recommendation systems, including the meta-features, the algorithms and the recommendation evaluation process. Section 4.2 describes the experiments carried out to validate each MTL proposal, while Sections 4.3 and 4.4 report and analyze the experimental results obtained. Section 4.5 describes a case study using an ecological dataset, whose experimental results are evaluated with support from a domain expert. Finally, Section 4.6 summarizes the main conclusions from this study.

4.1 Modelling the Algorithm Selection Problem

The algorithm selection problem was initially addressed by Rice (1976). In this study, an abstract model was proposed to systematize the algorithm selection problem. The main goal of this model is to predict the best algorithm when more than one algorithm is available. There are four components in this model: the problem instances (P) which are the datasets in MTL, the instance features (F), which are the meta-features, the algorithms (A), which are the ML algorithms used in the base-level experiments, and the evaluation measures (Y), which maps each algorithm to a set of performance measure values. For a problem instance p and the meta-features f , the model finds the algorithm α whose recommendation $S(f(p))$ maximizes the performance mapping $y(\alpha(p))$.

Smith-Miles (2008) improved this abstract model proposing generalizations related with automatic algorithm selection and algorithm design. In her proposed model, some components are added: MTL algorithms (S); generation of empirical rules or algorithm rankings; examination of the empirical results; theoretical support; and loops for refining the algorithms. Figure 4.1 illustrates this model.

Is important to observe that A is not necessarily a ML algorithm. This algorithm selection diagram can be used to support tasks like optimization and preprocessing. For preprocessing, as noise detection, the diagram can be adapted by replacing the A component by the NF techniques, the Y component by some evaluation measure for NF and by

Figure 4.1: Smith-Miles (2008) algorithm selection diagram.
(Adapted from Smith-Miles (2008))

adding specific meta-features for noise pattern identification in F . The recommendation system can be adapted to predict the NF performance or even the best NF technique. Next, each component in the adapted model will be detailed.

4.1.1 Instance Features

The meta-features (F) are designed to extract general properties of datasets. Called characterization measures, they are able to provide evidence about the future performance of the investigated techniques (Soares et al., 2001; Reif, 2012). These measures must be able to predict, with a low computational cost, the performance of a group of algorithms. According to Giraud-Carrier et al. (2009), the main standard measures used in MTL can be divided into three groups:

- **Simple, statistical and information-theoretic features.** These are the most simple measures for extracting general properties of the datasets. They can be further divided into simple features, based on statistics and information theoretic (Michie et al., 1994; Brazdil et al., 2009). Examples of simple features are the number of examples, the number of features and the number of classes in a dataset. Measures based on statistics describe data distribution indicators, like average, standard deviation, correlation and kurtosis. The information theoretic measures include entropy and mutual information.

- **Model-based features.** These measures describe characteristics of the investigated models (Peng et al., 2002; Bensusan et al., 2000). These meta-features can include, for example, the description of the DT induced for a dataset (Giraud-Carrier et al., 2009), like its number of leaf nodes and the maximum depth of the tree.
- **Landmarking.** Landmarking are simple and fast algorithms, from which performance characteristics can be extracted (Pfahring et al., 2000). These meta-features include accuracy, precision and recall obtained by the algorithms.

Once this study is concerned with noise detection, is important to use measures capable to describe the occurrence of noise in a dataset. Previous studies showed the effectiveness of the complexity measures described in Chapter 2 to characterize noisy datasets (Sáez et al., 2013; Garcia et al., 2015). In Sáez et al. (2013), complexity measures were used to measure the efficacy of using a NF technique for increasing the predictive performance of the k -NN classifier. The proposed methodology was able to predict whether the use of a filter should be statistically beneficial for some specific scenarios. In Garcia et al. (2015), the investigation of the effect of distinct levels of label noise in the values of the same complexity measures was extended to include multiclass classification tasks. The benefits of this extension were experimentally investigated. The experimental results showed the effectiveness of these measures to characterize noisy multiclass datasets.

Table 4.1 summarizes the characterization measures used to describe the noisy datasets: standard and complexity measures.

4.1.2 Problem Instances

The problem instances (p) are the datasets that will be used to generate the meta-dataset through the instance features extraction $f(p)$. As in any learning task, the ideal situation would be to use a large number of datasets, in order to induce a reliable meta-model. To reduce the presence of bias, datasets from several data repositories, like UCI (Lichman, 2013), Keel (Alcalá-Fdez et al., 2011) and standard repositories hosting services such as mldata.org¹ (Braun et al., 2014) and OpenML² (Vanschoren et al., 2013), can be used.

Other strategies to increase the number of datasets is the use of artificial data or changing the distribution of the classes to increase the number of examples in the meta-dataset (Hilario & Kalousis, 2000; Vanschoren & Blockeel, 2006). There are also more complex strategies, like the use of active learning for instance selection and the use of datasetoids, which is a data manipulation method to obtain new datasets from existing ones (Prudêncio & Ludermir, 2007; Prudêncio et al., 2011). In this work noisy versions of

¹<http://dataverse.org/>

²<http://www.openml.org/>

Table 4.1: Summary of the characterization measures.

Class	Type	Acronym	Description
Standard Measures	Simple features	Cls	Number of classes
		Atr	Number of features
		Num	Number of numeric features
		Nom	Number of nominal features
		Spl	Number of examples
		Dim	Spl/Atr
		NumRate	Num/Atr
		NomRate	Nom/Atr
		Sym (Min, Max, Mean, Sd, Sum)	Distribution of categories in the features
		Cl (Min, Max, Mean, Sd)	Classes distribution
	Statistical features	Sks	Skewness
		SksP	Skewness for normalized dataset
		Kts	Kurtosis
		KtsP	Kurtosis for normalized dataset
		AbsC	Correlation between features
		CanC	Canonical correlations between matrices
	Information-theoretic features	Fnd	Fraction of canonical correlations
		ClEnt	Entropy
		NCIEnt	Entropy for normalized dataset
		AtrEnt	Mean of feature entropy
		NAttrEnt	Mean of feature entropy for normalized dataset
		JEnt	Joint Entropy
		MutInf	Mutual Information
		EAttr	$ClEnt/MutInf$
	Model-based features (Tree)	NoiSig	$(AtrEnt - MutInf)/MutInf$
		Node	Number of nodes
		Leave	Number of leaves
		NodeAtr	Number of nodes per features
		NodeIns	Number of nodes per instances
		LeafCor	$Leave/Spl$
		L (Min, Max, Mean, Sd)	Distribution of levels of depth
		B (Min, Max, Mean, Sd)	Distribution of levels of branch
		Atr (Min, Max, Mean, Sd)	Distribution of features used
	Landmarking	Nb	Naive Bayes accuracy
		St (Min, Max, Mean, Sd)	Distribution of Decision Stumps
		StMinGain	Minimum Gain ratio of Decision Stumps
		StRand	Random Gain ratio of Decision Stumps
		NN	1-Nearest Neighbor
Complexity Measures	Overlap of feature values	F1	Maximum Fisher's discriminant ratio
		F1v	Directional-vector maximum Fisher's discriminant ratio
		F2	Overlap of the per-class bounding boxes
		F3	Maximum feature efficiency
		F4	Collective feature efficiency
	Classes separability	L1	Minimized sum of the error distance of a linear classifier
		L2	Training error of a linear classifier
		N1	Fraction of points on the class boundary
		N2	Ratio of average intra/inter class nearest neighbor distance
		N3	Leave-one-out error rate of the 1-nearest neighbor classifier
	Geometry, topology and density	L3	Nonlinearity of a linear classifier
		N4	Nonlinearity of the 1-nearest neighbor classifier
		T1	Fraction of maximum covering spheres

all datasets were produced by the random injection of label noise at different rates. The meta-dataset is created by extracting one meta-example from each real dataset described in Section 2.3.1. These meta-examples were generated by using the median of the values of the meta-features in order to avoid outliers.

4.1.3 Algorithms

The algorithms (α) represent a set of the algorithms that will be the candidates used in the algorithm selection process. Ideally, these algorithms must be sufficiently different from each other and represent all regions in the algorithm space. Brazdil et al. (2009) proposed four conditions that, when satisfied, increase the chances of build a bias-free

meta-dataset: the use of algorithms with different bias; at least one algorithm must have better performance than a reference, baseline, algorithm; the algorithm needs to be better than the others for at least a subset of datasets; and each algorithm needs to be better than each one of the others for at least one dataset.

The algorithms used in this study will be the NF techniques described in Chapter 3. A recommendation system based on MTL capable to suggest a specific NF technique or even the expected performance of a specific NF for a new dataset could not only improve the noise detection performance in the preprocessing step, but also provide information about particular areas of competence of the NFs.

4.1.4 Evaluation measures

The models induced by the algorithms can be evaluated by different measures (y). Most of the studies in the MTL use the accuracy measure for classification tasks, but other indices, like F_β , AUC and kappa, can also be used. For regression problems, the employment of Mean Squared Error (MSE) is usual. Other areas, like clustering and optimization have their own measures. In this study, the performance of the NF techniques will be evaluated with the measures described in Section 3.3. For NF techniques based on crisp decision, the measures precision, recall and F_β are good candidates to be used. For soft NF techniques, measures like $p@n$ and NR-AUC can be used.

4.1.5 Learning using the meta-dataset

After the extraction of the characterization measures from the datasets $f(p)$ and the evaluation of the algorithms $y(\alpha(p))$ for these datasets, the next step is labeling each meta-example in the meta-dataset. Brazdil et al. (2009) summarizes the four main properties frequently used to obtain labels: the algorithm that presented the best performance on the dataset; a ranking of the algorithms according to their performance on the dataset, where the algorithm in the top is the one that presented the best performance; the performance of each algorithm on the dataset; and the model description.

The first option is used when the information needed is only the best algorithm to be used. When it is important to recommend a group of algorithms, following a recommendation order, the ranking prediction is more suitable (Brazdil et al., 2003). For the cases where the best predictive performance is required, the use of regressors can provide an estimate of the performance of each algorithm (Bensusan & Kalousis, 2001). In some specific cases, only a description of the learning model is desired. This is the case of the model description approach. The recommendation system produced by using MTL can also predict the best values for the hyper-parameters of a specific algorithm (Pfahring et al., 2000; Kalousis, 2002). In this work we are interested to predict the performance of the noise filters and recommend the best filter for specific problems.

4.2 Evaluating MTL for NF prediction

This section presents the experiments carried out to evaluate the MTL approaches, when they are used to predict the expected performance of crisp NF techniques and to predict the best soft NF technique, among the techniques described in Chapter 3. As previously mentioned in this chapter, the meta-dataset contains noisy datasets as examples. This meta-dataset is employed in the induction of the meta-models for NF recommendation. In particular, these experiments aim to:

1. Evaluate the meta-models induced to estimate the predictive performance of crisp NF techniques and the best soft NF technique in label noise identification. For the crisp NF, meta-regressors are induced and the performance is measured by filter. For the soft NF, meta-classifiers are induced and the performance is measured in the overall recommendation of the best filter.
2. Validate the recommendation system on a real dataset with the support of a domain expert. A case study using a real dataset from the ecological niche modeling domain is presented. In it, a NF technique recommended by the second MTL induced model is evaluated. Herewith, it is possible to evaluate the quality of the noise predictions obtained and the relevance of MTL as a decision support tool for the recommendation of a suitable NF technique for a new classification dataset.

The first MTL approach investigated in this Thesis predicts the performance of crisp NF techniques. For such six filters analyzed in previous chapter are used: HARF, SEF, DEF, AENN, GNN and PruneSF. These NF techniques were selected because they are well known, have different biases and have presented good performance in recent studies (Frenay & Verleysen, 2014). The performance of the NF techniques was evaluated using the F_1 measure.

The second MTL approach recommends the best soft NF technique. Using the results from the previous chapter, and assuming the fact that most of the real datasets has low levels of noise, the soft filters chosen to label the meta-dataset were HARF and E11, an ensemble of NF techniques. The use of a recommendation system to choose, for a new dataset, between the best ensemble and the best individual filter, could not only improve the noise detection predictive performance for the cases where the individual filter already has a good performance, but also decrease the overall computational cost of the filtering step.

For the sake of generality, the meta-dataset is built using the noisy datasets described in Section 2.3.1. Each meta-example is described by a set of meta-features from the MTL literature and also complexity-based measures, as discussed in Section 4.1.1. These meta-features are described in Table 4.1. The parameters employed for the filters investigated in this work are the same as those described in Section 3.4.2.

Next section will detail the experimental protocol previously outlined.

4.2.1 Datasets

In the base-level, noisy versions of the datasets from Table 2.2 are created using the systematic model of noise imputation described in Section 2.3.1. For each dataset, random noise was added at rates of 5%, 10%, 20% and 40%. This data corruption was controlled so as to allow the identification of the noisy examples. Moreover, since the selection of the examples to be corrupted was random, 10 different noisy versions of the datasets were generated, for each noise level considered.

For the first approach, each meta-example is represented by the meta-features and labeled with the F_1 obtained by the six crisp NF techniques. To avoid outliers, each meta-example is represented by the median of the values of the meta-features. Thus, a meta-dataset was created with 90 meta-examples, 70 meta-features (combination of the characterization measures with the complexity measures) and the performance of the six crisp NF techniques.

In the second approach, the meta-examples were also generated using the median of the values of the meta-features to avoid outliers and labeled according to the recommended use of ensembles or not. If the ensemble E11 shows a better performance than HARF for a given dataset, the corresponding meta-example is labeled accordingly. If there are ties, the HARF technique is preferred, since it has a lower computational cost. This results in a meta-dataset with 90 meta-examples, 70 meta-features. The percentage of examples in the majority class, the class E11, is of 54.44%.

4.2.2 Methodology

In the first approach, the meta-dataset was fed into regression algorithms. Each algorithm induces a meta-regressor model for a particular filter, using the meta-dataset as input. When a new dataset is presented to the recommendation system, all meta-regressors are applied to the meta-feature values of the dataset to predict the expected performance of each filter for this dataset. The output values obtained for the different NF techniques will be used to recommend the most promising filter for this new dataset. The NF techniques with the highest predicted performance will be recommended.

The regressors were generated using the leave-one-out methodology. The average leave-one-out MSE performance of the meta-regressors was computed. The MSE values for the six NF techniques were compared with the MSE achieved when baseline strategies are employed. Two simple baselines are used: The first baseline, Random Technique (RD), randomly chooses label values from 0 to 1 for each example by sampling with replacement. The second baseline, Default Technique (DF), randomly draws a meta-example and assigns its label to the new example every time a prediction is required.

Three regression algorithms were employed to induce the meta-regressors: k -NN with gaussian kernel (Distance-weighted k -NN (DWNN)) (Mitchell, 1997), RF with 500 DTs (Breiman, 2001) and SVM with radial kernel function (Vapnik, 1995). These regression algorithms are representatives of different learning paradigms and are known for their good predictive performance in regression tasks. A Friedman statistical test (Demšar, 2006) with 95% of confidence value was applied to compare the predictive performance of the meta-regressors in each case.

In the second approach, meta-classifier models are also induced using the leave-one-out methodology. Five meta-classifier models were used: C4.5, 3-NN with minkowski distance, RF with 500 DTs and SVM with a radial kernel function. A baseline that always predicts the majority class of the meta-examples was also used. These models were evaluated using the accuracy measure obtained for the test data. A Wilcoxon signed-rank statistical test (Demšar, 2006) with 95% of confidence value was also applied to compare the predictive performance of the meta-classifiers against the baseline.

To investigate the importance of each meta-feature in the prediction of the performance of the filters, feature selection techniques were applied to the meta-dataset. The best subset of meta-features was selected using the Correlation-based Feature Selection (CFS) technique (Hall, 1999) with regression values discretized. This technique finds the feature subset using correlation measures and a best first search algorithm to the training data.

4.3 Experimental Evaluation to Predict the Filter Performance

This section presents the experimental results obtained in the MTL approach to predict the expected performance of the crisp NF techniques. Section 4.3.1 reports a meta-dataset analysis, while Section 4.3.2 presents the results obtained in the evaluation of the meta-regressors.

4.3.1 Experimental Analysis of the Meta-dataset

Figure 4.2 summarizes the distribution of the F_1 performance for each crisp NF technique in the meta-dataset. Figure 4.2(a) shows the number of times each filter presented the best F_1 performance in noise identification in the meta-dataset. In this figure, each column represents one NF and the y -axis shows to the number of wins for each NF. Another analysis performed was the number of times each filter presented the highest F_1 performance, compared to each one of the others filters, in noise identification. Figure 4.2(b) shows this result. The x -axis represents the NF techniques and the y -axis shows the number of wins for a specific NF. The HARF technique is shown by black dots, SEF by red triangles, DEF by blue squares, AENN by green crosses, GNN by purple hollow

squares with crosses inside and PruneSF by orange asterisks. The NF techniques with better performance will have a high number of wins. If there are ties, the number of wins increase for all the best NF techniques.

According to Figure 4.2(a), the performance of the NF techniques was imbalanced, but each technique presented the best performance for at least one dataset. The highest performance was obtained by DEF, followed by HARF and PruneSF. The filters AENN, SEF and GNN were considered the best filter only one time. The AENN was the best for the *monks2* dataset with $F_1 = 0.4879$, SEF for *planning-relax* dataset with $F_1 = 0.4851$ and GNN for *movement-libras* with $F_1 = 0.7359$ of F_1 performance. Thus, even unbalanced, the meta-dataset has all filters represented.

In Figure 4.2(b) the NF technique with best performance is DEF. It has a higher number of wins compared to all the other NF techniques. The filter HARF and SEF also had a high number of wins. The worst filters are GNN and PruneSF. GNN was better than AENN. PruneSF was better than AENN and GNN.

Overall, the results show that the filters DEF, HARF and PruneSF presented the best performance in noise filtering for the base datasets. The SEF filter showed intermediate performance when compared to the other NF techniques. The filters GNN and AENN were the worst and did not show a good performance in noise identification compared to the other techniques. Despite the low performance of the last two filters, the built meta-dataset respect the conditions proposed by Brazdil et al. (2003) and includes them to increase the chances of building a bias-free meta-dataset.

4.3.2 Performance of the Meta-regressors

The experiments presented in this section measure the predictive performance obtained by the meta-regressors in the prediction of the F_1 value of each crisp NF technique. Figure 4.3 shows boxplots of the MSE performance values obtained by the meta-regressors induced for each NF. In this figure, the meta-regressors DWNN, RF and SVM are represented using the gray color and the baselines RD and DF are represented in black. The y -axis shows the MSE in a logarithm scale, in order to emphasize the lowest values.

According to these results, the DWNN, RF and SVM meta-regressors presented lower MSE than the baselines and are, therefore, more accurate in most of the cases. DF is a more strict baseline, since, different from the other baseline, RD, DF uses training data information to obtain its predictions. In general, the meta-regressors also showed a more stable behavior when compared to the baselines, whose performance varied more. Among the meta-regressors, for almost all cases, SVM results presented the lowest MSE, but usually with the largest variation. The DWNN regressor presented the worst predictive performance, with higher MSE values.

Using the Friedman statistical test with the Nemenyi post-test at 95% of confidence

(a) Distribution of the number of times each NF presented the highest F_1 .

(b) Distribution of the number of times each NF presented highest F_1 when compared with each NF technique.

Figure 4.2: Performance of the six crisp NF techniques.

Figure 4.3: MSE of each meta-regressor for each NF technique in the meta-dataset.

level (Demšar, 2006), the following results can be reported for each NF technique and for each regression technique:

- For the NF techniques AENN, GNN and PruneSF: the DNN, RF and SVM meta-regressors presented better predictive performance than DF and RD. The DF meta-regressor obtained better predictive performance than RD. The best meta-regressor was better than the best baseline.
- For the NF techniques HARF, DEF and SEF: the DNN, RF and SVM meta-regressors predictive performances were better than those of DF and RD. The best meta-regressor was better than the best baseline.

According to the experimental results of the meta-regressors illustrated in Figure 4.3

and the statistical tests performed, all meta-regressors were able to predict the F_1 performance with higher accuracy than the baselines. The SVM regressor usually presented the lowest MSE values, except for the AENN filter, where RF presented the best predictive performance. The best baseline was DF, in some cases with statistical difference.

Figure 4.4 presents the increase of F_1 predictive performance obtained by the NF techniques when the NF predicted as best by the meta-regressors induced are used in noise detection (base-level) instead of the NF predicted by the baselines DF (Figure 4.4(a)) and RD (Figure 4.4(b)). The x -axis shows the meta-regressors and the y -axis represents the increase of F_1 predictive performance when compared to the corresponding baseline. Positive values indicate an increase of the F_1 predictive performance and negative values, a decrease of the predictive performance.

(a) Difference of performance in the base-level when using DF as baseline. (b) Difference of performance in the base-level when using RD as baseline.

Figure 4.4: Performance of the six crisp NF techniques.

In Figure 4.4(a), the increase in the base-level predictive performance obtained by using the meta-regressors DWNN, RF and SVM were higher than using the DF baseline. RD had a high decrease of performance. In Figure 4.4(b), all the meta-regressors increased the performance, including the DF baseline. The RF meta-regressor presented the best results in both cases. Therefore, although SVM meta-regressor presented the lowest MSE, the RF meta-regressor was more accurate to predict the performance of each NF technique.

Figure 4.5 shows the 10 top-ranked meta-features selected by CFS as the most important to predict the NF performance, independent of the meta-regressor. The x -axis represents the measures and the y -axis shows how frequently they were selected. The standard meta-features for the characterization of datasets are represented in black, while complexity-based measures are colored in gray.

Figure 4.5: Frequency with which each meta-feature was selected by CFS technique.

The meta-features selected as the most important are based on standard measures. Only one complexity measure is top-ranked, the N4 measure. The top-ranked meta-features include all landmarking measures, one information-theoretic measure related with mutual information, two statistical measures related with correlation and two simple features, which were number of examples and classes distribution. It is important to note that the measure N4 and NN have similar concepts, which can indicate redundant information. If we remove the redundancy, it is expected that the N4 measure would be ranked before.

4.4 Experimental Evaluation of the Filter Recommendation

This section evaluates the MTL approach to recommend the best soft NF technique for a new dataset. The goal is to decrease the computational cost of the preprocessing step by recommending HARF when it has a predictive performance similar to the best ensemble E11. Section 4.4.1 reports a meta-dataset analysis, while Section 4.4.2 presents the results obtained in the evaluation of the meta-classifiers.

4.4.1 Experimental analysis of the meta-dataset

Figure 4.6 shows the number of times each NF technique presented the best $p@n$ performance in noise identification. The x -axis represents the filters selected to label the meta-dataset, while the y -axis corresponds to the number of wins for each filter. If there are ties, the number of wins increase for all the column involved in the tie.

Figure 4.6: Distribution of highest $p@n$.

The highest performance was obtained by E11, which labels 54.44% of the meta-examples. In 10 datasets both HARF and E11 presented the same performance. In this case, the HARF filter was preferred to label the meta-examples, since it has a lower computational cost. Thus, the meta-dataset has 54.44% of E11 and 45.56% of HARF examples.

Overall, the results show that the meta-dataset is highly balanced and respect most of the conditions proposed by Brazdil et al. (2003). With respect to the condition about algorithms with different biases, even though E11 is composed by HARF, the similarity (in Figure 3.9) between these filters are low, which increase the chances of building a bias-free meta-dataset.

4.4.2 Performance of the Meta-classifiers

Figure 4.7 shows the accuracy of the meta-classifiers in the meta-level. The x -axis represents the classifiers used and the y -axis the predictive performance using leave-one-out. The horizontal line represents the performance of the baseline. The baseline is the classification in the majority class, which corresponds to the ensemble.

These results show that MTL can provide a good recommendation for the soft NF techniques for new datasets. According to Figure 4.7, the predictive performance of all meta-classifiers was better than the baseline. Among the classifiers, the C4.5 algorithm presented the best predictive performance, with almost 0.75 accuracy. SVM, RF and 3-NN presented a similar performance. The p -values of the Wilcoxon's test shown statistical difference for C4.5 at 95% of confidence level.

Figure 4.8 shows the percentual of increase in the predictive performance obtained by the NF technique when they are recommended by the meta-classifiers, instead of using one baseline. The x -axis represents the classifiers used and the y -axis the increase in the

Figure 4.7: Accuracy of each meta-classifier in the meta-dataset.

predictive performance. The horizontal line represents the performance of the baseline. In Figure 4.8(a) the baseline is the E11 filter and in Figure 4.8(b) the baseline is the HARF filter. In both cases we also added the performance of the filters without the use of MTL.

(a) Difference of performance in the base-level when using E11 as baseline.

(b) Difference of performance in the base-level when using HARF as baseline.

Figure 4.8: Performance of meta-models in the base-level.

These results show that the increase of predictive performance in the base-level for the classifiers C4.5 and RF was higher than using the baseline prediction. The predictive performance of 3-NN and SVM was lower than the baseline prediction. Thus, although they presented a good predictive accuracy in the meta-level, the same is not true for their recommended soft NF techniques.

If, on the other hand, HARF is used as baseline, given its lower computational cost, the predictive performance of the meta-classifiers in the base-level are also superior. Thus, the predictive performance of the NFs recommended by the meta-model was better than that of the NFs recommended when either E11 or HARF was used as baseline.

Figure 4.9 shows the pruned DT meta-model. The root and internal nodes are associated with the meta-features selected as the most important by the C4.5 algorithm and the leaf nodes are assigned to one of the two meta-classes (HARF or E11). The pruned DT also shows the number of training examples and the purity degree for each leaf. In each leaf, a rectangle shows the distribution of the examples from the two meta-classes in the leaf. The black region is associated with the E11 meta-class, and the white to the HARF meta-class. The larger the region, the larger the number of examples from the related class.

The meta-features regarded as the most important by the pruned DT are *Num*, *Dim*, *NodeIns* and N_4 . While *Num* and *Dim* are simple measures, *NodeIns* is a DT-based measure and N_4 is a complexity measure. The *Num* and *Dim* meta-features are related with the number of numeric attributes and the proportion of examples per attribute. *NodeIns* is based on the number of nodes per instance in a DT. N_4 is the nonlinearity of the 1-NN classifier. The value of these meta-features can define the best option, between an individual NF and an ensemble of NFs, for a new dataset. Among them, as N_4 appears in the root node, it can be considered the most informative meta-feature.

Another important information in the DT meta-model is the leaf purity degree for the training instances. The model has eight leaves, and six of them are almost 100% pure. Among the leaves with a high purity degree, two have more than 10 meta-examples. Among the leaves with a low purity degree, one has more than 10 meta-examples. Therefore, the meta-model has a high confidence level.

Besides the predictive performance analysis, this study also evaluated the additional computational cost due to the use of the recommendation system, when compared with the use of E11, which is the NF associated with the majority class. The additional computational cost includes the extraction of meta-feature values from a new dataset and the running time required by the recommendation system to recommend one of the two NFs, E11 or HARF. This evaluation used leave-one-out and averaged 10 executions for each dataset. The average and standard deviation of the running times in seconds were: 37.84 ± 0.17 , when using E11, and 14.73 ± 0.04 , when using the recommendation system. As can be seen, the recommendation system was able to decrease the running time when compared with E11. Therefore, also regarding the processing time, it is more suitable to first use the recommendation system than directly applying E11.

Figure 4.9: Meta DT model for NF recommendation.

4.5 Case Study: Ecology Data

This section validates the filtering importance using a real dataset from the Ecological niche modeling domain. This dataset was provided and analyzed by Dr. Augusto Hashimoto de Mendonça, who works at the Center for Water Resources & Applied Ecology from Environmental Engineering Sciences of the School of Engineering of São Carlos at University of São Paulo and Professor Dr. Giselda Durigan from the Forestry Institute of the State of São Paulo. Section 4.5.1 describes this real dataset along with its main features. Section 4.5.2 reports the use of the recommendation system to suggest the best filter and Section 4.5.3 presents the experimental results obtained.

4.5.1 Ecological Dataset

Ecological niche datasets show the presence or absence of species in georeferenced points. These datasets are usually imbalanced, since examples from the specie absence class are very often difficult to be sampled. The dataset employed here, named *species*, contains two classes, which represent the presence and absence of a non native specie *Hedy-chium coronarium* in a set of georeferenced points from protected areas of the Brazilian state of São Paulo. *H. coronarium* is originally from the Himalayas region of Nepal, India and China. It is characterized as a perennial flowering plant whose height varies from one to three meters, which propagates vegetatively and forms dense populations. This specie is expected to be found in humid habitats, with partial sun exposure in natural or disturbed areas, usually in the border of lowland areas, rivers and forest fragments. It grows in fertile soil that is preferably in shaded or semi-shaded areas, in wetland habitats and in environments with high temperatures during the whole year.

Redundant features and missing values were previously removed from the dataset *species*, resulting in a binary classification dataset with five predictive features, 1365 examples and an unbalance rate of 80%. The predictive features are: type of vegetation, degree of conservation of vegetation, place where the point was sampled, degree of green and aridity of the ground. Table 4.2 summarizes the predictive features.

Table 4.2: Summary the predictive features of the *species* dataset.

Features	Type	Values
Type of vegetation	Nominal	rain forest
		mixed rain forest
		semi-deciduous forest
		deciduous forest
		ecotone
		wetland
		grasslands
		cerrado stricto sensu
		high dense cerrado
		cerrado forest
		gallery forest
		open restinga
		restinga forest
Degree of conservation	Nominal	anthropic area
		degraded native vegetation
		native vegetation in regeneration
		native vegetation
Place sampled	Nominal	highway margins
		lowland
		riparian zone
		fragment edge
		inner fragment
Degree of green	Numeric	[981 : 5520]
Degree of aridity	Numeric	[6600 : 26968]

In some cases, the absence of the specie is a misclassification. Although in a georeferenced point the species is not present, it can be regarded as seen depending of the size

of the protected area analyzed, since it is situated next to another region that was not visited by the data collector. This is a classical example of label noise. In other cases, the presence or absence of the specie is temporal. At a given moment, a given individual could be present in a habitat incompatible with its niche characteristics or could be absent in a habitat compatible with its niche characteristics. By being present in a place incompatible with their needs, the probability of the species to remain and reproduce in this place is very limited. This case is a false presence in terms of environmental compatibility. The absence in a place compatible with their needs indicates that no dispersal event happened in that area before. This case represents a false absence.

Therefore, the NF techniques can assist in the identification of these two events: (i) noise in the absence class and (ii) examples that were classified as present or absent but in fact correspond only to a momentary state that might change in the future.

4.5.2 Filtering Recommendation

Initially, meta-features were extracted from the *species* dataset. The recommendation system created in the previous section was applied to this dataset. The C4.5 meta-model recommended the use of the E11 filter with 96% of confidence. To evaluate the prediction of the meta-classifier, HARF was also applied to the dataset and presented a lower predictive performance. The filter returned a higher number of safe examples in the subset of potentially noisy examples.

When E11 was applied to the *species* dataset, it returned the NDP values associated with all examples. The examples with NDP values higher than 0.75 were selected to be further analyzed by a domain expert. While the examples evaluated by the expert as noisy were regarded as true positives, those examples evaluated as safe examples were regarded as false positives. A low number of false positives corresponds to a good performance in NF detection.

4.5.3 Experimental Results

Using the previous filter and threshold, 59 examples were detected as noisy, 12 in the absence class and 47 in the presence class.

Regarding the noisy examples in the presence class, 40 of them presents conservation with primary vegetation status, no signs of disturbance and minimal human intervention. The conservation status largely reduces the chances of invasion of *H. coronarium*. Even if the type of vegetation presents good characteristics for the development of the species, the invasion does not occur either for lack of propagules or lack of opportunities (window invasion) generated by stochastic events or disorders that enable its establishment. Among the seven remaining noisy examples, five are in areas of vegetation that are in regeneration or degeneration, but located in inner fragments. These are also conditions that minimize

the chances of invasion. Only two examples were misclassified by the ensemble filter. This example has native vegetation in riparian zone, which is favorable to the invasion.

Regarding the noisy examples from the absence class, five are examples where the location and the conservation status do not favor the appearance of *H. coronarium*. These cases are in primary vegetation regions and are safe examples. The other seven examples are with conservation status of anthropic areas or state with vegetation in regeneration, located in fragment edges or highway margins. These examples are noisy and must be removed.

Even with a lower degree of importance, the type of native vegetation also influences the appearance of *H. coronarium*. Gallery forests are vegetations that grow in water bodies and create ideal conditions for the establishment of the species. In this case, the species is not only present in places where the example is located inside the primary vegetation fragment. The same happens for rain forests. This vegetation would be an ideal environment for the development of intrusive species. The rain forest is the Brazilian environment that most resembles the natural habitat of *H. coronarium*, usually where there is high incidence of solar radiation and rainfall.

The index of aridity and degree of green features are also indirectly related to the type of vegetation, since the availability of water and sunlight are the most important factors responsible for structuring and the composition of natural ecosystems. The NF identified absence examples with high dryness index values, which represent vegetation types with higher water availability. No pattern was identified in the degree of green.

Overall, the filtering step efficiently identified potentially noisy examples. For data modeling, these examples should be removed to avoid their negative interference in the induced model. From the expert point of view, these examples should be monitored, since they represent areas in process of degeneration.

4.6 Chapter Remarks

This chapter proposed and investigated the use of MTL for the recommendation of NF techniques. Two new approaches were proposed, one for NF performance prediction and another for NF technique selection. The two proposed approaches were experimentally evaluated using a large set of public datasets with different levels of artificially imputed noise. Two meta-datasets were created, one for each approach. These datasets had the same meta-features, which were standard and complexity meta-features. The label meta-feature was different for each approach.

The first approach evaluated the use MTL to predict the performance of crisp NF techniques. For such, meta-regressors were induced from the meta-dataset. The label features of the meta-dataset were the F_1 performance obtained by different filters. Six NF techniques with different biases were used. The experimental results obtained in

the recommendation of the performance of crisp NF technique showed a good predictive performance for all meta-regressors. These experimental results support that it is possible to predict the F_1 performance of the NF techniques with a low error rate.

The second approach investigated the use of MTL to recommend the most suitable soft NF technique for the identification of noisy data, taking computational cost into account. Two alternatives could be recommended, the NF technique that presented the best predictive performance in previous experiments, HARF, and an ensemble of soft NF techniques, E11. Therefore, this was a binary classification task and one label feature with two values was used in meta-dataset, one value for each class. The experimental results showed that, for the investigated datasets, the recommender system was able to reduce the cost keeping the predictive performance.

To complement and validate these results, the recommendation system was applied to a real dataset in a label noise prone application. An expert in the dataset domain analyzed the results of the filtering process in this real dataset. The experimental results confirmed the benefits and the good predictive performance of the recommendation system.

This chapter is based on the following papers:

- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2016). “Noise detection in the meta-learning level”. *Neurocomputing*, 176:14 - 25.
- Garcia, L. P. F., Lorena, A. C., Matwin, S., & de Carvalho, A. C. P. L. F. (2016). “Ensembles of label noise filters: a ranking approach”. *Data Mining and Knowledge Discovery*, *accepted*.

Chapter 5

Conclusion

Noise filtering is an important preprocessing step in the DM process, making data more reliable for pattern extraction. Although a large number of NF techniques have been proposed and are able to reduce the presence of noise in datasets, a growing number of studies identify problems related to low quality data (Sluban et al., 2014; Frenay & Verleysen, 2014; Smith et al., 2014; Sáez et al., 2016). This suggests that there is still room for improvements.

This Thesis investigated existing NF techniques and proposed new NF techniques able to increase the data understanding and to improve the noise detection performance. In this direction, the main research issues investigated in this Thesis are: the use of data complexity measures capable to characterize the presence of noise in datasets; the development of new NF techniques; and the recommendation of the most adequate NF techniques for a new dataset using MTL.

The presence of noise in a classification dataset can affect the complexity of the classification task, making the discrimination of objects from different classes more difficult, and requiring more complex decision boundaries for data separation. This Thesis investigated how noise affects the complexity of classification tasks, by monitoring the sensitivity of several measures of data complexity in the presence of different label noise levels. To characterize the complexity of a classification dataset, measures based on geometric, statistical and structural concepts extracted from the dataset were used. The experimental results show that some measures were more sensitive than others to the addition of noise in a dataset. Some of these measures were also used in the development of a new preprocessing technique for noise identification.

The new NF techniques proposed in this work were experimentally validated and, according to the experimental results, they presented a good predictive performance. In particular, our dynamic ensemble was always among the best performing NF techniques. To highlight the most unreliable instances, this Thesis also adapted various NF techniques to provide a degree of confidence regarding their noise prediction and combined multiple soft NF techniques into ensembles to increase the noise detection accuracy. To evaluate

the filters, a new evaluation measure based on AUC was proposed.

The bias of each NF technique influences its predictive performance on a particular dataset. Therefore, there is no single technique that can be considered the best for all domains or data distributions and choosing a particular filter is not straightforward. MTL has been largely used in the last years to support the recommendation of the most suitable ML algorithm(s) for a new dataset. This Thesis proposed two MTL-based recommendation systems: the first to predict the expected performance of crisp NF techniques and the second to recommend the best soft NF technique for a new dataset. The experimental results show that MTL can predict the expected performance of the investigated NF techniques and provide a good recommendation of the most promising NF techniques to be applied to new classification datasets.

A case study using a real dataset from the ecological niche modeling domain was also presented and evaluated, with the results validated by an expert in the dataset application domain. The soft NF technique applied to this dataset was recommended by the second MTL model. This meta-model recommended the use of an NF ensemble with high confidence. According to the experimental results, the recommended technique obtained a good predictive performance in the detection of noisy examples.

The rest of this chapter is structured as follows. Section 5.1 presents the main contributions of this Thesis. Section 5.2 discusses the main limitations of this research, including some related to experiments about Imbalance Ratio (IR) in preprocessed datasets. Section 5.3 presents some possibilities for future work and emphasizes the maximum theoretical performance for the MTL system. Finally, Section 5.4 enumerates the publications originated from this Thesis.

5.1 Main Contributions

The main contributions from this Thesis are:

1. Showing that the presence of label noise at different levels influences the complexity of a classification task. This was performed by monitoring a group of measures able to characterize the complexity of a classification task from different perspectives;
2. Analyzing a new set of meta-features able to characterize the complexity of a classification task by modeling a classification dataset through a graph structure. These measures consider distinct topological properties of the graph built from the underlying classification dataset;
3. Highlighting the measures that are most sensitive to label noise imputation and using some of them to propose a new preprocessing technique able to identify label noise in a dataset;

4. Proposing of a new NF technique based on ensemble of classifiers for noise identification and the adaptation of various NF techniques to provide a soft decision, which is a degree of confidence in noise prediction;
5. Comparing of the performance of individual and ensemble NF techniques using a large number of datasets with distinct noise levels with a new evaluation measure for the soft decision filters;
6. Proposing of a new MTL approach based on the induction of meta-regressors able to predict the expected performance of crisp NF techniques in the identification of noisy data;
7. Proposing of a MTL approach to recommend the best soft NF technique for a new dataset and validation of the proposed approach on a real dataset with an application domain expert;
8. Showing the relevance of MTL as a decision support tool for the recommendation of the most adequate NF technique for a new classification dataset.

5.2 Limitations

The real datasets used in this work already had an intrinsic noise level that was not considered in the analysis, since it is usually not possible to assert that an example really has a noisy label. Thus, for some datasets, the NF accuracy may be overshadow. The artificial datasets have limitations, too. They were selected according to (Smith et al., 2014), which points out overlap between the classes as the main contributor to instance hardness. Other characteristics, like class separability and geometry and topology, were not considered to generate the data (Amancio et al., 2013). Finally, even the analysis of the ecological dataset has limitations. The domain expert responsible for the analysis of the potential noisy examples pointed by the NF did not consider the false negative (FN) prediction, which is the number of noisy examples disregarded by the filter.

In some recent work (Cummins, 2013; Lorena & de Souto, 2015), some limitations of the complexity measures used in this Thesis were also signalized. Cummins (2013) points out that the F2 measure, which calculates the volume of overlapping of the features values, is incorrect when no overlap occurs. Cummins (2013) proposes changes to deal with such cases, by counting the number of examples where there is overlap, which is only suitable for discrete features. This problem can also happen for measures F3 and F4.

The parameters used by the NF techniques were those adopted on the reference papers. This indicates that the evaluation of the NF techniques was restricted and could be improved with parameter tuning. Furthermore, other types of noise could be added to the datasets, other noise levels could be added and different β parameters in F_β -score and

n in $p@n$ could be used in favor of a better analysis. With respect to ML, we could apply the noise detection in a DM process. This should validate the benefit of NF in the model induction for classification problems.

By monitoring the IR (Tanwani & Farooq, 2010) values before and after the usage of the crisp NF techniques, we can identify another gap related to the effect of the noise detection in the minority class. Figure 5.1 shows the 8 datasets with higher IR. The x -axis represents the noise levels while the y -axis shows the IR for preprocessed dataset for each noise level. The IR after applying HARF is shown by black dots, DEF by red triangles and the perfect noise preprocessing technique (Best) by blue squares. This corresponds to the technique that intentionally, is able to correctly identify all noisy cases. The IR results for Best remain the same for different noise rates, since a uniform random noise imputation method was used, which tends to affect all classes uniformly.

Figure 5.1: IR achieved by the best crisp NF techniques in datasets with the higher IR.

Regarding the IR values, most of the NF techniques tend to produce more imbalanced datasets compared to perfect filtering, except in the *abalone*, *car* and *cardiotocography* datasets. Therefore, they seem to have eliminated safe examples from the minority classes. This is a harmful effect that can be due to the intrinsic data noise level, increasing its probability of being labelled as noisy. Nonetheless, the increase of IR seems to be a harmful effect of noise preprocessing, despite the NF technique employed. The use of NF techniques allied to imbalanced data handling techniques can minimize these effects. This could decrease the reduction of the minority class examples by the filters.

The MTL approaches also have some limitations. The main ones are related to the feature and instance selection steps. In Garcia et al. (2016) a wrapper of the meta-

regressors was used to select the most appropriated features for the problem and to fit the model. The instance selection was also made by a stack of 10 different combinations of meta-examples to produce the meta-datasets. These approaches are highly cost but they could be an alternative to increase the performance of the meta-models. Another problem is the number of meta-examples. An increase in the number of meta-examples could leverage the robustness of the MTL system.

5.3 Prospective work

The main limitations previously appointed in noise detection for classification problems can indicate the future work in this area. Some direct future work should be fine tuning the parameters of the NF techniques, to develop NF techniques specific for each dataset, and to study the noisy patterns in the data. The recommendation of NF techniques with the support of an expert could also increase the knowledge and light the preprocessing step with a background, which is very rare in this area.

Direct related to this Thesis, our experimental protocol and graph-based measures can also be used in other types of analysis, such as in verifying the effects of data imbalance, feature selection, feature discretization, among others. It is also possible to use other combinations of measures to devise new preprocessing filters. We also plan to employ feature selection strategies to evidence the best measures able to characterize noisy data. It would be interesting to investigate how the graph-based measures are affected by the choice of the ϵ parameter used to build the graph. We also plan to use some of the highlighted measures to develop new noise-tolerant algorithms to compare GNN with other up-to-date noise filters.

We would also like to observe the influence of the intrinsic noise level of the datasets in the results, which was not considered in the reported experiments, since it is usually not possible to assert that an example really has a noisy label. To overcome this issue, a hard instance analysis can be done before the filtering process. Another possibility is the use of real datasets which can be validated by specific rules.

Related to the recommendation systems, we plan to evaluate other MTL approaches, like ranking the filters or combining them to be used for a particular dataset. Another gap in the MTL proposal is the increase of base-level performance. Figure 5.2 suggests the increase of F_1 performance obtained by the crisp NF techniques when the NF predicted by meta-regressors were used in the base-level. The x -axis shows the meta-regressors and the y -axis represents the increase of F_1 predictive performance when compared to the baseline. Different from the experiments of Section 4.3.2, these results include the perfect meta-regressor (Best).

The results indicate that, the increase in the base-level predictive performance obtained by the meta-regressors DWNN, RF and SVM were higher than using the DF, but lower

Figure 5.2: Increase of performance by the Best meta-regressor in the base-level when using DF as baseline.

than using the Best meta-regressor. Thus, there is a margin of improvement on MTL for noise detection. The use of meta-features with higher levels of information about the noisy patterns presented in the data seems to be the most simple way to increase the performance in the MTL recommendation systems.

We also plan to investigate other strategies able to improve the filters performance in imbalanced data, especially for the minority classes. It is also relevant to develop a method able to automatically set the threshold for the NDP value to define whether an example is noisy. Possible alternatives are to use complexity measures or cumulative sums of probabilities of NDP until an abrupt change in percentages obtained by the NF techniques.

5.4 Publications

I have published conference and journal papers throughout the research carried out during my PhD. Most of them are directly related to this Thesis. I also had some contributions related to the implementation of filters and making them available in an R package. We also preprocessed the UCI repository and made it available in ARFF files in the project UCI++. Next, I present the list of papers, packages and projects.

Journal papers

- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2015). “Effect of

label noise in the complexity of classification problems”. *Neurocomputing*, 160:108 - 119.

- Garcia, L. P. F., Sáez, J. A., Luengo, J., Lorena, A. C., de Carvalho, A. C. P. L. F., & Herrera F. (2015). “Using the One-vs-One decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems”. *Knowledge-Based Systems*, 90:153 - 164.
- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2016). “Noise detection in the meta-learning level”. *Neurocomputing*, 176:14 - 25.
- Garcia, L. P. F., Lorena, A. C., Matwin, S., & de Carvalho, A. C. P. L. F. (2016). “Ensembles of label noise filters: a ranking approach”. *Data Mining and Knowledge Discovery*, *accepted*.

Conference papers

- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2012). “A study on class noise detection and elimination”. *Brazilian Symposium on Neural Networks (SBRN)*, 13 - 18.
- Garcia, L. P. F., de Carvalho, A. C. P. L. F., Lorena, A. C. (2013). “Noisy data set identification”. *Hybrid Artificial Intelligent Systems (HAIS)*, 629 - 638.
- Lorena, A. C., Garcia, L. P. F., & de Carvalho, A. C. P. L. F. (2015). “Adapting Noise Filters for Ranking”. *Brazilian Conference on Intelligent Systems (BRACIS)*, 299 - 304.

Project

- Garcia, L. P. F. (2015). “A huge collection of preprocessed ARFF datasets for supervised classification problems”. *GitHub Software Repository*, <http://dx.doi.org/10.5281/zenodo.13748>.

R-Package

- Morales, P., Luengo, J., Garcia, L. P. F., Lorena, A. C., de Carvalho, A. C. P. L. F., Herrera F. (2016). “NoiseFiltersR: Label Noise Filters for Data Preprocessing in Classification”. R package version 0.1.0. <https://CRAN.R-project.org/package=NoiseFiltersR>.

References

- Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287. (Cited on page 71.)
- Amancio, D. R., Comin, C. H., Casanova, D., Travieso, G., Bruno, O. M., Rodrigues, F. A., & da F. Costa, L. (2013). A systematic comparison of supervised classifiers. *CoRR*, abs/1311.0202:1–23. (Cited on pages 23 and 93.)
- Batista, G. E. A. P. A. & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533. (Cited on page 3.)
- Bensusan, H., Giraud-Carrier, C., & Kennedy, C. (2000). A higher-order approach to meta-learning. Technical report, University of Bristol. (Cited on page 71.)
- Bensusan, H. & Kalousis, A. (2001). Estimating the predictive accuracy of a classifier. In *12th European Conference on Machine Learning (ECML)*, volume 2167, pag. 25–36. (Cited on page 73.)
- Braun, M. L., Ong, C. S., Hoyer, P. O., Henschel, S., & Sonnenburg, S. (2014). mldata.org: machine learning data set repository. <http://mldata.org/>. (Cited on page 71.)
- Brazdil, P., Giraud-Carrier, C. G., Soares, C., & Vilalta, R. (2009). *Metalearning - Applications to Data Mining*. Cognitive Technologies. Springer, 1 edition. (Cited on pages 4, 67, 70, 72 and 73.)
- Brazdil, P., Soares, C., & da Costa, J. P. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277. (Cited on pages 73, 77 and 82.)
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. (Cited on pages 27, 29, 36 and 76.)

- Brodley, C. E. & Friedl, M. A. (1996). Identifying and eliminating mislabeled training instances. In *13th National Conference on Artificial Intelligence (AAAI)*, pag. 799–805. (Cited on pages 9 and 35.)
- Brodley, C. E. & Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167. (Cited on pages 3, 33, 34 and 35.)
- Brown, G. (2010). *Encyclopedia of Machine Learning*. Springer. (Cited on page 44.)
- Castiello, C., Castellano, G., & Fanelli, A. M. (2005). Meta-data: Characterization of input features for meta-learning. In *Modeling Decisions for Artificial Intelligence (MDAI)*, volume 3558, pag. 457–468. (Cited on page 68.)
- Craswell, N. (2009). Precision at n. In *Encyclopedia of Database Systems*, pag. 2127–2128. (Cited on page 45.)
- Csardi, G. & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*:1–9. (Cited on page 24.)
- Cummins, L. (2013). *Combining and choosing case base maintenance algorithms*. PhD thesis, National University of Ireland. (Cited on page 93.)
- de Souza, B. F., de Carvalho, A. C. P. L. F., & Soares, C. (2010). Empirical evaluation of ranking prediction methods for gene expression data classification. In *12th Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, volume 6433, pag. 194–203. (Cited on page 67.)
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30. (Cited on pages 48, 53, 59, 65, 76 and 79.)
- Eskin, E. (2000). Detecting errors within a corpus using anomaly detection. In *1st North American Chapter of the Association for Computational Linguistics Conference (NAACL)*, pag. 148–153. (Cited on page 33.)
- Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *2nd International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pag. 82–88. (Cited on pages 1 and 9.)
- Frenay, B. & Verleysen, M. (2014). Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869. (Cited on pages 9, 33, 35, 47, 74 and 91.)
- Gamberger, D. & Lavrač, N. (1997). Conditions for occam’s razor applicability and noise elimination. In *9th European Conference on Machine Learning (ECML)*, pag. 108–123. (Cited on page 37.)

- Gamberger, D., Lavrač, N., & Dzeroski, S. (2000). Noise detection and elimination in data preprocessing: Experiments in medical domains. *Applied Artificial Intelligence*, 14(2):205–223. (Cited on page 2.)
- Gamberger, D., Lavrač, N., & Groselj, C. (1999). Experiments with noise filtering in a medical domain. In *16th International Conference on Machine Learning (ICML)*, pag. 143–151. (Cited on pages 3, 11, 33, 35 and 38.)
- Ganapathiraju, A. & Picone, J. (2000). Support vector machines for automatic data cleanup. In *International Conference on Spoken Language Processing (ICSLIP)*, pag. 210–213. (Cited on page 33.)
- Ganguly, N., Deutsch, A., & Mukherjee, A. (2009). *Dynamics On and Of Complex Networks: Applications to Biology, Computer Science, and the Social Sciences*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser. (Cited on page 13.)
- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, 160:108–119. (Cited on pages 3, 33, 35, 39, 42 and 71.)
- Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2016). Noise detection in the meta-learning level. *Neurocomputing*, 176:14–25. (Cited on page 94.)
- Garcia, L. P. F., Lorena, A. C., & de Carvalho, A. C. P. L. F. (2012). A study on class noise detection and elimination. In *Brazilian Symposium on Neural Networks (SBRN)*, pag. 13–18. (Cited on pages 3, 9, 33, 34, 35 and 36.)
- García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining*. Springer. (Cited on page 34.)
- Giraud-Carrier, C. & Martinez, T. (1995). An efficient metric for heterogeneous inductive learning applications in the attribute-value language. Technical report, University of Bristol. (Cited on page 24.)
- Giraud-Carrier, C. G., Brazdil, P., Soares, C., & Vilalta, R. (2009). Meta-learning. In *Encyclopedia of Data Warehousing and Mining*, pag. 1207–1215. (Cited on pages 70 and 71.)
- Giraud-Carrier, C. G., Vilalta, R., & Brazdil, P. (2004). Introduction to the special issue on meta-learning. *Machine Learning*, 54(3):187–193. (Cited on page 67.)
- Hall, M. A. (1999). Correlation-based feature selection for machine learning. Technical report. (Cited on page 76.)

- Hickey, R. J. (1996). Noise modelling and evaluating learning from examples. *Artificial Intelligence*, 82(1-2):157–179. (Cited on page 9.)
- Hilario, M. & Kalousis, A. (2000). Quantifying the resilience of inductive classification algorithms. In *4th European Conference on Principles of Data Mining and Knowledge Discovery*, volume 1910, pag. 106–115. (Cited on page 71.)
- Ho, T. K. (2008). Data complexity analysis: linkage between context and solution in classification. In *Structural, Syntactic, and Statistical Pattern Recognition (SSPR)*, pag. 986–995. (Cited on page 12.)
- Ho, T. K. & Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300. (Cited on pages 4, 5, 10, 13, 18, 21, 30, 67 and 68.)
- Hodge, V. J. & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126. (Cited on page 3.)
- Hulse, J. V., Khoshgoftaar, T. M., & Huang, H. (2007). The pairwise attribute noise detection algorithm. *Knowledge and Information Systems*, 11(2):171–190. (Cited on page 3.)
- Hulse, J. V., Khoshgoftaar, T. M., & Napolitano, A. (2011). An exploration of learning when data is noisy and imbalanced. *Intelligent Data Analysis*, 15(2):215–236. (Cited on page 3.)
- Kalousis, A. (2002). *Algorithm Selection via Meta-Learning*. PhD thesis, University of Geneva, Faculty of Sciences. (Cited on page 73.)
- Kanda, J., de Carvalho, A. C. P. L. F., Hruschka, E. R., & Soares, C. (2011). Selection of algorithms to solve traveling salesman problems using meta-learning. *International Journal of Hybrid Intelligent Systems*, 8(3):117–128. (Cited on page 67.)
- Khoshgoftaar, T. M. & Rebours, P. (2004). Generating multiple noise elimination filters with the ensemble-partitioning filter. In *IEEE International Conference on Information Reuse and Integration (IRI)*, pag. 369–375. (Cited on page 42.)
- Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. Springer Series in Statistics. Springer. (Cited on pages 4, 10 and 18.)
- Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *10th European Conference on Machine Learning (ECML)*, pag. 4–15. (Cited on page 36.)

- Li, L. & Abu-Mostafa, Y. S. (2006). Data complexity in machine learning. Technical Report CaltechCSTR:2006.004, Caltech Computer Science. (Cited on page 12.)
- Lichman, M. (2013). UCI machine learning repository. <http://archive.ics.uci.edu/ml>. (Cited on pages 6, 24, 66 and 71.)
- López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141. (Cited on page 3.)
- Lorena, A. C., Costa, I. G., Spolaôr, N., & de Souto, M. C. P. (2012). Analysis of complexity indices for classification problems: Cancer gene expression data. *Neurocomputing*, 75(1):33–42. (Cited on page 26.)
- Lorena, A. C. & de Carvalho, A. C. P. L. F. (2004). Evaluation of noise reduction techniques in the splice junction recognition problem. *Genetics and Molecular Biology*, 27(4):665–672. (Cited on page 1.)
- Lorena, A. C. & de Souto, M. C. P. (2015). On measuring the complexity of classification problems. In *22nd International Conference on Neural Information Processing (ICONIP)*, volume 9489, pag. 158–167. (Cited on pages 13 and 93.)
- Lorena, A. C., Garcia, L. P. F., & de Carvalho, A. C. P. L. F. (2015). Adapting noise filters for ranking. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pag. 299–304. (Cited on pages 43 and 46.)
- Maciá, N. & Bernadó-Mansilla, E. (2014). Towards UCI+: a mindful repository design. *Information Sciences*, 261:237–262. (Cited on pages 24 and 26.)
- Maletic, J. I. & Marcus, A. (2000). Data cleansing: Beyond integrity analysis. In *Information Quality (IQ)*, pag. 200–209. (Cited on pages 1 and 2.)
- Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., & de Carvalho, A. C. P. L. F. (2015). To tune or not to tune: Recommending when to adjust SVM hyperparameters via meta-learning. In *International Joint Conference on Neural Networks (IJCNN)*, pag. 1–8. (Cited on page 67.)
- Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood. (Cited on page 70.)
- Miranda, A. L. B., Garcia, L. P. F., de Carvalho, A. C. P. L. F., & Lorena, A. C. (2009). Use of classification algorithms in noise detection and elimination. In *Hybrid Artificial Intelligence Systems (HAIS)*, volume 5572, pag. 417–424. (Cited on page 3.)

- Miranda, P. B. C., Prudêncio, R. B. C., de Carvalho, A. C. P. L. F., & Soares, C. (2014). A hybrid meta-learning architecture for multi-objective optimization of {SVM} parameters. *Neurocomputing*, 143:27–43. (Cited on page 67.)
- Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill series in computer science. McGraw Hill. (Cited on pages 17, 27, 29, 35, 36, 67 and 76.)
- Mollineda, R. A., Sánchez, J. S., & Sotoca, J. M. (2005). Data characterization for effective prototype selection. In *Pattern Recognition and Image Analysis*, volume 3523, pag. 27–34. (Cited on page 13.)
- Morais, G. & Prati, R. C. (2013). Complex network measures for data set characterization. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pag. 12–18. (Cited on pages 10, 21, 24 and 48.)
- Orriols-Puig, A., Macià, N., & Ho, T. K. (2010). Documentation for the data complexity library in C++. Technical report, La Salle - Universitat Ramon Llull. (Cited on pages 4, 13, 15, 18, 24 and 68.)
- Peng, Y., Flach, P. A., Soares, C., & Brazdil, P. (2002). Improved dataset characterisation for meta-learning. In *5th International Conference on Discovery Science (DS)*, volume 2534, pag. 141–152. (Cited on page 71.)
- Pfahring, B., Bensusan, H., & Giraud-Carrier, C. G. (2000). Meta-learning by landmarking various learning algorithms. In *17th International Conference on Machine Learning (ICML)*, pag. 743–750. (Cited on pages 67, 71 and 73.)
- Prudêncio, R. B. C. & Ludermir, T. B. (2007). Active learning to support the generation of meta-examples. In *17th International Conference on Artificial Neural Networks (ICANN)*, volume 4668, pag. 817–826. (Cited on page 71.)
- Prudêncio, R. B. C., Soares, C., & Ludermir, T. B. (2011). Uncertainty sampling-based active selection of datasetoids for meta-learning. In *21st International Conference on Artificial Neural Networks (ICANN)*, volume 6792, pag. 454–461. (Cited on page 71.)
- Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann, 1 edition. (Cited on pages 1 and 9.)
- Quinlan, J. R. (1986a). The effect of noise on concept learning. In *Machine Learning, An Artificial Intelligence Approach*, pag. 149–166. (Cited on pages 1, 11 and 48.)
- Quinlan, J. R. (1986b). Induction of decision trees. *Machine Learning*, 1(1):81–106. (Cited on pages 1, 9, 27, 29, 33, 35, 36 and 48.)

- Redman, T. (1998). The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 41(2):79–82. (Cited on page 2.)
- Redman, T. C. (1997). *Data quality for the information age*. Artech House, 1 edition. (Cited on page 2.)
- Reif, M. (2012). A comprehensive dataset for evaluating approaches of various meta-learning tasks. In *1st International Conference on Pattern Recognition Applications and Methods*, pag. 273–276. (Cited on page 70.)
- Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers*, 15:65–118. (Cited on page 69.)
- Rossi, A. L. D., de Carvalho, A. C. P. L. F., Soares, C., & de Souza, B. F. (2014). MetaStream: a meta-learning based method for periodic algorithm selection in time-changing data. *Neurocomputing*, 127:52–64. (Cited on pages 67 and 68.)
- Sáez, J. A., Galar, M., Luengo, J., & Herrera, F. (2016). INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control. *Information Fusion*, 27:19–32. (Cited on pages 3, 33, 42 and 91.)
- Sáez, J. A., Luengo, J., & Herrera, F. (2013). Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recognition*, 46(1):355–364. (Cited on pages 2, 3, 4, 10, 12 and 71.)
- Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203. (Cited on page 42.)
- Sahu, A., Apley, D. W., & Runger, G. C. (2014). Feature selection for noisy variation patterns using kernel principal component analysis. *Knowledge-Based Systems*, 72:37–47. (Cited on page 3.)
- Schubert, E., Wojdanowski, R., Zimek, A., & Kriegel, H.-P. (2012). On evaluation of outlier rankings and outlier scores. In *12th SIAM International Conference on Data Mining (SDM)*, pag. 1047–1058. (Cited on page 45.)
- Shanab, A. A., Khoshgoftaar, T. M., Wald, R., & Hulse, J. V. (2012). Evaluation of the importance of data pre-processing order when combining feature selection and data sampling. *International Journal of Business Intelligence and Data Mining*, 7(1-2):116–134. (Cited on page 2.)
- Shearer, C. (2000). The CRISP-DM model: The new blueprint for data mining. *Data Warehousing*, 5(4):13–22. (Cited on page 2.)

- Singh, S. (2003). PRISM: a novel framework for pattern recognition. *Pattern Analysis and Applications*, 6(2):134–149. (Cited on page 12.)
- Sluban, B., Gamberger, D., & Lavrač, N. (2010). Advances in class noise detection. In *19th European Conference on Artificial Intelligence (ECAI)*, pag. 1105–1106. (Cited on pages 3, 9, 33, 34, 35 and 37.)
- Sluban, B., Gamberger, D., & Lavrač, N. (2014). Ensemble-based noise detection: noise ranking and visual performance evaluation. *Data Mining and Knowledge Discovery*, 28(2):265–303. (Cited on pages 3, 9, 12, 33, 35, 37, 38, 42, 43, 44, 45 and 91.)
- Smith, M. R., Martinez, T., & Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256. (Cited on pages 1, 9, 12, 14, 23, 27, 33, 43, 91 and 93.)
- Smith-Miles, K. A. (2008). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):1–25. (Cited on pages xix, 67, 68, 69 and 70.)
- Soares, C., Brazdil, P., & Kuba, P. (2004). A meta-learning method to select the kernel width in support vector regression. *Machine Learning*, 54(3):195–209. (Cited on page 67.)
- Soares, C., Petrak, J., & Brazdil, P. (2001). Sampling-based relative landmarks: Systematically test-driving algorithms before choosing. In *Progress in Artificial Intelligence (EPIA)*, pag. 88–95. (Cited on pages 68 and 70.)
- Spolaôr, N., Cherman, E. A., Monard, M. C., & Lee, H. D. (2013). ReliefF for multi-label feature selection. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pag. 6–11. (Cited on page 46.)
- Strong, D. M., Lee, Y. W., & Wang, R. Y. (1997). Data quality in context. *Communications of the ACM*, 40(5):103–110. (Cited on pages 1 and 2.)
- Tanwani, A. & Farooq, M. (2010). Classification potential vs. classification accuracy: A comprehensive study of evolutionary algorithms with biomedical datasets. In *Learning Classifier Systems*, volume 6471, pag. 127–144. (Cited on page 94.)
- Teng, C.-M. (1999). Correcting noisy data. In *16th International Conference on Machine Learning (ICML)*, pag. 239–248. (Cited on pages 3 and 12.)
- Tomek, I. (1976). An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):448–452. (Cited on pages 3, 9, 33, 35 and 40.)

- Vanschoren, J. & Blockeel, H. (2006). Towards understanding learning behavior. In *15th Annual Machine Learning Conference of Belgium and the Netherlands*, pag. 89–96. (Cited on page 71.)
- Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). OpenML: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60. (Cited on page 71.)
- Vapnik, V. N. (1995). *The nature of Statistical learning theory*. Springer-Verlag. (Cited on pages 16, 17, 27, 29, 33, 35, 36 and 76.)
- Verbaeten, S. & Assche, A. V. (2003). Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems*, volume 2709, pag. 317–325. (Cited on pages 3, 9, 34 and 42.)
- Wang, R. Y., Storey, V. C., & Firth, C. P. (1995). A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):623–640. (Cited on pages 1, 2 and 9.)
- Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, 2(3):408–421. (Cited on pages 3, 33 and 40.)
- Wilson, D. R. & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286. (Cited on page 40.)
- Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2):241–259. (Cited on page 67.)
- Wu, X. (1995). *Knowledge Acquisition from Databases*. Tutorial Monographs in Artificial Intelligence. Greenwood. (Cited on page 1.)
- Wu, X. & Zhu, X. (2008). Mining with noise knowledge: Error-aware data mining. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(4):917–932. (Cited on pages 2, 3 and 34.)
- Yang, Y., Wu, X., & Zhu, X. (2004). Dealing with predictive-but-unpredictable attributes in noisy data sources. In *Knowledge Discovery in Databases (PKDD)*, volume 3202, pag. 471–483. (Cited on page 3.)
- Zhu, X., Lafferty, J., & Rosenfeld, R. (2005). *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science. (Cited on page 18.)
- Zhu, X. & Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210. (Cited on pages 2, 3, 11, 12, 24, 26 and 35.)

Zhu, X., Wu, X., & Chen, Q. (2003). Eliminating class noise in large datasets. In *20th International Conference on Machine Learning (ICML)*, pag. 920–927. (Cited on pages 3 and 12.)