Poznan University of Technology
Institute of Computing Science

Krystyna Napierała

Improving Rule Classifiers For Imbalanced Data

Doctoral Dissertation

Submitted to the Council of the Faculty of Computer Science of Poznań University of Technology

Supervisor: Ph.D., Dr. Habil., Jerzy Stefanowski

October 2012

Institute of Computing Science
Poznan University of Technology
Piotrowo 2, 60-965 Poznan, Poland
http://www.cs.put.poznan.pl

BibTeX:

```
@phdthesis{ key,
    author = "Krystyna Napiera{\l}a",
    title = "Improving Rule Classifiers For Imbalanced Data",
    school = "Poznan University of Technology",
    address = "Pozna{\'n}, Poland",
    year = "2012",
}
```

Contents

Preface

Extracting *knowledge* from the data is one of the fundamental tasks of *machine learning* and *data mining*. This knowledge can be used either to describe the unknown concepts and patterns in the data or to make predictions for the unknown observations. The latter task, called *supervised classification*, is the most common application of machine learning. Based on the historical data, for which the correct decisions (class labels or categories) are known, the learning system should produce a knowledge representation (e.g. decision rules) which can be used to assign new, unseen observations (examples) to the classes.

As an example, consider a problem of credit assignment. Historical data collected by a bank consists of the clients' profiles together with their credit risk (good or bad client). As the historical clients' records comprise only a small part of all possible profiles, the goal of a learning system is to generalize the learning data sample to be able to classify new potential clients as good (who should be allowed a credit) or bad (who should be refused).

Note that in the presented credit assignment problem, there is a high probability that the learning examples will unevenly represent both classes. In the bank's historical data, there will (hopefully) be much more data describing good clients than bad clients. Such an uneven distribution of learning examples is known as the *class imbalance* problem and it has been recently gaining much interest both from the research community and from the business sector.

It has been observed that the *of-the-shelf* learning algorithms do not work well with such data as they have been designed with the assumption that the distribution between the classes is approximately balanced. As a result, they tend to concentrate on recognizing the larger classes (called majority classes) and neglect the smaller (*minority*) class.

The need for addressing this problem is an important research challenge from a practical point of view, as the class imbalance problem has been reported in many application domains such as medicine (diagnosing rare ilnesses and assigning therapy or treatment), detecting fraudulent banking operations, detecting network intrusions, managing risk, predicting failures of technical equipment or information filtering. In all those applications the correct classification of the minority examples is of key importance. For instance, a failure in recognizing an illness and not assigning a proper treatment is much more dangerous than misdiagnosing a healthy person, whose diagnosis can be corrected in an additional examination.

The aim of this thesis is to provide some insights and solutions to the problem of learning *decision rules* from imbalanced datasets. We concentrate on the decision rules, as it is one of the most popular knowledge models, due to its intuitive and natural representation. Moreover, rule learning algorithms are sensitive to the imbalance problem. Although some modifications of either the rule induction phase or of the classification strategy have already been proposed, we think that their effectiveness is not sufficient. Further research on constructing efficient rule classifiers for imbalanced domains is still needed.

First, we would like to carry out a thorough analysis of a problem at hand, to show *why* the learning algorithms have difficulties with imbalanced data. We will identify the problems

on data characteristics level as well as on algorithmic level. The first ones are related to the characteristics of the data distribution in imbalanced datasets, which particularly negatively affect learning. The problems on algorithmic level are related to the construction of the standard rule learning algorithms which may cause the undesired bias towards majority classes. A theoretical analysis will be backed up by a set of experiments carried out on specially designed artificial datasets as well as on real-world datasets.

Then, based on the results of this analysis, we propose some approaches to improve learning rules from imbalanced data. We introduce a new learning algorithm, BRACID, which comprehensively addresses the data-level and algorithmic-level problems with class imbalance. Another proposal, ABMODLEM algorithm, uses expert argumentation to explain the decisions for the selected problematic learning examples to tackle the problem of recognizing correctly the minority class examples. The usefulness of these solutions will be verified in the extensive computational experiments.

Acknowledgments. This research has been supported by the "Scholarship support for PH.D. students specializing in majors strategic for Wielkopolska's development" (sub-measure 8.2.2 Human Capital Operational Programme, co-financed by European Union under the European Social Fund) and partly by the Ministry of Science and Higher Education, grant no. N N519 441939.

I would like to thank my supervisor, Professor Jerzy Stefanowski, for his invaluable guidance and support, insightful remarks and inspiring discussions, which gave shape to this thesis.

Chapter 1

Introduction

1.1 Problem Setting

Classification. Machine learning is a subfield of computer science and artificial intelligence. It concerns designing systems that can learn from experience. According to Tom Mitchell's definition [90], a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

The learning algorithm uses a set of previously observed training examples (also called learning examples, objects or observations), described by a finite set of attributes (called conditional attributes). Attributes can be defined on different domains, usually numeric (integer or real values), ordinal or nominal.

In the problem of classification, the examples are described by an additional class (decision) attribute defined on a finite domain. Here we consider a problem of supervised learning, in which the value of this attribute is known a-priori for the training examples. Such data is fed to the learning system, which creates a model that can be used to predict a value of the class attribute (or class label) for unseen objects, using the known values of conditional attributes. A supervised learning system that can perform classification is called a classifier.

More formally, given a set of learning examples described by vectors of conditional attributes $\mathbf{x} = (x_1, x_2, ...x_n) \in X$, where X is called attribute space, and class labels $y \in Y$, where $Y = \{y_1, y_2, ...y_k\}$, a learning task is to construct a classifier which represents a mapping f from X to Y ($f(x) = y$) that allows for future prediction of class y based only on the observation of $\mathbf{x}$.

Several methods have been proposed in the literature to solve this task, from more statistical analysis, through symbolic approaches such as classification trees, to more complex support vector machines or ensemble techniques. Among these approaches, no dominant solution can be distinguished. For their review, see e.g. [90].

Rule-based learning. There are many representations in which the learned model can be expressed. This thesis focuses on the rule-based representation, which is one of the most popular symbolic representations of knowledge discovered from data. The rule-based classifier consists of a set of rules, which are represented as symbolic expressions of the following form:

$$IF\ (conditions)\ THEN\ (target\ class)$$

or, more formally, $P \mapsto Q$, where conditions P are formed as a conjunction of elementary tests w on values of attributes describing the learning examples

$$P = w_1 \wedge w_2 \wedge \cdots \wedge w_i$$

The rule consequence Q indicates the assignment of an example satisfying the condition part of the rule to a given class $y \in Y$.

We concentrate on rule-based classifiers for several reasons. First of all, it is claimed that they are more comprehensible and human-readable than other representations, in particular "black box" representations such as neural networks or support vector machines. Individual rules constitute "blocks" of knowledge, which can be easily analysed by human experts. Additionally, there exists a direct relation of each rule to facts (examples) in the training data. Such comprehensibility and explicability of the rule representation is highly appreciated when constructing intelligent systems, as it often results in an increased willingness of decision makers to accept the provided suggestions and solutions (e.g. applications in medicine [87]).

Although a tree representation shares similar characteristics, a set of rules is typically more compact [87, 109, 150] than a comparable decision tree. Rule representation can be also more powerful because it is not constrained by the arborescent structure of the tree. That is why tree-based classifiers are often converted a posteriori to the set of rules.

Finally, rules have been successfully used in many applications, see e.g. [70, 97] or some chapters in [87, 119, 65].

Class imbalance. A dataset is considered to be imbalanced if it is characterized by an unequal distribution of examples between the classes. The smaller class is called the *minority class*, while the other classes are called *majority classes*. In the imbalanced datasets, the minority class is usually of primary interest to the decision maker, i.e. not recognizing the minority class examples is much more serious than raising so called "false alarms" (assigning a majority example to the minority class). For this reason, the most popular performance measures such as total accuracy are not useful in the context of class imbalance, as they are biased towards the majority classes. Thus, for imbalanced domains other performance measures have to be used (they will be described in detail in Section 2.2).

There is no unique opinion about the degree of such imbalance between the class cardinalities. Some researchers have studied the datasets where one class was several times smaller than other classes, while others have considered more severe imbalance ratios as, e.g., 1:10, 1:100 or even greater. Without naming the precise values of this ratio, we repeat after [139] that the problem is associated with lack of data (absolute rarity), i.e. the number of examples in the rare (minority) class is too small to detect properly the regularities in the data. This kind of data characteristics is also called *between-class imbalance* [60].

The imbalance of a learning dataset can be either intrinsic (in the sense that it is a direct result of the nature of the data space) or it can be caused by too high costs of acquiring the examples from the minority class, e.g. due to economic or privacy reasons [139].

Although in this work we usually consider binary (two-class) problems in which there is one minority and one majority class, the problem may in general concern also the multiclass data in which imbalance exists between various classes.

1.2 Motivations

The difficulties in learning from imbalanced data have been encountered in many domains of application, but the techniques applied were rather simple, such as modifying prior distributions in Bayesian classifiers or transforming the problem to the cost-sensitive learning. Since the end of 90s, an interest in this problem has grown and new methods have been introduced. In general, two classes of approaches may be disstinguished: methods on data level try to artificially re-balance the

learning data, while methods on algorithmic level modify the learning algorithms. Among different methods proposed, it is difficult to indicate the best approach. Typically, a given method is shown to outperform other on a group of imbalanced problems, while on a different set of problems, chosen for an experimental setup in a different publication, it performs worse. Therefore, it is interesting to look for the settings in which a given method should be used to improve learning from imbalanced data. Also, there is still an interest in finding the reasons for the difficulties in learning from imbalanced data.

The reasons why learning from imbalanced data is problematic are more complex than they at first have seemed to be. Initially, the difficulty was attributed solely to the rarity of one of the classes [56]. However, some researchers have observed that imbalance ratio itself may not be a problem when the classes are clearly separated – they pointed out that the difficulties arise only when other data characteristics occur *together* with class imbalance (see e.g. one of the first works – "Class Imbalances: Are we Focusing on the Right Issue?" [56]). Up to now, several issues related to the distribution of examples in the imbalanced datasets were named, such as overlapping of examples from different classes [44, 9] or small disjuncts (decomposition of a class into smaller sub-concepts, in which the number of examples is too small – also called *within-class imbalance* [60]). We will review them in detail in Section 2.1. However, a further research in this topic is stilll needed.

The above mentioned data factors have been shown to deteriorate the classifiers' performance using the specially prepared artificial datasets, in which the data distribution was known a-priori. Using these datasets, the researchers could draw interesting conclusions, for instance how different classifiers behave in face of different data distributions. However, what is still missing in our opinion, is a method to shift these observations to the real-world datasets. In other words, there is a need for a method which could identify these data factors in the datasets in which the distribution of examples in the attribute space in not known a-priori. Such a method (or a set of methods) would, first, help to confirm that the discovered data characteristics are common in real-world problems and, second, give a possibility to analyse the dataset before deciding which learning method is the most suitable for a particular dataset.

Imbalance-related problems affect various types of classifiers; in fact, none of them are completely insensitive to the class imbalance. This also refers to the rule-based classifiers, which reveal an undesirable bias towards the majority classes [22]. As mentioned before, rule representation is especially useful in the domains where the explicability of the classifier's decisions is important, such as medicine or banking. At the same time, in these domains class imbalance is often an intrinsic characteristics of the learned problem. For this reason, we think that improving rule-based classifiers to better deal with imbalanced data is a particularly important research problem. Although some attempts have already been made to improve rule classifiers, in our opinion they lack the understanding of the complex characteristics of the imbalanced data, the key properties of its underlying distribution and their consequences (we will discuss it in Section 4.5). We think that there is still a place for new algorithms that could resolve these issues in a more comprehensive way.

1.3 Aims and Objectives

The thesis is devoted to learning rules from imbalanced datasets. The general goal of the thesis is formulated as follows:

> Analyse factors on data-level and on algorithmic-level which make learning rules from imbalanced data difficult; based on these observations, introduce new rule learning

techniques, which are more efficient than the existing solutions in terms of performance measures dedicated for class imbalance.

Four major objectives can be distinguished within this goal. We characterize them briefly below, giving reference to the chapters in which they are achieved.

Study of data-level sources of difficulty. Although some studies of the influence of data factors on the learning abilites have already been carried out, we think that there is no unifying framework, analysing and comparing all these factors together. Moreover, there is a lack of methods which would allow to carry out this analysis on real-worlds datasets. Chapter 3 is devoted to this objective. We distinguish four types of minority class examples: safe, borderline, rare and outliers. We introduce a method to identify these types of examples in real-world datasets, which is based on analysing the local neighbourhood of minority examples. Additionally, we show how to use the visualization methods, which allow to present multi-dimensional data on the two-dimensional graphs, to analyse the distribution of examples in imbalanced datasets. Using these methods we show that real-world imbalanced datasets can have different proportions of the four distinguished types of examples. Considering these observations in a comprehensive experimental study allows us to differentiate the performance of popular classifiers as well as of the preprocessing methods. Moreover, by analyzing the accuracies for each type of testing examples we can identify the sources of difficulties for the classiffiers, in particular rule-based ones, and the areas of competence for the preprocessing methods.

Study of algorithmic-level sources of difficulty. Classic rule-based learners were designed with the assumption that the distribution between the classes is balanced. In Chapter 4 we provide a comprehensive analysis of standard approaches to rule learning with reference to class imbalance problem. We show how different stages of creating a rule classifier – from the sequential covering induction technique, through measures used to evaluate rules, to classification strategies – are implicitely biased towards the majority classes. We also review the existing approaches to improve rule learning, which try to address these issues.

Bottom-up induction of Rules And Cases for Imbalanced Data. Based on the analysis from the two previous chapters, in Chapter 5 we introduce a new rule induction algorithm, BRACID, which aims at improving the classiffication performance of classifiers learned from imbalanced data. It achieves this goal by changing these phases of the induction process which might be biased towards the majority class, such as greedy sequential covering, top-down induction technique and classification strategy. Moreover, it takes into account different types of learning examples and processes them differently. In Chapter 6 the usefulness of the proposed BRACID algorithm is evaluated in a series of experiments conducted on 22 imbalanced datasets. We compare it against popular rule induction algorithms as well as the selected specific approaches dedicated for handling the imbalanced data. We also analyse, for which types of examples (defined in Chapter 3), BRACID is the most competent classifier.

Using expert argumentation for learning rules from imbalanced data. Using expert knowledge to direct the rule induction can help to obtain rules more consistent with the domain knowledge, and as a result more intuitive and acceptable to the decision maker. It can be useful also for learning from imbalanced data. When the learning minority examples are rare and they represent very sparsely the whole attribute space, it may be difficult for the learner to construct the correct hypothesis, and the learner's bias towards the majority class may become even more evident.

In Chapter 7 we adapt the paradigm of argument-based learning for the imbalanced domain. In this approach, an expert can give additional arguments for the selected difficult examples, explaining the decision taken for them. Such arguments are then used in the induction of rules. We adapt this paradigm to the MODLEM rule learning algorithm, and propose an ABMODLEM extention of it (argument-based MODLEM). We also propose how to automatically select a small number of most critical examples which should be explained by an expert. In the experimental study (presented in Chapter 8) we show that such argumentation can improve both the interpretability of rules and the classification performance, especially when the minority class is concerned.

The main achievements related to these four objectives have been published in the scientific journals. The list of publications is listed in Appendix B.

Chapter 2

Basic Concepts of Learning from Imbalanced Data

Class imbalance has been observed in many application domains. For example, in detection of frauds in telephone calls [35] and credit card transactions [18] the number of legitimate transactions is much higher than the number of fraudulent ones. In direct marketing, where the goal is to identify likely buyers of certain products and adjusting the promoting of the products, the minority class representing a response rate for marketing campaigns is also small (about 1% [75]). Class imbalance is also an intrinsic property of medical datasets, e.g. predicting pre-term birth [47]. Other examples include detecting oil spills from satellite images [68], telecommunication equipment failures [140], network intrusion, managing risk or information filtering – for their review see, e.g., [19].

In these and other works it has been shown that when the learning set is imbalanced, standard classifiers have a difficulty in correctly recognizing the minority class. For instance, in [141], the authors analysed the relationship between the imbalance ratio and the performance of classification trees and showed that the minority class predictions are more error-prone. Japkowicz *et al.* in [2] carried out an analogous analysis for Support Vector Machines, also showing that their performance deteriorates with a growing imbalance ratio.

The problem of dealing with class imbalance has been receiving a growing research interest from academia and industry, which is reflected by a growing number of publications in this topic. Figure 2.1 presents the estimation of the number of publications concerning imbalanced learning (based on the number of publications with a "class imbalance" keyword stored in the Assocation for Computing Machinery database, ACM). A high activity in this field is also reflected by a number of conferences, special sessions and workshops dedicated for this problem, e.g. the International Conference on Machine Learning workshop on Learning from Imbalanced Data Sets (ICML'03), special issue on Learning from Imbalanced Data Sets in ACM SIGKDD Explorations Newsletter 2003 or the most recent Workshop on Class Imbalances: Past, Present, Future on the ICMLA-2012 conference. In this Chapter we present the current understanding of the imbalanced learning problem and review the state-of-the-art solutions proposed to address it.

2.1 Nature of the Problem

It has been shown that class imbalance ratio is not the only factor that impedes learning. The experimental studies carried out, e.g., in [9, 59, 141] suggest that when there is a clear separation of the decision classes, the definition of both classes can be correctly learnt regardless of the imbalance ratio. These works showed that the *data complexity*, understood here as the distribution of examples from both classes in the attribute space, has a crucial impact on learning. It is not

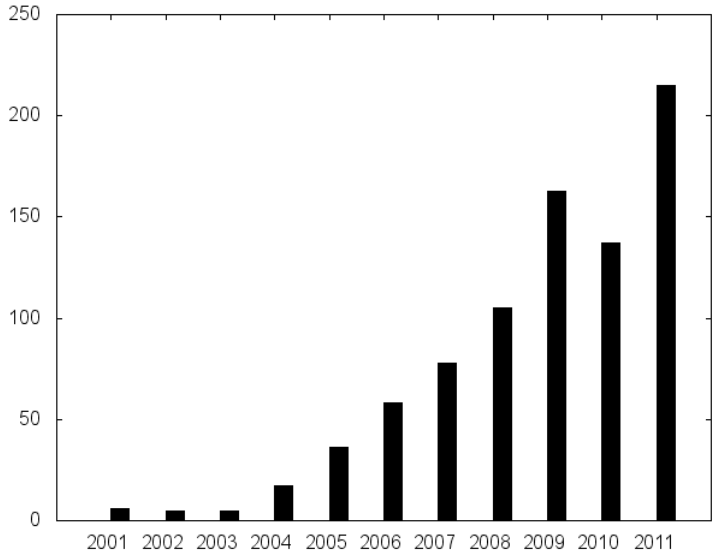


Figure 2.1: Number of publications on imbalanced learning.

particularly surprising, as it could be expected that data complexity should affect learning also in balanced domains. However, when data complexity occurs *together* with the class imbalance factor, the deterioration of classification performance is amplified and it affects mostly (or even only) the minority class.

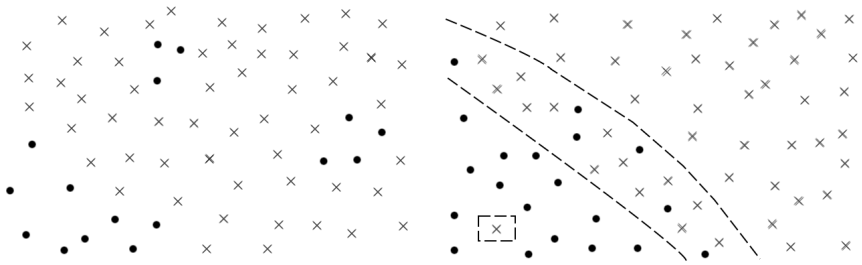
The term "data complexity" can comprise different data distribution patterns. Up to now, the researchers have distinguished several factors which hinder learning in imbalanced domains, such as overlapping, small disjuncts, outliers or noise. We describe them briefly below.

Overlapping between the classes

In the boundary regions between classes, the examples from different classes may overlap (see Fig. 2.2b – black circles represent minority examples). In such case, it is difficult for the learner to decide where exactly the border line separating the class definitions should be placed. As the minority class is underrepresented in the dataset, it will most probably be underrepresented also in the overlapping region. As a result, the learners will have a tendency to shift the border definition too close to the minority class, treating the whole overlapping area as belonging to the majority class definition. Indeed, the experiments on mainly artificial data with different degrees of overlapping showed that overlapping deteriorated the classifier performance, especially when the minority class was concerned. What is more, increasing the overlapping of classes was more critical for the recognition of the minority class examples than increasing the overall imbalance ratio [44, 9] - we will discuss it more in Section 3.1.

Data decomposition leading to small disjuncts

Another difficult distribution of the data concerns the situation when a class is scattered into smaller sub-parts representing separate sub-concepts. Japkowicz in her research named it *within-class imbalance* [60]. This is closely related to the problem of small disjuncts (see Fig.2.2a). Briefly speaking, a classiffier learns a concept by generating disjunct forms (represented as rules [55]) to describe it. Small disjuncts are these parts of the learned classiffier which cover a too small number of examples [139]. It has been observed in the empirical studies that small disjuncts contribute to the classiffication error more than larger disjuncts [108].



(a) Small disjuncts in the minority class (b) Overlapping and noisy examples

Figure 2.2: Difficult data distributions in imbalanced datasets

Although a problem of within-class imbalance may occur in both minority and majority classes, small disjuncts are more characteristic and more critical for a minority class. In a majority class, the sub-concepts will be most often represented by a vast number of examples forming large disjuncts, while in the minority class, in which the examples are already rare, their further decomposition into several sub-concepts will produce small disjuncts, represented by a too small number of examples to be correctly learnt. Japkowicz and co-authors in their experiments with artificial data showed that a high level of decomposition combined with a too small number of examples in the minority class resulted in a poor recognition of this class [56, 60]. At the same time, they showed that for much larger datasets with low level of decomposition or with a suficient number of examples in the sub-concepts, the imbalance ratio alone did not decrease so much the classification performance.

Presence of noisy/outlying examples

Single examples from one class, located far from the decision boundary inside the other class, are usually called in the literature noisy examples. However, according to the definition, e.g., in [73], such examples may in fact be a result of three kinds of data imperfections:

- noise, i.e. random errors in training examples and background knowledge
- insufficiently covered example space, i.e. too sparse training examples from which it is difficult to reliably detect correlations,
- inexactness i.e. an inappropriate description language which does not contain/facilitate an exact description of the target concept.

Other works distinguish between class noise (erroneous classification labels), attribute noise (erroneous attribute values) and outliers (non-typical class representatives) [41].

Handling noise (in its broad definition described above) is also an important issue in imbalanced data. Noisy majority examples are particularly harmful for the minority class, as they can cause the fragmentation of the minority class and increase the difficulties in learning its definition. On the other hand, for distant minority examples surrounded by the majority class examples, it is important to distinguish outliers from noise as distant minority examples might often be a result of the *insufficiently covered example space* rather than of random errors in the training data.

Learning systems usually have a single mechanism for dealing with the three kinds of imperfect data [73] – after the identification of "suspicious" examples, such instances are eliminated from the learning set or class (attribute) values are corrected or these examples are neglected in the learning phase (e.g. by using pruning in the decision trees). These approaches have been shown to improve the total classification accuracy in the classic learning perspective (see a review in [7]). However,

in the class imbalance setting, using standard approaches for handling noise "can be catastrophic", as pointed out e.g. in [133]. It may lead to removal or relabeling of most minority examples, or to pruning all the decision rules for the minority class. The study in [16] showed that when there is an abundance of data, it is better to detect properly "bad data" at the expense of throwing away "good data", while in case when the data are rare, more conservative filters are better. Therefore, some specialized approaches dedicated for noise in imbalanced datasets have been proposed, e.g., in [133], which treat differently the minority and majority distant examples. We will return to this topic in Section 3.1.

2.2 Evaluating Classifiers Learned from Imbalanced Data

Evaluation measures, reflecting the classification abilities of a classifier learned from imbalanced data, are usually designed for two-class problems in which class labels for the minority and majority classes are called positive and negative, respectively. In case when a dataset contains several majority classes, they can be aggregated into one negative class. Thus, the performance of a classifier can be presented in a confusion matrix as in Table 2.1.

Table 2.1: Confusion matrix for performance evaluation

	Predicted Positive	Predicted Negative
True Positive	TP	FN
True Negative	FP	TN

As it is more straightforward to compare the performance of classifiers using single values rather than comparing the matrices, several point measures are created from the confusion matrix. Four simple measures concerning the recognition of the positive and negative classes, are:

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$False\ Positive\ Rate = \frac{FP}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

True Positive Rate is also called *Sensitivity* and *Recall*. True Negative Rate is called *Specificity*.

In balanced domains, the most popular assesment measures are global accuracy and error rate:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$Error\ rate = 1 - Accuracy = \frac{FP + FN}{TP + FN + FP + TN}$$

However, as described in Section 1.1, in imbalanced domains the recognition of minority class examples is more important than the recognition of a majority class. We expect from a classifier to provide a good recognition of the minority class, even at a cost of misclassifying some majority examples. Therefore, performance metrics such as a total accuracy or an error rate do not provide the desired information about a classifier as they are sensitive to the imbalance ratio and are biased towards the majority class – a conventional classifier which can recognize correctly all majority examples and no minority examples, will achieve very high accuracy (e.g. 90% if the imbalance ratio in a dataset is 1:9), even though such a classifier would present no value for a decision maker.

Thus, when evaluating a classifier learnt in the imbalanced environment, alternative performance measures are needed, in which minority and majority classes are treated independently. Often, simply Sensitivity and Specificity measures are looked at separately to compare two classifiers. However, these measure usually comprise a trade-off – the improvement on the minority class accuracy usually comes at a cost of deterioration of a majority class accuracy. As a result, looking at these two measures separately it is difficult to decide which classifier is better, as one of them will usually win on one of the measures, and another one on the other measure. Therefore, aggregative measures which consist of two measures representing both classes were proposed. Kubat and Matwin [69] proposed to use the geometric mean of Sensitivity and Specificity defined as:

$$G\text{-}mean = \sqrt{Sensitivity \cdot Specificity}$$

This measure promotes the classifiers which maximise the recognition of both minority and majority classes while keeping these accuracies balanced. An important, useful property of G-mean is that it is independent of the distribution of examples between classes [53]. An alternative criterion is the F-measure, aggregating Precision and Recall:

$$F\text{-}measure = \frac{(1+\beta)^2 \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}$$

where β is a coefficient expressing the relative importance of Precision and Recall (typically $\beta = 1$). For discussion of its properties see, e.g., [53].

For the classifiers which yield a probabilistic decision representing a degree to which an example is a member of a class, such as, e.g., Naive Bayes or Neural Networks, a threshold can be used to produce a series of evaluation measures such as TP- and FP-$rate$, which can produce graph-based evaluation measures as the ROC curve (plot of TP-$rate$ over the FP-$rate$). Generally speaking, one classifier is better than another if its ROC curve is over the other curve (see Figure 2.3). A random classifier produces a ROC curve located on a diagonal, therefore a classifier with a curve under the diagonal performs worse than random guessing [34]. To quantify the ROC curve results, a point measure representing the area under curve (AUC) is often used. A random classifier obtains AUC equal to 0.5, while for a perfect classifier it equals to 1. AUC is said to give more weight to the correct classification of the minority class, thus outputting fairer results than the classification accuracy [58]. There are also some modifications to AUC, for instance a weighted AUC proposed in [142]. It addresses the undesired characteristic of AUC which values equally the performance of a classifier in the high TP-$rate$ region and in the low TP-$rate$ region while only the former one is truly interesting to the decision maker. In weighted AUC, more emphasis is put on the performance in the high TP-$rate$ regions.

Other curve measures proposed for the class imbalance setting are PR-curves and Cost Curves. PR-curves plot Precision over Recall and are claimed to be a better measure than ROC for highly imbalanced data for which the FP-$rate$ used in ROC curves does not change significantly. Cost Curves, on the other hand, plot the classification performance over varying misclassification costs and class distributions. They are more useful than ROC curves if one wants to answer the question "for what class probabilites one classifier is preferable over the other" [58]. For the review of these measures, see e.g. [53].

Although the graph-based measures are applicable rather for probabilistic classifiers than for the deterministic ones [58], the deterministic classifiers can be sometimes adapted to give probablistic answers (for instance, pruned decision trees can estimate the probability according to the proportion of examples from different classes in the leaves), or the classifier can be evaluated on a dataset in which the class imbalance is step-wise changed by sampling, producing a point on a graph for each imbalance ratio [53].

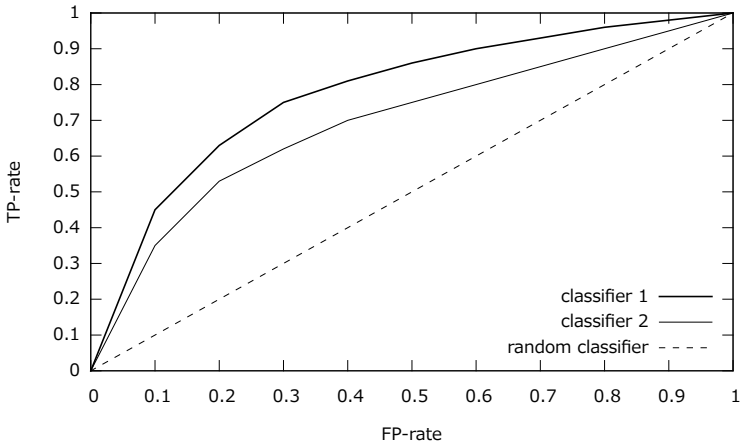


Figure 2.3: ROC curve. Ideal point is (1,1). Classifier 1 is better than classifier 2; both are better than random guessing.

2.3 Review of Existing Methods

Solutions proposed for the class imbalance problem are often divided in the literature into methods on data level and methods on algorithmic level. Most of the research concentrates on the data level techniques, which resample the original learning data to balance the classes, so that the learning could be performed using classic, well-studied learning algorithms. Less numerous works concern the algorithmic-level solutions, which modify the techniques used in the learning algorithms and the classification strategies.

We do not intend to present here a complete review of all methods and algorithms representing these two types of approaches. We rather want to describe the main areas of research and send the reader to the more comprehensive review works, such as [53, 44]. The categorization of the most popular techniques, together with the most representative algorithms, is presented on Figure 2.4. In these families of approaches, we will describe some methods, focusing on the proposals which we will use in our work (either as inspiration or as comparatory solutions).

2.3.1 Methods on Data Level

Sampling methods modify an imbalanced dataset to provide a more balanced distribution. Balancing the examples in the classes can be obtained by undersampling the majority class and/or oversampling the minority class. Although it has been argued whether balancing the data set to the proportion 1:1 between the classes leads to the best results [136], in general changing the class distribution towards a more balanced one improves the performance for most datasets and classifiers [53]. We describe the most well known sampling methods below.

Simple under- and oversampling

Random oversampling consists in replicating randomly chosen minority examples, while *random undersampling* removes randomly selected majority examples from the original dataset. While both methods can balance the original dataset to a desired level, it should be remembered that they have their consequences. Removing the majority examples by means of undersampling may cause the classifier to miss some important subconcepts of the majority class if a random selection will pick too many examples representing this concept. Oversampling, on the other hand, may lead to ovefitting, especially if multiple copies of noisy examples are introduced. A rule learnt from

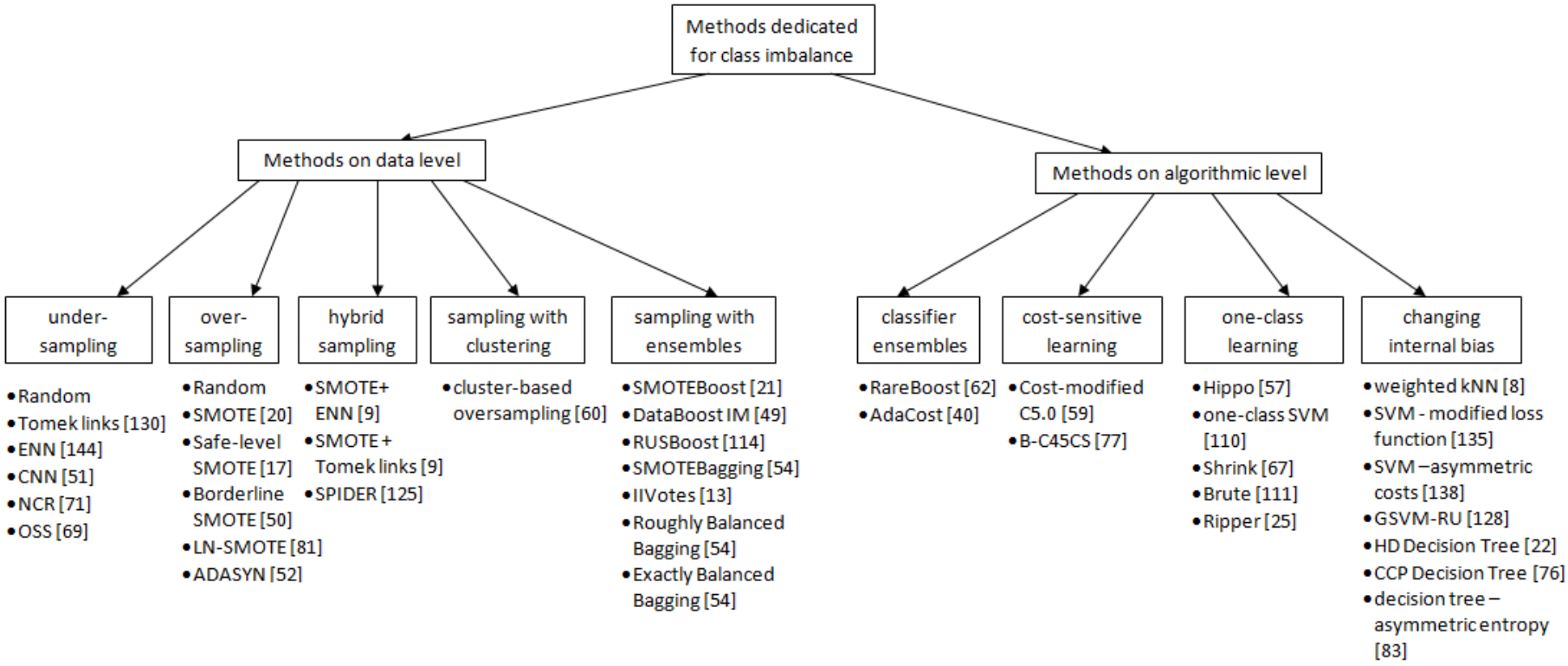


Figure 2.4: Classification of existing methods

such examples will appear confident and accurate, while it may in fact cover only one learning example.

Informed undersampling

To overcome the problems with simple random undersampling, informed undersampling is used to remove only the redundant majority examples. It is usually based on analysing the local neighbourhood of learning examples, to differently process the examples representing the overlapped, noisy or safe regions. In these neighbourhood-based approaches, a measure estimating the distance between the examples (described by nominal and numerical attributes) is needed. We will come back to this topic in Section 2.4, assuming now that such a measure is given. There are also some proposals which estimate the nature of examples without the use of distance measure – for instance, in [112, 133], a classifier's ability to recognize correctly the examples is used to identify difficult (noisy) examples which should be removed.

The first group of informed undersampling methods aims to remove the noisy and overlapping majority examples. *Tomek links* [130] is one of such methods. Tomek link can be defined as a pair of examples belonging to different classes with distance between them equal to $d(E_i, E_j)$ if there are no other examples E_k such that $d(E_i, E_k) < d(E_i, E_j)$ or $d(E_j, E_k) < d(E_i, E_j)$. If two examples form a Tomek link, than either one of them is noise or both lie in the borderline region, where overlapping may be present. In such case, the majority example from such pair is removed to move the decision border further from the minority class.

Edited Nearest Neighbour method (ENN [144]) also tries to discard unreliable majority examples, by removing any majority examples whose class label differs from the class of at least two of its three nearest neighbors. *Neighbour Cleaning Rule* method (NCR, proposed by Laurikkala in [71]) modifies ENN to clean even more majority examples. Similarly to ENN, if a majority example is surrounded by at least two minority examples, it is removed. Additionally, if a minority example is surrounded by at least two majority examples, than its majority neighbours are also removed. The effect of this preprocessing method is presented on Figure 2.5 – compare the original dataset (Fig. 2.5a) with the the result of preprocessing using NCR (Fig. 2.5d).

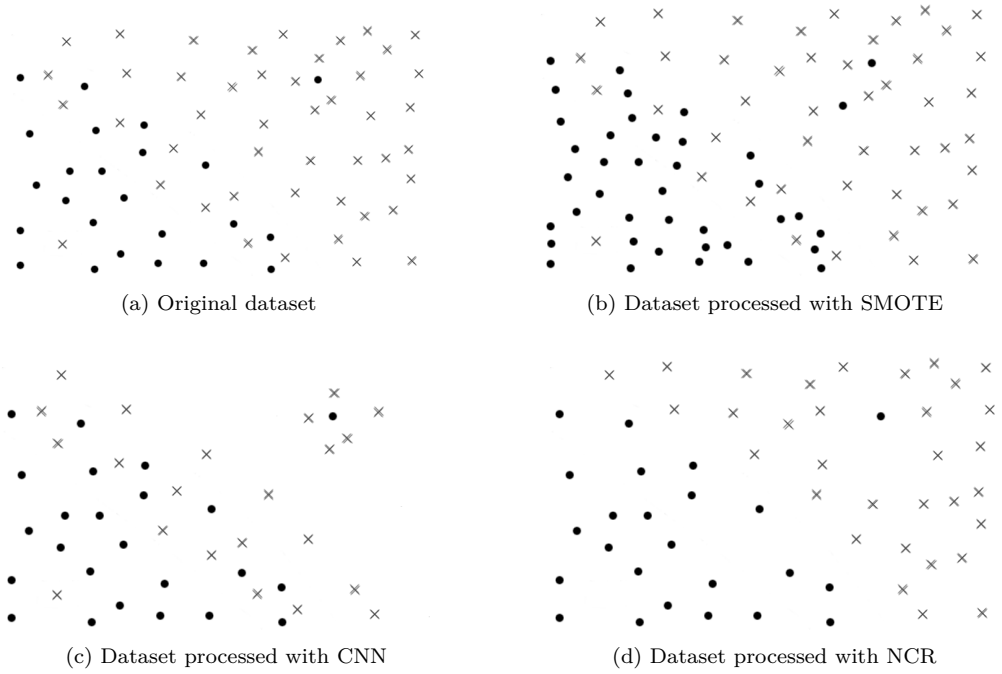
Other methods try to eliminate the examples from the majority class which lie in the homogeneous, safe regions and are distant from the decision border, considering these examples less relevant for learning. An example of such method is *Condensed Nearest Neighbour* Rule (CNN [51]), which preserves only the majority examples misclassified by k nearest neighbours (in the original proposal, $k = 1$) – compare Fig. 2.5a with Fig. 2.5c. *One-sided selection* method (OSS [69]) integrates Tomek links with CNN. First, CNN is used to remove redundant (safe) examples. Then, majority examples participating in Tomek links (representing overlapping and/or noisy examples) are discarded.

Informed oversampling

The most well-known representative of this approach is SMOTE [20]. To avoid overfitting which may occur when the exact copies of minority examples are added to the original learning set, this method introduces new *synthetic* minority examples. Specifically, for each minority example E_i it finds k-nearest minority neighbours, and generates the synthetic examples in the direction of some or all of them, depending on the amount of the oversampling required. A new artificial example is introduced on a randomly chosen point on the line joining the selected example and its neighbour. In this way, SMOTE allows the classifier to build larger decision regions that contain the nearby examples of the minority class [20].

Although SMOTE can significantly improve learning, its main drawback is that when generating the synthetic examples, it does not take into consideration the neighbouring examples from other

classes, which can increase the overlapping of classes and introduce additional noise (compare Fig. 2.5a with Fig. 2.5b). As a result, several extensions of this algorithm have been proposed. Borderline-SMOTE [50] generates synthetic instances only for minority examples close to the decision border. ADASYN [52] generates more synthetic examples from the examples which are "harder to learn" and fewer examples from the easier learning instances. Safe-level SMOTE [17] and LN-SMOTE [81] try to generate new synthetic examples in the direction of the regions populated by the minority class, to avoid introducing artificial examples inside the majority class regions.



(a) Original dataset (b) Dataset processed with SMOTE

(c) Dataset processed with CNN (d) Dataset processed with NCR

Figure 2.5: Comparison of different preprocessing methods

Sampling with clustering

Cluster-based sampling addresses the problem of *within-class imbalance*, when the minority class is separated between several smaller sub-concepts. *Cluster-based oversampling* method (CBO [60]) puts more emphasis on oversampling the smaller subconcepts than larger ones. It first uses the K-means method to identify clusters in the dataset and then oversamples each cluster so that all the clusters from a given class are of the same size (to overcome the problem of *within-class imbalance*), assuring at the same time that the distribution of examples between the classes is even (addressing the problem of *between-class imbalance*).

Hybrid informed resampling

These methods combine the use of oversampling with cleaning. In [9], two such methods were proposed and evaluated, which integrate SMOTE with ENN and SMOTE with Tomek links. In these algorithms, cleaning is done for both classes as a post-processing after oversampling with SMOTE, to additionally remove the artificial minority class examples introduced too deeply in the majority class space. Another hybrid method, SPIDER, was proposed in [125]. It also analyses the local neighbourhood of examples, but contrary to the previous two methods it does not introduce any artificial examples and it cleans only the majority class. In this method, noisy majority examples are removed, while borderline and outlying minority examples are replicated. Contrary to, e.g., SMOTE, the number of example copies is not constant among the whole dataset – it

depends on the number of neighbours from the opposite class and the idea is to introduce such a number of copies so that the original example was correctly classified by its nearest neighbours.

Sampling with ensembles

Finally, there are some more complex algorithms which integrate different kinds of resampling with either boosting or bagging. For instance, SMOTE-Boost [21] integrates SMOTE with AdaBoost introducing synthetic sampling at each boosting iteration to make the successive classifier ensembles focus more on the minority class. Building each component classifier on a different data sample helps to build well-defined regions for the minority class [53]. Other examples of integrations with boosting are DataBoost-IM [49] or RUSBoost [114]. There are also some approaches which combine sampling with bagging, e.g. IIVotes (IVotes combined with SPIDER), SMOTEBagging (bagging with oversampling) or Exactly Balanced Bagging and Roughly Balanced Bagging [54] (bagging with special undersampling). For their review, see e.g. [40].

2.3.2 Methods on Algorithmic Level

Methods on algorithmic level adapt the existing algorithms and techniques to the problem of class imbalance. The most popular groups of methods are cost-sensitive learning, one-class classifiers and classifier ensembles. We will review them briefly below. The discussion of modifications proposed for rule-based classifiers is shifted to Section 4.5.

Cost-sensitive learning

Cost-sensitive learning assumes that together with the learning examples, different misclassification costs are specified for different classes. The goal of learning is to minimize the total misclassification cost. Although the problem of class imbalance is not the same as cost learning as, in general, the costs of misclassification for the minority and majority classes are unknown, the cost-sensitive learning framework can be adapted for class imbalance [82]. For instance, Japkowicz and Stephen propose a cost-sensitive C5.0 decision tree in which they estimate the costs based on the class imbalance ratio [59], assigning higher costs for false negatives than for false positives. Liu and Zhou in B-C45CS algorithm, based on the C4.5 decision tree, normalize the error costs in terms of the number of examples in each class [77].

One-class learning

Systems trying to learn the definition for both classes may tend to ignore the minority class. Therefore, some researches have proposed to use a one-class learning paradigm, in which the goal is to recognize only the minority class objects and distinguish them from the majority class. There are some solutions which adopt this recognition-based approach for neural networks (Hippo [57]), Support Vector Machines [110] or rules (Shrink [67], Brute [111], Ripper [25]). Rule approaches will be described in more detail in Section 4.5.

Changing the internal bias

Some techniques try to internally bias the learning algorithm to take into account the characteristics of the data distribution in imbalanced datasets. For instance, in [8], a distance function in k-nearest neighbour classifier is modified to compensate for the fact that minority examples are rare and thus have a smaller chance to belong to a neighbourhood of a classified example. In this approach, a weighting factor is introduced to the distance function depending on the class of the learning example, so that the distance to the minority examples becomes smaller. In Support Vector Machines, on the other hand, different loss functions are used for both classes, to push the hyperplane further from the minority class [135]. Other approaches directly modify the weights of

slack variables in the objective function for the quadratic programming – see e.g. [91] or even more complex solutions in [137, 138]. Some other works integrate re-sampling of examples (by SMOTE or undersampling) inside SVM – see e.g. [128].

There are also some works concerning decision trees. They focus mostly on the evaluation measure used during the generation of a tree to choose the best splitting condition. For example, in [83] the authors point out that in classic decision trees an entropy measure is often used to evaluate the splitting condition, which assigns the worst value when a branch of a tree covers an equal number of examples from both classes. In class imbalance setting however, such leaf may carry an interesting information to the user. Therefore, the authors propose to use a new evaluation measure based on an asymmetric entropy measure. Cieslak and Chawla in [22, 76] show that another popular evaluation measure used in decision trees, Information Gain, is sensitive to class imbalance. They propose to use instead a Hellinger distance measure (HD [22]) and show it to be insensitive to the class imbalance ratio. In [76] they propose yet another measure – Class Confidence Proportion (CCP).

Classifier ensembles

Classifier ensembles are known to improve the accuracy of single classifiers by combining them together. Their adaptation for the class imbalance problem includes not only the resampling of the training samples as described before, but there are also some proposals which modify the learning procedure itself (for a review see [40]), for instance by applying cost-sensitive learning in the ensemble framework. A representative of this approach is RareBoost [62], which gives a different treatment to positive and negative predictions by assigning different weights to the examples in the subsequent iterations, depending on the class of the example. Here the weights of false-positive examples are scaled in proportion to how well they are distinguished from true-positive examples, while false-negative examples are scaled in proportion to how well they are distinguished from true-negative examples. Other cost-sensitive approaches which modify the weight update formula include AdaCost, AdaC1, AdaC2 or AdaC3. Their review can be again found in [40].

2.4 Measuring the Distance Between the Examples

Many methods presented in this Chapter (especially the methods on data level, described in Section 2.3.1) are based on analysing a local neighbourhood of examples. Also, the methods introduced by us in the next chapters will be based on it. Measuring the distance between the examples which are described by both numeric and nominal attributes is not trivial and can influence the results of the methods such as informed preprocessing methods, k-NN classifiers etc. Although measuring the distance between the examples is a general problem, not related directly to the class imbalance problem, we have decided to discuss it here in a separate section, as we will refer to this topic in almost all the subsequent chapters.

The basic distance measures assume that the examples are defined only on numeric attributes. The distance between the examples x and y can be then defined with a standard *Euclidean* distance measure. It is calculated as

$$D(x,y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

where k is the number of conditional attributes and x_i and y_i are the attribute values of the examples x and y, respectively. Other measures for numeric attributes include *Minkowski* or *city-block* (also called *Manhattan*) measures – see review of these and other measures in [145, 84].

To handle the datasets with both numeric and nominal attributes, so-called *heterogeneous distance functions* that use different attribute distance functions on different kinds of attributes, have been proposed. The HEOM measure (Heterogeneous Euclidean-Overlap Metric) uses a normalized Euclidean distance for numeric attributes. For nominal ones, the attribute distance equals 1 if the attribute values are equal and 0 otherwise. Unknown attribute values are handled by returning an attribute distance of 1. Other simple heterogeneous distance functions include Gower, ER, GEM – for their review, see [80].

The HVDM (Heterogenous Value Difference Metric) differs from the above measures in how it treats the nominal attributes. Instead of simple value matching, HVDM makes use of the class information to compute conditional probabilities by using a Stanfil and Valtz value difference metric for nominal attributes [145, 115]. For numeric attributes, it uses a normalized Euclidean distance. HVDM is defined as:

$$D(x,y) = \sqrt{\sum_{i=1}^{k} d_i(x_i,y_i)^2}$$

All distances for single attributes are normalized in range 0 to 1. If one of the attribute values of x_i, y_i is unknown, the distance d_i is equal to 1. The distance for nominal attributes is defined as:

$$d_i(x_i,y_i) = \begin{cases} 0 & if \quad x_i = y_i \\ svdm & if \quad x_i \neq x_i \end{cases}$$

Value difference metric (a simplified form without tuning attributes' weights) is defined as [115]:

$$svdm = \sum_{l=1}^{k} \left| \frac{N(x_i,K_l)}{N(x_i)} - \frac{N(y_i,K_l)}{N(y_i)} \right|$$

where k is the number of classes, $N(x_i)$ and $N(y_i)$ are the numbers of examples for which the value on i-th attribute is equal to x_i and y_i respectively, $N(x_i,K_l)$ and $N(y_i,K_l)$ are the numbers of examples from the decision class K_l, which belong to $N(x_i)$ and $N(y_i)$, respectively.

The attribute distance for numeric attributes is defined as

$$d_i(x_i,y_i) = \frac{|x_i - y_i|}{4\sigma_a}$$

where σ_a is a standard deviation of the numeric values of attribute a.

Among the distance measures, HVDM is one of the most popular and is often claimed to perform better than other heterogeneous functions. For instance, in [80] the authors compared the performance of the Euclidean distance, HEOM, Gower, ER, HVDM and GEM distance measures used with 3-NN classifier on 12 medical datasets, and concluded that HVDM was better that the other measures on the *True Positive Rate*. In [72], where HVDM was compared against other distance measures on 21 datasets, HVDM was shown to treat the nominal attributes more appropriately than the other compared functions, which resulted in a better performance on total accuracy and on TPR (the differences were statistically significant according to the Wilcoxon test).

Let us mention that, despite the popularity of HVDM, other more sophisticated measures have been proposed. IVDM and WindowedVDM, proposed by Wilson and Martinez in [145], calculate the attribute distance for numeric attributes in a more advanced way, by discretising them to calculate sample probabilities. The main drawback of these methods (including also HVDM) is that they assume that attributes are independent of each other. Thus, two highly correlated attributes will contribute twice as much to the distance value as they should [84]. Therefore, more sophisticated methods have been proposed, which use the regular simplex methods

or covariance measures for the nominal attributes to compute the Mahalonobis-type distance [84]. The experimental comparison of these new methods to the HVDM method showed however, that HVDM performed quite well in most cases, even when the assumption of attribute independence was violated.

HVDM measure was also succesfully used in the related studies concerning class imbalance – especially in the preprocessing methods which analyse the local neighbourhood of examples (e.g. in SMOTE, SPIDER, NCR). Therefore, we will use the HVDM measure in this thesis.

Chapter 3

Types of Examples and Their Influence on Learning of Classifiers

In this Chapter we analyse the relationship between the types of examples in imbalanced data and the performance of learning methods. As described in Chapter 2, learning from imbalanced data with clearly separated classes is not difficult for most classifiers. Recognizing the minority class becomes more difficult when the distribution of examples from different classes is more complex. A number of experimental studies can be found in the literature, which show that the mutual position of examples has a crucial impact on learning from imbalanced data [44, 69].

In Section 3.1 we review the most important related works, which experimentally evaluate the impact of different data types on learning of classifiers and on the preprocessing methods. Most of these works are carried out on artificial datasets in which the distribution of the data is known a-priori and can be precisely controlled. Although they focus on the important aspects of data distribution, such as, e.g, the role of small disjuncts in the minority class, their conclusions might not be evident and easy to directly apply in the real-world settings, as it may be difficult to precisely estimate the occurence of these data factors in the real-world datasets.

Some comparative studies are carried out with real-world datasets [7, 63, 108], however they focus rather on studying single data factors or just compare the performance of the selected classifiers. They usually do not consider several data factors occuring together and do not propose straight-forward methods for the identification of these types of examples in the real data.

Therefore, we look for new simple techniques which could help to identify the difficult types of example distributions in imbalanced data and which could lead to new studies on their influence on learning typical classifiers and the main preprocessing methods. We propose first to adapt the visualisation methods to confirm the occurence of different types of examples in real-worlds datasets and then we introduce a method of their identification in the datasets, based on analysing the local neighbourhood of examples.

We focus on, so-called, *safe*, *borderline* and *outlier* minority examples. We also introduce an additional type of examples, called *rare* examples, which in our opinion can also influence the learning methods. We will show that in real-world imbalanced datasets, the classes can rarely be clearly separated. In most datasets a mixture of types of examples can be observed. Then, we will relate the results of the preprocessing methods and classifiers' performance to the results of the analysis of the data distribution, showing which types are the most difficult and which methods are sensitive to the particular types of examples.

3.1 Experimental Perspectives on Types of Examples – Literature Study

In this Section we review the most important related works which study the properties of the imbalanced data distributions and their consequences for the learning classifiers and the preprocessing methods. First, let us discuss the types of examples which have been distinguished and studied in these works.

The most common is the distinction between *safe* and *unsafe* examples [69]. *Safe* examples are located in the homogenous regions populated by the examples from one class only. Otherwise they are treated as *unsafe*. Unsafe examples are considered to be less reliable and are often further discriminated between *borderline* and *noisy* examples [69]. *Borderline* examples are placed close to the boundary regions between classes. They can be unreliable as even a small amount of attribute noise can send the example to the wrong side of the decision border [69], resulting in the overlapping of classes. Singular examples located deeper in the regions where the opposite class prevails are usually treated as noisy examples. Finally, examples forming small separated groups, called in [59] *small disjuncts*, have been distinguished – their definition was introduced in Section 2.1.

These four types of examples (safe, borderline, noisy, small disjuncts), together with such factors as imbalance ratio, size of the datasets and size of the minority class, have been a subject matter of several experimental studies concerning class imbalance. Let us briefly review the main works.

Impact of decomposition of the minority class into small disjuncts

Japkowicz and Stephen in [59] carried out a large number of experiments with simulated data studying the relationship between the fragmentation of the class, the size of the training set and the class imbalance ratio. By introducing these three types of disturbance and manipulating with their degree, their influence on the recognition of minority classes and on the abilities of particular classifiers were analysed. The experiments were carried out on artificial data described by one numerical attribute. The minority class was initially divided into two separate intervals, and was systematically partitioned into smaller sub-groups (interpreted as small disjuncts). The experimental results showed an important role of sparsity of the minority class when it contains very small sub-groups. Jo and Japkowicz in [60] conducted a comparative study (also on artificial, one-dimensional data) of various sampling methods with a recommendation for using informed cluster-based techniques. A relationship between the class imbalance and small disjuncts was investigated also in [108] using real-world UCI datasets and the C4.5 tree-based classifier. Here, the leaves covering only few examples were treated as representatives of small disjuncts and the authors measured (with the so-called *error concentration* measure) if the incorrectly labeled testing examples are a result of using these leaves for classification. Their results suggest that pruning may not be effective for dealing with small disjuncts and class imbalance, that SMOTE might increase the error concentration around small disjuncts and that simple random oversampling may sometimes compete with more advanced preprocessing methods.

Impact of borderline examples and overlapping

The role of overlapping has been a subject matter of many experimental studies concerning class imbalance. In [107], the authors used the artificial, five-dimensional datasets described by numerical attributes. Minority and majority classes formed two spherical clusters. By changing the imbalance ratio and the distance between the clusters (from the separated clusters to the entirely overlapped ones), the authors analysed the relationship between these two factors and their influence on the C4.5 classifier with respect to the AUC measure. They concluded that when the clusters

were separated, even a high imbalance ratio did not deteriorate AUC. Increasing overlapping was more influential than increasing class imbalance, leading to the deterioration of AUC.

An analogous experiment, but comparing more classifiers on more evaluation measures, was carried out in [44]. Here, two-dimensional artificial datasets described with numerical attributes were generated, and the degree of overlapping and imbalance ratio were systematically changed. The classes formed two rectangular regions, which were overlapped by moving them along the x-axis. Six learning algorithms were compared (1-NN, Naive Bayes, C4.5 tree learner, Support Vector Machine and Neural Networks) using Specificity and Sensitivity measures. Increasing overlapping between the classes degraded the recognition of the minority class more than changing the imbalance ratio. However, it affected various classifiers in a different degree. In case of a very high overlapping, 1NN performed the best on the minority class but degraded the most the recognition of the majority class, while SVM was the worst classifier on the minority class. In the additional experiment, the authors manipulated with the imbalance ratio in the overlapping region, showing that when the minority examples prevailed in this region, the majority class started to be recognized poorer than the minority class. The same experimental setup was then used to analyse in more detail the kNN classifier, with k changing from 1 to 15 [43]. It showed that when the overlapping increased, more local classifiers (with smaller k) performed better on the minority class.

The study in [30] focused on the effects of overlapping and class imbalance on Support Vector Machines. Using the two-dimensional artificial datasets, the authors showed that when the overlap level was high, it was unlikely that collecting more training data would produce a more accurate classifier. They also observed that the performance of SVM decreased gradually with the increasing imbalance ratio and overlapping, and that there was a sudden drop when the imbalance ratio equaled to 20% and the overlapping level exceeded 60%, regardless of the training set size. After this threshold, SVM failed to recognize the minority examples.

Finally, in [121] the effect of overlapping was studied together with other factors such as decomposition of the minority class into smaller sub-concepts. The experiments were carried out on artificial two-dimensional datasets with more complicated non-linear borders. The experiments showed that the combination of class decomposition with overlapping makes learning very difficult. Increasing overlapping was more influential than increasing the number of subconcepts.

Impact of noise

Some other experimental studies concern the role of noisy examples in learning from imbalanced data. Anyfantis et al. [7] evaluated the effectiveness of techniques for handling class noise in imbalanced datasets using the C4.5, Naive Bayes and 5NN as classifiers and the G-mean performance measure. They carried out the experiments using 7 UCI real-world datasets which were considered noise-free, and the class noise was introduced by randomly relabelling some of the learning examples. In [63], class noise was also introduced to real-world datasets by randomly relabelling the learning examples. The experimental results showed that all learners were sensitive to noise, however some of them, as Naive Bayes and nearest neighbor learners, were often more robust than more complex learners such as support vector machines or random forests. In [112], the impact of noise was studied on both artificial and real-world datasets. In real-world data, the authors introduced both class noise and attribute noise, by either changing the class label or the attribute values, respectively. The comparison concerned the SMOTE preprocessing method and its several extensions, used with several clasisifiers and evaluated with an AUC measure. It showed that SMOTE was sensitive to the noisy data and its extensions, cleaning the additional noise introduced by SMOTE, were necessary.

Impact of imbalance ratio

Van Hulse et al carried out a large study with 35 real-world datasets, 11 classifiers and 7 preprocessing methods in [132]. They grouped the datasets into 4 categories with respect to the imbalance ratio and compared the learning strategies within these categories. According to the authors, random undersampling worked better than other approaches for data with the most severe imbalance ratio ($< 5\%$). Unlike other studies, they claimed that simpler random re-sampling often performed better than more sophisticated informed re-sampling methods. Having many experimental configurations, the authors drew a conclusion that algorithms respond differently to various preprocessing methods (e.g. results for decision trees are not valid for neural networks) and it depends on the evaluation measures (e.g. G-mean or F-measure show higher improvements than AUC).

Yet another study concerning impact of imbalance ratio was carried out in [10]. 20 real-world datasets were used to analyse the behaviour of 7 learning methods on the AUC measure. Averaging the results for all the datasets, the authors have observed that the loss of performance started to be significant when the minority class represents 10% of the data or less. SVM was less affected by the class imbalance ratio than other classifiers for all except the most imbalanced distributions. Then, they analysed the performance of two preprocessing methods, random oversampling and SMOTE, and concluded that the preprocessing methods usually could not improve the performance by more than 30%.

Impact of data size

Batista et al in [9] developed a wide systematic experimental study with 15 real-world UCI datasets and 10 different preprocessing methods used with the C4.5 decision trees. The oversampling methods provided better AUC than undersampling ones. Considering data factors, the authors took into account the data size and claimed that SMOTE combined with informed undersampling (ENN or Tomek links) led to the best results for smaller data with few minority examples while simple random oversampling was competitive to other methods for datasets containing a high number of the minority examples (>100 according to the authors).

3.2 Rare and Outlying Examples

In our opinion, in the above works concerning the role of types of examples in learning from imbalanced data, not enough attention has been focused on the singular minority examples distant from the decision border. Single minority examples surrounded by many examples from majority classes were usually treated as noise (more precisely, class or attribute noise). As a result, they were usually removed from the data [7, 63]. However, as the minority class can be underrepresented in the data, these examples can be outliers, representing a rare but valid subconcept of which no other representatives could be collected for training. This opinion was expressed e.g. in [69], where the authors suggested that minority examples should not be removed as they are too rare to be wasted, even under the danger that some of them are noisy. In [149], which concerns the detection of noise in balanced datasets, the authors suggest to be cautious when performing automatic noise correction, as it may lead to ignoring outliers which is "questionable, especially when the users are very serious with their data". In our opinion, the minority class examples conform to this case. The interesting results can be found in [41]. Although this work was not carried out in the context of class imbalance, it concerns medical domains in which class imbalance often occurs. The authors have consulted the results of noise identification filter with an expert to verify if the identified examples were rather noise or outliers. In some datasets a large number of examples was denoted by an expert as outliers (e.g. 11 out of 13). Their removal would be harmful for the

learner. The authors suggested that examples representing real noise should be eliminated from the training set, whereas outliers should be, after hypothesis construction, added as exceptions to the generated rules.

On the other hand, distant majority examples located in the minority class regions are most probably a true noise. Since the majority class is well represented in the dataset, such example is rather not a representant of a very rare subconcept, but more likely a result of a assigning a wrong class label (*class noise*) or a faulty measurement of some conditional attributes (*attribute noise*). Noisy majority examples are undesired as they can cause the fragmentation of the minority class and increase the difficulties in learning its definition. Therefore, in our work, we will treat the distant minority examples as outliers which should be kept in the learning set, while majority examples as noise which should be removed.

Moreover, it would be worth to distinguish yet another type of so-called *rare examples*. These are pairs or triples of minority class examples, located in the majority class region, which are distant from the decision boundary so they are not borderline examples, and at the same time are not singular examples, so they are not exactly outliers. The role of these examples has been preliminary studied by us in the experiments with special artificial datasets [102, 121]. It has been shown that rare examples significantly degraded the performance of classifiers. Also, various preprocessing methods (based on oversampling or undersampling) performed differently on such rare examples.

3.3 Identifying Types of Examples in Real-world Datasets

3.3.1 Motivations

Related works, discussed in Sections 3.1 and 3.2, showed that the deterioration of classification performance in imbalanced datasets is related not only to the class imbalance ratio, but that the data distribution characteristics, such as small disjuncts, overlapping, noisy, rare and outlying examples, have a crucial impact on learning, especially when the minority class is concerned. These dependecies have been examined using mostly artificial datasets [121, 107, 44], in which the data distribution was given a priori and the degree of each factor could be precisely controlled by augmenting or diminishing the degree of overlapping [107, 44], the number and cardinality of small disjuncts [59, 60] or noisy/outlying examples [112]. The datasets used were usually one- or two-dimensional, the examples were mostly described by only numerical attributes and they formed very basic shapes (rectangles, spheres). Some works were carried out using real-world datasets [7, 63, 108, 112]. In most of them the experimental setup assumed that the original datasets were free of the analysed factor, e.g. class noise, and the dataset was then artificially disturbed, e.g. by relabelling a certain number of examples to introduce class noise. Based on the results of these experiments the researchers were able to draw conclusions about, e.g., a level of certain disturbance which is critical for the performance of a given classifier or a preprocessing method. Thanks to these works, we have now a broader view of the data distribution characteristics in imbalanced data which influence the performance of learning methods and of the mutual dependencies between these data factors.

However, what in our opinion is missing to broaden this picture, is the analysis of the *natural* underlying distribution of real-world imbalanced datasets. If, for example, the results presented in [30] suggest that the performance of SVMs is seriously downgraded when the number of overlapping examples in the imbalanced dataset exceeds 60%, it would be interesting to know if such strong overlapping often occurs in real-world applications. What is more, if we could estimate the data distribution in the real-world dataset, e.g. estimate the degree of overlapping in some learning

problem at hand, we could use the conclusion of [30] and not apply SVM to this dataset. Moreover, by analysing a representative collection of real-world imbalanced datasets, we could observe what are the most common data distribution patterns - e.g. if noise can be observed across many datasets, what percentage of the dataset it can constitute – and which data distribution factors often appear together. Such knowledge would help to point out the most promising directions for the development of methods dedicated for class imbalance.

Some simple data distribution factors which are known to influence learning from imbalanced data, can be directly measured in the real-world datasets - e.g. imbalance ratio, data set size or minority class size. As a result, the observations from the experimental studies carried out in the related works which are based on these characteristics (e.g. [10, 132, 9]) can be easily applied to the real-world settings. However, the data distribution factors which have been shown to influence learning more than imbalance ratio or dataset size – such as noise or overlapping – are not that straightforward to measure. Therefore, we think that there is a need for methods which could help estimate the presence and level of these factors in the real-world datasets.

Some works have been proposed which try to tackle this problem. For example, in [30] (concerning the effects of overlapping and imbalance on the SVM classifier), the authors propose to estimate the degree of overlapping in real-world datasets by measuring a number of support vectors which can be removed from the classifier without deteriorating the classification accuracy. In [134], on the other hand, a method for estimating the number of noisy examples was proposed. It uses ensemble methods, such as cross-validated committees, bagging and boosting, to identify the examples which cannot be correctly classified by all or the majority of the classifiers built on parts of the training set. The conclusion was that using the majority vote gives good results for large datasets, while for small datasets or when the examples are costly, conservative approaches (in which all the classifiers have to mislabel the example) are better.

However, these methods usually measure only one characteristics of the dataset. To analyse the dataset set in a more comprehensive way, one would have to apply these methods one by one (using the methods described above, it would require building an SVM classifier to estimate overlapping, and building an ensemble of classifiers to evaluate the presence of noise). We would like to propose a single method, which evaluates the occurrence of all the interesting data factors at once. What is more, we would like to keep the method simple and intuitive. We do not want to propose a method which is directly related to a particular classifier, but rather analyse the mutual positions of the learning examples in the attribute space. We will be interested in four types of minority examples: safe, borderline, rare and outlying examples. An illustrative artificial dataset containing all these types of examples is presented in Fig. 3.1a (the dataset concept comes from our earlier paper [102]). The minority class (black circles) is divided into five sub-concepts (clusters). In each of these concepts, the examples lying near the center of the cluster can be considered as *safe*. Many more examples belong to the border between the classes, in which the majority examples overlap with the minority ones. Finally, there are some examples more distant from the clusters, which represent *outliers* or *rare examples*.

3.3.2 Data Visualisation

Data visualisation techniques can be used to gain insight into a structure of multidimensional data. A term "structure" is understood here as geometric relationships between the examples. Examples of structure include clusters, regular patterns, outliers, distance relations, proximity of data points etc. [27]. The real-world problems are often described by more than two attributes, so to visualize such data, methods which can project multidimensional data points into a lower dimensional space such that the structural properties of the data are preserved, have to be used. Projection methods

create new dimensions (artificial attributes), which should aggregate well the original attributes. A large number of dimensionality reduction techniques have been proposed – their review can be found in [74]. One of the most popular technique is Multidimensional Scaling (MDS). It performs a linear mapping of dimensions with the aim of preserving the pairwise distances between data points in the original high dimensional data space in the projected low dimensional space [26]. Let us remark that while projecting a dataset to a lower dimensional space using MDS, it is important to monitor how much it preserves the data variance after the projection. If the preserved variance is too small, than there is a risk that too much information has been lost because the number of dimensions used was too low, and that the resulting visualisation is not reliable.

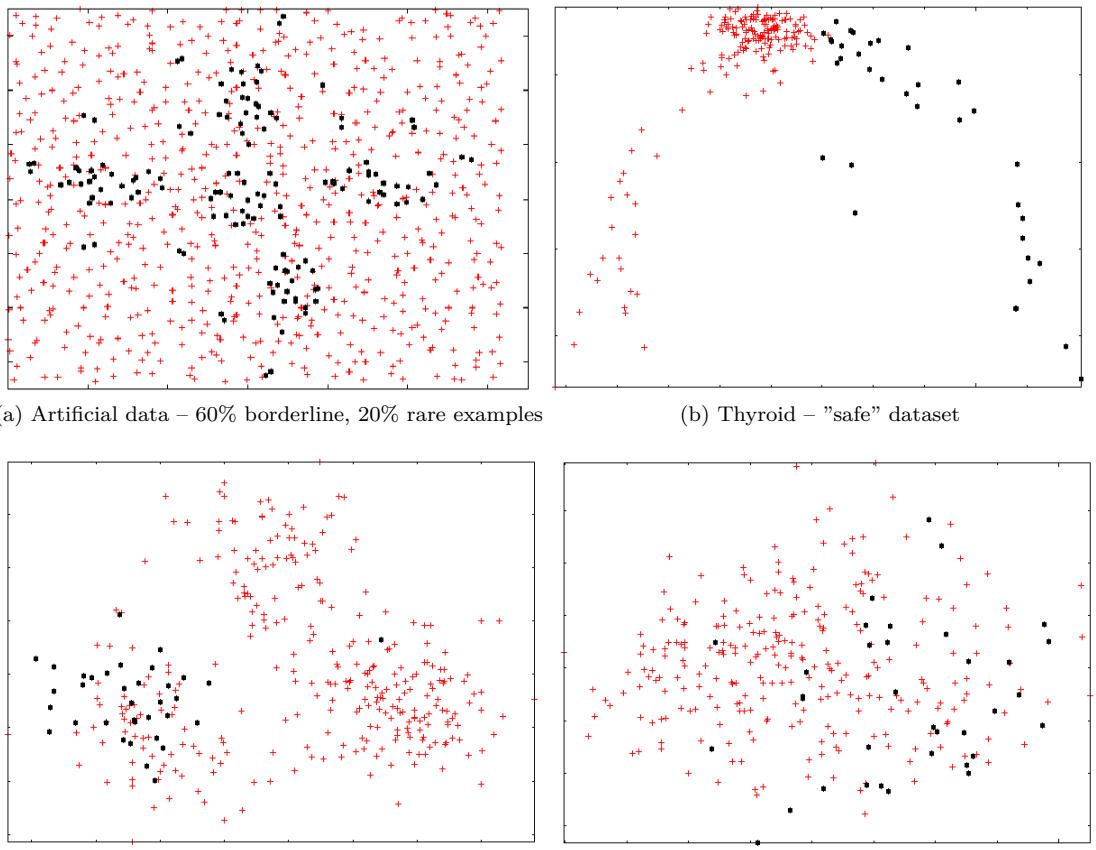
Despite its popularity, the MDS method is sometimes criticized for focusing too much on keeping the dissimilar datapoints far apart. For high-dimensional data, it is usually more important to keep the very similar points close together, which is typically not possible with a linear mapping [74]. Among many non-linear methods proposed (see their review in, e.g., [74]), t-SNE method (*t-Distributed Stochastic Neighbour Embedding*) is one of the most recent dimensionality reduction methods, which does not concentrate on preserving *all* the pairwise distances, but puts more emphasis on preserving *local* distances to keep similar examples together, rather than on preserving the exact distances between dissimilar examples [131]. T-SNE is a modification of the earlier SNE projection method, aiming to correct the drawback of "crowding" the examples too much in the center of the map. According to the experiments in [131], t-SNE is capable of capturing much of the local structure of the high-dimensional data very well, while also revealing global structure such as the presence of clusters.

Projection methods can be a helpful tool to visually inspect the distribution of examples in the multidimensional imbalanced data. By using a projection to a 2-dimensional space, we can plot the dataset and observe if data factors such as overlapping, rare or outlying examples are present in the dataset. As most of the real-world datasets are described by both numeric and nominal attributes, we calculate the distances between the points using the HVDM metric – see a discussion and justification of this choice in Section 2.4. To verify if the observed distribution of a dataset is not a result of the applied projection method rather than an intrinsic characteristics of the data, we apply on the same datasets the MDS and t-SNE projection, which are based on different principles, and compare the results.

We present the visualisations after the MDS projection of three imbalanced datasets from the UCI repository[1], often used in the experimental studies concerning class imbalance: thyroid, ecoli and cleveland (Fig. 3.1b-3.1d). For these datasets, the percentage of preserved variance was high enough to analyse the data. Looking at Fig. 3.1b-3.1d, one can notice that the three datasets are of different nature. In thyroid dataset (Fig. 3.1b), the classes are clearly separated (even linearly), so most of the minority examples represent safe examples. In ecoli dataset (Fig. 3.1c) on the other, the classes seriously overlap. The consistent region belonging solely to the minority class (on the very left) is rather small – most examples lie in a mixed region between the classes. Finally, the cleveland dataset (Fig. 3.1d) is even more difficult to learn, as the minority class is very scattered – the examples form very small groups of few examples and some of the other are singular observations, surrounded by the opposite class. This dataset consists mostly of rare examples and outliers.

Fig. 3.2 presents the results for thyroid and ecoli datasets after the t-SNE projection (run with the default parameters). The cleveland dataset is not used in this comparison as t-SNE method does not handle nominal attributes. It can be observed that the dimensions to which the datasets were projected are different, e.g. for ecoli, the t-SNE visualisation is rotated. Also, the mutual

[1] `http://www.ics.uci.edu/~{}mlearn/MLRepository.html`



(a) Artificial data – 60% borderline, 20% rare examples

(b) Thyroid – "safe" dataset

(c) Ecoli – "borderline" dataset

(d) Cleveland – "rare" / "outlier" dataset

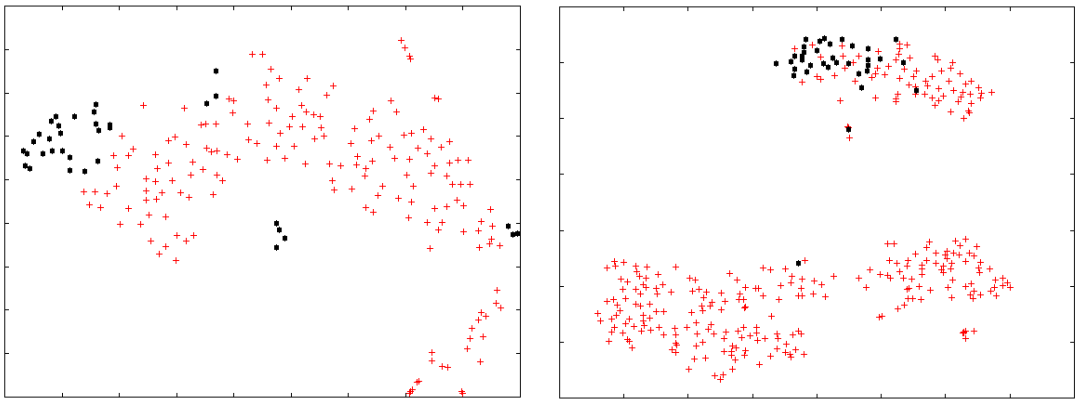
Figure 3.1: MDS visualisation of selected imbalanced datasets

positions of examples differ – the three clusters in the t-SNE projection of ecoli dataset are better separated than in the MDS method (which is consistent with the assumptions of t-SNE), and in the thyroid dataset the minority class forms several clusters instead of one. However, for both datasets the principle observations of distribution characteristics remain the same: in the thyroid dataset the classes can be easily separated, while in the ecoli dataset, in one of the clusters the examples from both classes strongly overlap.

To conclude, the visualisation methods can help to inspect the distribution of examples in the real-world dataset and estimate the types of minority examples. However, the applicability of these methods is limited. First of all, they cannot be used for very large datasets, as the visualisation of thousands of points would be difficult to read. Secondly, the projection to two dimensions may not always be feasible, as the dataset may be intrinsically characterized by more dimensions. For instance, we could not use the MDS technique to visualize another well-known imbalanced dataset from the UCI repository, *hepatitis*, as MDS with two dimensions preserved only 25% of variance in the dataset. Finally, some methods, such as t-SNE, were designed to use the numeric datasets. Therefore, there is still a need for more flexible methods, which can estimate the characteristics of the dataset. We propose such a method below.

3.3.3 Labelling the Minority Class Examples

To automatically identify different types of examples, we propose a simple procedure which assesses the type of example by analysing the class labels of the examples in its local neighbourhood.



(a) Thyroid – "safe" dataset (b) Ecoli – "borderline" dataset

Figure 3.2: T-SNE visualisation of selected imbalanced datasets

We think that analysing a "local" distribution of examples may be better suited for this task than "global" approaches, especially when the minority class is considered, as this class is often decomposed into smaller subconcepts with difficult, nonlinear borders between the classes. What is more, such "local" analysis has been used in many preprocessing methods dedicated for class imbalance (such as OSS, NCR, CNN, SMOTE or SPIDER).

Our aim is to distinguish whether a minority class example is a *safe*, *borderline*, *rare* or *outlying* example. Let us recall that by *safe* examples we understand the examples located in the homogenous regions populated only by the examples from the same class. Borderline examples are placed in the boundary regions between the classes where the examples from different classes overlap. They can also be the examples lying close to the complex borderline separating the classes, even when there is no overlapping. Minority examples located inside the majority class, far from the decision boundary, are considered to be *outliers* or *rare cases*. Outliers are more distant, single examples surrounded by many examples from majority classes, while rare cases are not single ones and may form isolated groups of few examples.

To identify these four types of examples, for each minority example we analyse the class assignment of its k-nearest neighbours. Other methods modelling the local neighbourhood could also be used (see e.g. [12]), hovewer we wanted to stay with a simple approach. In Section 3.4 we will consider also an alternative approach based on kernel functions.

The performance of the k-nearest neighbour approach depends on the distance function and the value of k. To calculate the distance between the examples we use the HVDM distance metric – see a discussion of its choice in Section 2.4. As for the value of k, we have based our choice on the suggestions given in [11, 43]. In [11], the authors compare the values of k from 1 to 27 for several distance functions, analysing the total accuracy on a set of UCI datasets (not considering class imbalance). They concluded that for the HVDM measure, values between 5 and 11 were the best (and the differences of performance between them were very small). The comparison in [43] concerned the use of k-NN in the class imbalance setting, and the suggestion was that for difficult data distributions, more local classifiers (with small k) are recommended.

Following these suggestions, we have decided to use $k = 5$. $K = 1$ and $k = 3$ may poorly distinguish the nature of examples, especially if we want to assign them to four types. Neighbourhood of size 5 seems enough for the purpose of this analysis, and it is often used in the related preprocessing methods for class imbalance. We will also evaluate experimentally in Section 3.4 that using $k = 7$ does not change the results too much.

With $k = 5$, the proportion of neighbours from the same class against neighbours from the opposite class can range from 5:0 (all neighbours are from the same class as the analysed example) to 0:5 (all neighbours belong to the opposite class). Depending on this proportion, we propose to assign the labels to the examples in the following way:

- 5:0 or 4:1 – an example is labelled as a safe example (further denoted as S).
- 3:2 or 2:3 – a borderline example (denoted as B). The examples with the proportion 3:2 are correctly classified by its neighbours, so they might still be safe. However, the number of neighbours from both classes is approximately the same, so we assume that this example could be located too close to the decision boundary between the classes.
- 1:4 – labelled as a rare example (denoted as R), only if its neighbour from the same class has the proportion of neighbours either 0:5 or 1:4 (additionally, in case of 1:4, it must point to the analysed example). Otherwise there are some other examples from the same class in the proximity (although not in the immediate surrounding of $k = 5$), which suggests that it could be rather a borderline example B.
- 0:5 – an example is labelled as an outlier and denoted as O.

A more probabilistic interpretation of this method will be given in Section 3.4.

3.3.4 Validation of the Labelling Method

The presented method is based on a simple analysis of a fixed number of neighbours. We are aware that it gives rather an approximation of a real distribution of examples, so we would like to verify its usefulness. To check whether the assigned labels can precisely reflect the known distribution of examples, we use the artificial datasets. Inspired by a good experience with such data in [102], we generated a number of such datasets. They contained 800 examples described by 2 numerical attributes. The minority class formed elliptical subconcepts, surrounded by uniformly distributed majority class examples. The datasets were characterized by various imbalance ratios (from 1:5 to 1:9) and a different number of the minority class sub-concepts (from 1 to 5). In these datasets we changed the percentage of safe, borderline, rare and outlying minority examples. Table 3.1 presents the description of several analysed datasets and the labelling results.

Table 3.1: Labelling of artificial datasets

Dataset Description					Identified Labels			
Imbalance Ratio	Sub-concepts	Border [%]	Rare [%]	Outlier [%]	Safe [%]	Border [%]	Rare [%]	Outlier [%]
1:5	1	60	20	0	17.04	60.74	21.48	0.74
1:5	3	60	20	0	18.52	57.78	23.70	0.00
1:5	5	60	20	0	17.78	64.44	17.78	0.00
1:5	5	0	0	10	64.44	25.93	0.00	9.63
1:7	5	0	0	10	54.00	36.00	0.00	10.00
1:9	5	0	0	10	52.00	36.00	2.00	10.00

The first three datasets are disturbed in the same way (60% of borderline examples and 20% of rare examples), but differ in the number of sub-concepts. One of them (with 5 sub-concepts) is plotted in Fig. 3.1a. Proportions of the identified labels show that our labelling method can correctly reconstruct the percentage of safe, borderline and rare examples, regardless of the number of sub-concepts. The other three datasets contain 10% of outliers and differ according to the

imbalance ratio. Here, the labels also correctly reflect the percentage of outliers regardless of the changing imbalance ratio. However, although the classes in these datasets are not overlapped, a considerable number of examples is labelled as borderline. Let us recall that by these examples we understand also the examples lying close to the borderline between the classes, even when there is no overlapping. The examples close to the border between the classes can contain in their neighbourhood some examples from the opposite class, so our labelling method will also assign them to the B category.

3.4 Analysing Real-world Datasets – Experimental Study

Experimental Setup

In this experiment we want to analyse the distribution of different types of minority examples in real-world datasets. We analyse 21 imbalanced datasets often used in the related works on class imbalance, which represent different sizes, imbalance ratios, domains and have both continuous and nominal attributes. Most of the datasets come from the UCI repository, while 4 datasets are retrospective medical datasets which were used in the earlier works on class imbalance[2]. The characteristics of the datasets is presented in Table 3.2.

Table 3.2: Characteristics of the datasets

Dataset	No of examples	Imbalance ratio [%]	No of attributes (numeric)	Minority class name
abdominal-pain	723	27.94	13 (0)	positive
acl	140	28.57	6 (4)	1
new-thyroid	215	16.28	5 (5)	hyper
vehicle	846	23.52	18 (18)	van
car	1728	3.99	6 (0)	good
scrotal-pain	201	29.35	13 (0)	positive
credit-g	1000	30	20 (7)	bad
ecoli	336	10.42	7 (7)	imU
hepatitis	155	20.65	19 (6)	die
ionosphere	351	35.89	34 (34)	bad
haberman	306	26.47	3 (3)	died
cmc	1473	22.61	9 (2)	l-term
breast-cancer	286	29.72	9 (0)	rec-events
cleveland	303	11.55	13 (6)	positive
glass	214	7.94	9 (9)	v-float
hsv	122	11.48	11 (9)	4.0
abalone	4177	8.02	8 (7)	0-4 16-29
postoperative	90	26.66	8 (0)	S
solar-flare	1066	4.03	12 (0)	F
transfusion	748	23.8	4 (4)	yes
yeast	1484	3.44	8 (8)	ME2

Labelling the Datasets

The results of labelling the minority class examples in all the datasets are presented in Table 3.3. To facilitate the analysis, we have sorted the datasets from the "easiest" to the "most difficult" (in terms of the complexity of the data distribution).

[2]We are grateful to prof. W. Michalowski and the MET Research Group from the University of Ottawa for abdominal-pain and scrotal-pain datasets; and to prof. K. Slowinski from Poznan Univerisity of Medical Science for hsv and acl datasets.

The first observation is that most of the datasets contain the examples of all four types. Moreover, a majority of datasets contains rather a small number of safe examples - only in the top four datasets (from abdominal-pain to vehicle) safe minority examples prevail and they have almost no rare or outlying examples. Some datasets, on the other hand, do not contain any safe examples – such as cleveland, glass or solar-flare.

Datasets from car to ionosphere consist of safe and borderline examples in comparable proportions and they do not have many rare or outlying examples. In these datasets there is probably a complicated border between the classes or some overlapping occurs.

Then, we can distinguish a group of datasets in which the borderline examples dominate in the distribution of the minority class - these are datsets from credit-g to haberman. A high number of borderline examples may suggest that there is a strong overlapping of classes on the border between the classes in these datasets.

Several datasets contain many rare examples. Although they are not that numerous as B or S examples, they can constitute even 20-30% of the minority class. Datasets from haberman to postoperative have at least 20% of rare examples. Other datasets contain less than 10% of these examples.

Finally, some datasets contain a relatively high number of outlier examples – datasets from cmc to yeast contain more than 20% of these examples. Sometimes the outlying examples constitute more than a half of the whole minority class (see cleveland, abalone, hsv). This observation confirms the discussion in Section 3.2, in which we claimed that lonely minority examples cannot be treated entirely as noise. If the standard noise-handling techniques, which relabel or remove such examples, were used in the datasets like hsv, it would seriously degrade the recognition of this class, as it would result in removing more than a half of all the minority examples. Finally, it is interesting to observe that for many datasets rare and outlying examples appear together.

Note that the results of this labelling method are consistent with the observations of the MDS visualisations. The three datasets visualised in Fig. 3.1b-3.1d also show that new-thyroid contains mostly safe examples, ecoli has a lot of borderline examples, while cleveland constitutes mostly of rare and outlying examples.

Influence of Parameters on the Labelling Results

Although some of the results of our proposed labelling method are confirmed by the MDS and t-SNE visualisations, we would like to verify in more detail if the results presented in Table 3.3 are not related to the used identification method rather than to the distribution of the analysed datasets.

First, we want to verify if the results depend strongly on the used size of the neighbourhood, by comparing the results of using $k = 5$ with the neighbourhood of size 7. To do that, new thresholds for assigning the examples to the four categories based on the distribution of neighbours have to be established.

Note that we can treat the proposed neighbour-based approach as a discrete estimator of the underlying (continuous) probability distribution in the small region containing the analysed example x, where the probability of class membership of x is estimated as

$$p(C_{min}|x) = \frac{K_{min}}{K}$$

C_{min} is a minority class, K is the number of neighbours and K_{min} is the number of minority class neighbours [12]. Looking at our identification method from this perspective, the proportion of neighbours 3:2 (in case of which we treat an example as borderline) is equivalent to the distribution estimation $p(C_{min}|x) = \frac{3}{5} = 0.6$, while proportion 4:1 (safe example) is equivalent to the distribution estimation 0.8. Interpolating between these values, we can say that our method labels

Table 3.3: Labelling of datasets

Dataset	S [%]	B [%]	R [%]	O [%]
abdominal-pain	59.90	22.28	8.90	7.92
acl	67.50	30.00	0.00	2.50
new-thyroid	68.57	31.43	0.00	0.00
vehicle	74.37	24.62	0.00	1.01
car	47.83	39.13	8.70	4.35
scrotal-pain	38.98	45.76	10.17	5.08
ionosphere	44.44	30.95	11.90	12.70
credit-g	9.33	63.67	10.33	16.67
ecoli	28.57	54.29	2.86	14.29
hepatitis	15.63	62.50	6.25	15.63
haberman	4.94	61.73	18.52	14.81
breast-cancer	24.71	25.88	32.94	16.47
cmc	17.72	44.44	18.32	19.52
cleveland	0.00	31.43	17.14	51.43
glass	0.00	35.29	35.29	29.41
hsv	0.00	0.00	28.57	71.43
abalone	8.36	20.60	20.60	50.45
postoperative	0.00	41.67	29.17	29.17
solar-flare	0.00	48.84	11.63	39.53
transfusion	18.54	47.19	11.24	23.03
yeast	5.88	47.06	7.84	39.22

the example as safe if its distribution is greater than 0.7. Calculating the remaining thresholds between the categories analogously, our identification method with $k = 5$ can be translated to:

Definition 3.1
if $1 \geq p(C_{min}|x) > 0.7$ *then label x as safe;*
if $0.7 \geq p(C_{min}|x) > 0.3$ *then label x as borderline;*
if $0.3 \geq p(C_{min}|x) > 0.1$ *then label x as rare;*
if $0.1 \geq p(C_{min}|x) > 0$ *then label x as outlier;*

To use the identification method with $k = 7$ while preserving the above probability thresholds, the labels are assigned in the following way:

- 7:0 or 6:1 or 5:2 – a safe example
- 4:3 or 3:4 – a borderline example; again, the examples with proportion 4:3 are correctly classified by their neighbours, but the number of neighbours from both classes are approximately the same, so they might lie in the borderline region
- 2:5 or 1:6 – a rare example; analogously to the original method we verify if, in case of 1:6, the example and its neighbour are the only minority examples in the proximity, while for 2:5 proportion we verify if an example with its *two* neighbours form an isolated group with no other minority examples in their surrounding
- 0:7 – an outlier

We have analysed again the same 21 chosen datasets with $k = 7$ and compared the results with the original method ($k = 5$). For most of the datasets and types of examples, the differences in percentage of examples assigned to a given type were small (up to 5-10%). Higher differences were observed only for glass and solar-flare datasets (they resulted from changing between rare

and borderline categories) and for car and hepatitis datasets (change between safe and borderline categories) – the detailed results can be found in Appendix A (Table 1). Despite these small differences, we can assume that the analysis presented in Table 3.3 was not the result of the selected k, and that $k = 5$ is sufficient to analyse the datasets.

Then, we wanted to verify if analysing the local neighbourhood based on fixed k (either to 5 or 7) does not influence negatively the results, as the datasets might have different densities in different regions. Again, looking on the problem from the probability estimation perspective, we can estimate the local distribution either by fixing the number of neighbours and determining the area for which the estimation is calculated from the data (as in k-nearest-neighbour methods) or by fixing the area and estimating the number of neighbours from the data, giving rise to kernel approaches [12]. Therefore, we compare the results of our method with using a kernel approach.

In this method, a kernel function is used to determine which neighbours should be taken into account in the class probability estimation and what weights should be assigned to them, based on their distance from the analysed example. We apply a commonly used kernel function, Epanechnikov (presented on Fig. 3.3), which gives more weight to the neighbours closer to the analysed example. The width of the function (which determines the maximum distance up to which the examples are treated as neighbours), has been set for each dataset separately, and it is equal to the average distance to the 5^{th} neighbour of each minority example in the dataset, to keep the average number of analysed neighbours comparable to the one used in our original method. The type of a minority example is assigned according to the thresholds given in Def. 3.1. We have observed that some examples do not have any neighbours closer than *width*; in this case we assume that we do not have enough information about these examples and do not take them into account in our analysis. In practice, such examples constituted up to 5-10% of the dataset. Only in the ionosphere dataset, there were 40% of such examples, so the results for this dataset should be treated with caution. The labelling of examples based on the kernel approach is presented in Table 3.4.

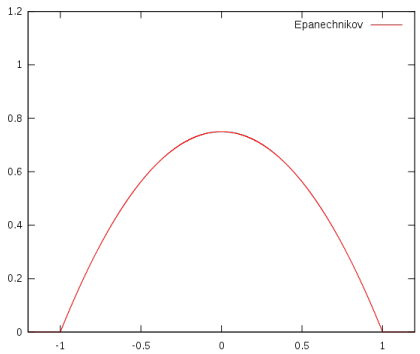


Figure 3.3: Epanechnikov kernel function

Comparing the results in Tables 3.3 and 3.4, we can observe that using the kernel method does not change the results more than by 10% for most of the datasets. Only in three datasets the differences are more visible. In postoperative dataset, 24% of examples changed its label from borderline to outlier. However, it should be remembered that it is a very small dataset, and this difference refers in fact to only 5 minority examples. In breast-cancer dataset, also more examples are labeled as outliers and less as borderline in the kernel approach. Finally, there are bigger differences for the ionosphere dataset (there are shifts between safe and borderline examples and between rare and outlier examples). However, let us recall that in this dataset 40% of examples remained unlabeled by the kernel method which might have influenced the results.

Table 3.4: Labelling of datasets – the kernel density method

Dataset	S [%]	B [%]	R [%]	O [%]
abdominal-pain	62.0	21.9	5.3	10.7
acl	72.2	22.2	0.0	5.6
new-thyroid	62.5	37.5	0.0	0.0
vehicle	77.4	18.9	0.0	3.7
car	47.8	43.5	8.7	0.0
scrotal-pain	24.4	53.3	11.1	11.1
ionosphere	12.9	62.9	1.4	22.9
credit-g	13.9	63.3	6.4	16.3
ecoli	25.8	61.3	3.2	9.7
hepatitis	13.6	63.6	9.1	13.6
haberman	15.1	56.2	16.4	12.3
breast-cancer	18.8	46.3	33.8	1.3
cmc	17.2	44.3	10.4	28.2
cleveland	6.7	30.0	13.3	50.0
glass	6.7	40.0	26.7	26.7
hsv	0.0	0.0	16.7	83.3
abalone	7.8	23.7	11.4	57.1
postoperative	0.0	65.2	30.4	4.3
solar-flare	7.1	45.2	7.1	40.5
transfusion	15.1	57.8	9.6	17.5
yeast	15.2	37.0	2.2	45.7

We have also tested other kernel functions, such as Gaussian, triangular or uniform functions; we have also tested other kernel widths (calculated as the average distances to the 3^{rd}, 7^{th} and 9^{th} neighbour), but it did not influence too much the results. Therefore, we can assume that the revealed distributions are rather inherent characteristics of the datasets than of the particular neighbourhood used. We can also say that the simple method based on analysing the local neighbourhood of fixed size 5 is sufficient to analyse the distribution of a dataset. We will base on its results in the following Sections.

3.5 Influence of Types of Examples on Learning of Classifiers – Experimental Study

Having shown that the analysed imbalanced datasets differ in their distribution of minority examples, we would like to verify whether they constitute a different degree of difficulty for the learning algorithms, and whether different classifiers reveal different sensitivity to the particular types of examples. We could expect that the datasets with a lot of safe minority examples will be easier to learn than the datasets with borderline examples and that rare or outlying examples will be particularly difficult for most of the classifiers.

We want to focus on the basic classifiers rather than on the complex ones such as classifiers ensembles. We have decided to compare five learning algorithms which have been often considered in related works and which represent different learning strategies. They are: tree learning by C4.5, rule induction with PART algorithm, k-nearest neighbour (kNN), neural network based on radial functions (RBF) and support vector machine SVM[3]. C4.5 and PART are run without pruning as pruning may be harmful for imbalanced datasets [108]. kNN is used with $k = 1$ and $k = 3$, as it has been suggested e.g. in [43] that for difficult imbalanced datasets more local classifiers (with

[3]The WEKA implementations are used. We use J48 version of C4.5 and SMO version of SVM.

Table 3.5: Sensitivity [%] of compared classifiers

Dataset	1NN	3NN	J48	PART	RBF	SVM
abdominal-pain	76.4	78.5	69.8	72.6	75.0	71.8
acl	72.0	78.5	85.5	80.0	84.0	82.5
new-thyroid	96.3	90.2	92.2	93.3	99.5	89.8
vehicle	89.1	87.9	87.0	88.3	88.0	95.2
car	3.1	3.1	77.7	90.0	49.6	88.2
scrotal-pain	58.4	58.7	55.3	63.4	62.5	65.9
ionosphere	69.4	65.5	82.7	84.0	94.2	89.0
credit-g	50.3	39.9	46.5	47.7	43.6	52.2
ecoli	52.2	50.8	58.0	42.0	54.7	58.5
hepatitis	44.0	37.0	43.2	45.7	60.7	51.5
haberman	30.1	26.9	41.0	33.4	18.3	1.3
breast-cancer	40.4	27.6	38.7	41.1	40.8	45.3
cmc	37.6	33.8	39.2	37.7	12.1	5.2
cleveland	20.3	12.5	23.7	25.2	9.5	9.0
glass	30.0	16.0	30.0	34.0	25.0	0.0
hsv	0.0	0.0	0.0	2.0	1.0	0.0
abalone	20.5	16.5	30.4	18.8	12.3	0.2
postoperative	4.3	0.0	4.7	10.3	13.7	7.0
solar-flare	9.1	8.2	20.9	18.7	10.2	15.7
transfusion	31.9	34.3	41.3	42.9	32.9	2.2
yeast	38.1	26.2	30.9	26.7	15.1	0.0

smaller k) perform better on the minority class. Standard values of parameters for RBF and SVM have failed to recognize the minority class. For RBF we have scanned several configurations trying to get the best sensitivity measure on all 21 datasets. As a result, we changed a number of clusters to 5 and minimum standard deviation to 0.1. The similar optimization has been done for the SVM classifier. It is used with the RBF kernel, complexity $C = 50$ and gamma parameter 1.0. We will come back to the issue of SVM parametrisation later in this Section. The performance of the classifiers is evaluated with Sensitivity and two aggregation measures – G-mean and F-measure. Their values are estimated by means of a 10-fold stratified cross-validation repeated 5 times to reduce possible variance. We resign from using the ROC curves and AUC measure, as most of the selected classifiers give determinstic predictions and these measures are more suited for the probabilstic classifiers - see discussion in Section 2.2. In datasets with more than one majority class, they are aggregated into one class to have only binary problems.

First, we compare the performance of classifiers on the 21 imbalanced datasets. Table 3.5 presents the Sensitivity measure. Let us comment these results, relating them to the labelling of the datasets presented in Table 3.3. The datasets in Table 3.5 are sorted in the same way as in Table 3.3 – from the simple to the more complex distributions. We can observe how with the increasing difficulty of the dataset distribution, the performance of all the classifiers decrease. For datasets where safe examples prevail (abdominal-pain -new-thyroid), all classifiers learn the minority class quite well – they recognize 70-90% of the minority examples. In datasets with more borderline examples (car-haberman), the classifiers usually recognize 40-60% of the minority class. When many rare and/or outlying example are observed (datasets haberman-yeast), the Sensitivity ranges between 0% and 40%. Finally, for the datasets with a lot of outlying examples (e.g. cleveland, hsv, abalone), it is impossible to recognize more than 30% of the minority examples (for some data even no examples can be classified). As for a majority class, all the classifiers can recognize this class in a similar degree, reaching 80–100% on Specificity for all the datasets (see Appendix A, Table 4 for details).

The second observation is that different classifiers reveal different sensitivity to the particular types of example. To examine the importance of differences between the classifiers on a collection of datasets, we apply the non-parametric ranked Friedman test ([66, 58]) which globally compares the performance of several methods on multiple datasets with a null hypothesis saying that all methods perform equally. It uses ranks of all classifiers on each of the datasets – the higher rank, the better classifier. We also carry out a post-hoc analysis (a Nemenyi test [58]) of differences between the average ranks of classifiers. In both tests we use a confidence level $\alpha = 0.05$.

First, we compare how the classifiers perform on average on all datasets, not taking into account their nature. The null hypothesis, saying that all the classifiers are equal, is rejected with $p \leq 0.007$. The ranking of classifiers according to their average ranks is presented in Fig 3.4a. The best classifiers are PART and J48, while SVM and 3NN perform the worst. The critical difference (CD) according to the Nemenyi test is 1.65 – so we cannot say that differences between the best performing classifiers are significant, however the first classifiers are better than the last ones.

Classifier	Avg. rank
PART	5.2
J48	4.5
RBF	4.2
1NN	4.0
SVM	3.8
3NN	2.7

(a) All datasets

Classifier	Avg. rank
SVM	5.5
PART	4.7
RBF	4.5
J48	4.1
1NN	3.3
3NN	2.2

(b) Safe and border datasets

Classifier	Avg. rank
PART	5.9
J48	5.5
1NN	4.2
RBF	3.6
3NN	2.8
SVM	2.5

(c) Rare and outlying datasets

Figure 3.4: Rankings of classifiers depending on the nature of the dataset (based on Sensitivity).

In our opinion, averaging over the datasets of different nature might hide the interesting characteristics of the learning methods. Our aim is rather to analyse the influence of types of examples on the performance of classifiers. To carry out such analysis, we have decided to divide the collection of datasets into two groups. In the first group we place the datasets where the are a lot of safe and borderline examples, and only a small number of outlier or rare examples – these are datasets from abdominal-pain to haberman. In the second group we put the datasets where many rare and/or outlying examples were observed - these are datasets from haberman to yeast[4]. If we consider only the datasets from the first group, the ranking of classifiers is definitely different – SVM is becoming the best classifier (see ranking in Fig. 3.4b). While considering the second group of datasets, we observe an opposite behaviour (Fig. 3.4c). PART and J48 perform well (the critical difference CD is now 2.27, so they dominate some of the remaining classifiers). 1NN also becomes one of the best classifiers and it becomes much better than 3NN. RBF or SVM fail at these datasets. It is also reflected by the results of the Wilcoxon test, which we use as a supplementary test to additionally analyse the differences between the selected pairs of classifiers. The results of this test showed that, e.g., PART or J48 are better than RBF with $p \leq 0.025$. If we look solely on the most difficult datasets with a lot of outlying examples (e.g. cleveland, hsv, abalone, glass, postoperative, yeast) in Table 3.5, J48, PART or 1NN can classify a few examples while SVM usually cannot recognize the minority class at all.

The results on G-mean and F-measure show quite similar trends (the corresponding tables can be found in Appendix A – Tables 2 and 3). On safe and borderline datasets, F-measure favors SVM and RBF over other classifiers even more than Sensitivity: SVM (5.5) $\succ$ RBF (5.2) $\succ$ PART (3.7)

[4]Let us note that we initially wanted to divide the datasets into four groups, one for each type of minority examples. However, the number of datasets in each group would be too small to be able to draw meaningful conclusions. E.g. there would be only four datasets representing the "safe" group.

$\succ$ J48 (3,6) $\succ$ 3NN (3.4) $\succ$ 1NN (3.2). The differences on G-mean are more subtle. When datasets with rare and outlying examples are considered, for both measures J48 and PART dominate other classifiers. For G-mean the ranks are almost identical as for Sensitivity. F-measure results are also similar, except that RBF swaps with 1NN.

The results of RBF and SVM classifiers need an additional comment. It is important to remember that these two classifiers are known to be sensitive to the configuration of parameters. As we did not want to favor any of the classifiers, in our experimental setup they were used with one configuration for all the datasets (similarly to other compared classifiers), which was on average the best on the whole collection of datasets. A similar approach was taken, e.g., in the experimental setup in [132]. In total, more than one hundred combinations of parameters were tested, so there was already a lot of effort put in the tuning of these classifiers. However, tuning the parameters for each group of datasets separately (e.g. selecting another configuration for datasets with mostly safe and borderline examples and another for those with a lot of rare and outlying examples) or even for each dataset separately, might improve the performance of these classifiers, especially on the rare and outlying examples. We have done a preliminary study in this topic. When we performed tuning for a whole group of rare and outlying datasets, we could not identify a single configuration which would improve all the datasets. When we optimized the parameters for each dataset separately, for some datasets with rare and outlying examples the results of SVM and RBF could be improved. However, there were only a few configurations among a hundred of combinations tested which yielded an improvement. Therefore, although the results of some rare and outlying datasets presented in Table 3.5 could be higher if a more fine-grained tuning was performed, the general conclusion remains the same: the SVM and RBF classifiers are very sensitive to the rare and outlying minority examples.

Classifiers' performance with respect to the types of testing examples

Note that examples of different types often occur together. So, one can ask a question which types of examples actually contribute to the "global sensitivity" and which types are the most difficult. For instance, yeast dataset was included in the second group of datasets as it contains a lot of outlying examples and almost no safe examples, but still many of its examples are of type B (for details see Table 3.3). As a result, "global sensitivity" in yeast may rely more on the recognition of B than of O examples. As our labelling method enables to identify a type of each testing example, we record the "local accuracy" for each type of testing examples separately. In Table 3.6 we present the results for two classifiers – PART and 1NN – which represent different learning strategies and achieve a high "global sensitivity", especially for more difficult datasets. What is more, PART is a rule classifier on which this thesis is focused. When analysing these results, one should keep in mind that the minority class is small, and partitioning the testing examples with respect to their types makes the representation of each type even more sparse. Thus, we do not present the results for too small datasets (less than 300 examples) to ensure that there are enough representatives in each category. Also, as some datasets do not have any examples of a particular type or their number is too small (e.g. cleveland, vehicle) – in Table 3.6 we left the corresponding cells empty.

Note that these results confirm the previous observations. Safe examples are easy to recognize for both classifiers (except for car dataset where 1NN cannot recognize the minority class at all). Borderline examples are more difficult, but still a large number of them can be correctly learnt. Rare examples are usually recognized within the range of 10–30%. Outlier examples are extremely difficult for these and other classifiers. Only for cmc, credit-g and cleveland datasets some of them are partly recognized, while for other datasets these examples are mostly neglected.

Table 3.6: Local accuracies for labeled testing examples [%]

	1NN				PART			
Dataset	S	B	R	O	S	B	R	O
abalone	81.4	45.3	22.5	2.1	74.3	27.8	12.4	10.4
abdominal-pain	99.5	74.5	5.7	0.0	91.7	69.5	17.1	8.8
car	5.5	1.5	0.0	0.0	91.5	91.1	100	46.7
cleveland		35.0	8.9	20.0		45.0	22.2	16.7
cmc	81.0	35.9	26.3	18.2	63.1	37.5	34.9	19.1
credit-g	72.9	53.9	42.1	39.2	65.7	53.3	40.5	32.0
ecoli	100.0	49.4		0.0	84.0	32.9		12.0
haberman	100.0	43.5	18.1	0.0	90.0	48.2	20.6	5.0
ionosphere	96.4	69.7	49.5	0.0	98.6	92.1	67.6	37.5
solar-flare		16.3	14.0	0.0		27.5	32.0	2.4
transfusion	80.6	41.9	12.4	0.0	86.1	64.5	21.2	1.6
vehicle	97.6	71.8			93.1	74.1		
yeast	100.0	62.2	52.0	0.0	73.3	50.0	20.0	2.0

3.6 Addressing Types of Examples by Preprocessing Methods – Experimental Study

The preprocessing methods, which balance the distribution of examples between the classes, can improve learning. These methods are based on different principles – e.g they clean the majority class or oversample the minority class. In this experiment we want to compare the performance of the preprocessing methods, depending on the type of minority examples in the dataset. We consider four different methods: simple Random Oversampling, SMOTE – a representative of informed oversampling, informed undersampling method NCR [71] and a hybrid approach SPIDER [102] – see their definition in Section 2.3.1. SMOTE is used with $k = 5$ and the oversampling ratio aimed to balance the distribution between the classes (the configuration suggested e.g. in [132] and many other studies). SPIDER also uses $k = 5$ to preserve consistency with SMOTE. All the methods use the HVDM distance measure.

Analogously to the experiments in the previous section, we compare the preprocessing methods using Sensitivity, F-measure and G-mean measures. We conduct this analysis for only four best classifiers – PART, one kNN classifier (1NN as it is better than 3NN especially for rare and outlier examples), RBF and J48. We resigned from the SVM classifier as it was the worst on rare and outlier datasets and it was difficult to parametrize for all datasets. As in this thesis we are interested mostly in rule-based classifiers, we present the detailed results for one classifier, PART. We chose it also because on average it performed the best on all the datasets. The Sensitivity on each dataset is presented in Table 3.7[5]. For other analysed classifiers, we comment their results as well, especially when they differ from the PART's results.

First, we compare the preprocessing methods on all 21 datasets, not taking into account their underlying distribution. The null hypothesis in the Friedman test is rejected. The order of preprocessing methods is given in Table 3.5a. The critical difference CD is 1.33. When all the datasets are concerned, a cleaning method (NCR) is followed by a hybrid method (SPIDER), followed by oversampling SMOTE, however the differences are statistically insignificant. All informed methods perform significantly better than Random Oversampling, and all preprocessing methods are better than no preprocessing.

However, if we split the datasets into two groups as in the experiments described in the previous

[5]Random oversampling is denoted as RO, SPIDER – SP, SMOTE – SM, no preprocessing – None.

section, the order is different (and in both cases the null hypothesis is rejected). For safe and borderline datasets, the NCR cleaning method performs even better (see ranks in Fig. 3.5b). SPIDER, which also performs some cleaning, is second. The critical difference CD is equal to 2.15.

For datasets with rare and outlier categories, on the other hand, oversampling methods seem to be better suited. The ranking, presented in Fig. 3.5c, shows that the oversampling method SMOTE and hybrid method SPIDER are the best and almost equal to each other. When we look only on the datasets with a lot of outliers in Table 3.7 (e.g. hsv, abalone, yeast), the advantage of SMOTE is even more visible.

Considering F-measure and G-mean, the results are similar, with slightly better average ranks for NCR in case of datasets with rare and outlying examples over SMOTE, which might suggest that SMOTE's increased performance on the minority class comes at a too high cost of the majority class recognition. This is consistent with the literature studies, which were based on artificial datasets [81, 9] – see discussion in Section 2.3.1. Results for F-measure and G-mean for PART can be found in Appendix A (Tables 5 and 6).

Table 3.7: Sensitivity for PART and preprocessing [%]

Dataset	None	RO	NCR	SM	SP
abdominal-pain	72.6	76.0	86.1	73.8	85.0
acl	80.0	83.5	91.0	86.5	87.5
new-thyroid	93.3	90.2	86.3	94.0	91.0
vehicle	88.3	90.6	92.6	92.4	91.4
car	90.0	75.6	92.6	88.3	91.2
scrotal-pain	63.4	66.6	74.9	69.7	72.1
ionosphere	84.0	84.0	86.8	88.9	85.0
credit-g	47.7	47.5	69.7	53.3	60.6
ecoli	42.0	55.0	71.2	74.0	72.8
hepatitis	45.7	57.3	63.3	54.8	56.7
haberman	33.4	55.3	59.7	68.3	70.3
breast-cancer	41.1	43.7	67.9	44.3	55.9
cmc	37.7	48.5	59.8	49.7	55.9
cleveland	25.2	16.7	42.2	28.8	24.5
glass	34.0	33.0	56.0	46.0	43.0
hsv	2.0	9.0	7.0	15.0	10.0
abalone	18.8	38.2	31.1	52.8	50.2
postoperative	10.3	21.7	42.3	17.0	36.0
solar-flare	18.7	35.6	45.5	33.9	46.1
transfusion	42.9	59.1	50.3	72.4	70.3
yeast	26.7	33.3	30.3	47.9	37.2

Preprocessing	Avg. rank
NCR	4.0
SPIDER	3.8
SMOTE	3.6
RO	2.2
None	1.4

(a) All datasets

Preprocessing	Avg. rank
NCR	4.2
SPIDER	3.7
SMOTE	3.4
RO	2.0
None	1.5

(b) Safe and border datasets

Preprocessing	Avg. rank
SPIDER	3.9
SMOTE	3.8
NCR	3.6
RO	2.2
None	1.2

(c) Rare and outlying datasets

Figure 3.5: Rankings of preprocessing methods used with PART, depending on the nature of the dataset (based on global sensitivity).

Table 3.8: PART and preprocessing: local accuracies recorded on borderline testing examples [%]

Dataset	None	RO	NCR	SM	SP
ionosphere	92.1	92.1	95.2	93.3	92.7
car	91.1	69.6	92.6	89.6	86.7
scrotal-pain	64.0	69.6	74.4	68.8	77.6
credit-g	53.3	54.1	76.9	58.8	67.9
ecoli	32.9	60.0	78.8	90.6	80.0
hepatitis	65.7	80.0	82.9	80.0	80.0
haberman	48.2	69.4	73.5	85.3	86.5

Rankings of preprocessing methods used with J48 are the same. For 1NN, SMOTE is often a better classifier. It becomes the first in the ranking for all datasets, second for safe and border datasets and first for rare and outlier datasets. Considering F-measure and G-mean, again NCR gets better ranks than SMOTE on rare and outlying datasets, suggesting that SMOTE can deteriorate too much the results of the majority class.

For RBF we observed different behaviour of Random Oversampling, which seems to work better than for the former classifiers. On Sensitivity, it is particularly good for the datasets with rare and outlying examples: SPIDER (3.9) = SMOTE (3.9) $\succ$ RO (3.5) $\succ$ NCR (2.6) $\succ$ None (1.0). Considering datasets with a lot of outliers, Random Oversampling becomes the best classifier. It is also better when G-mean is concerned – it wins in both groups of datasets. The results for RBF on Sensitivity can be found in Appendix A (Table 7).

Performance of preprocessing with respect to the types of testing examples

Again, to get more precise results, we analyse the local accuracies for each type of testing examples separately. Tables 3.8–3.10 present the values of accuracies for preprocessing integrated with the PART rule classifier. As already mentioned in Section 3.5, some datasets contain too few examples of a given type to provide reliable results. Therefore in Tables 3.8-3.10 we present only the datasets which have a sufficient number of examples of a given type.

When comparing the results of all methods to the use of PART without preprocessing (None column in Table 3.8), PART without preprocessing can recognize about 40-60% of the borderline examples (except for ionosphere and car on which PART performs better), 10-20% of rare examples (Table 3.9) and usually not more than 10% of outlying examples (Table 3.10). The preprocessing methods can increase the results on each type of minority examples by 10-30%.

If one considers borderline testing examples in the datasets given in Table 3.8, the Friedman test for all the classifiers rejects the null hypothesis and gives the ranking in which NCR method is on the first place. Ranks for PART are presented in Fig. 3.6a ($CD = 2.3$, so the difference is statistically significant only for RO and NCR). J48 and 1NN produce the similar order. For the RBF classifier and borderline testing examples, Random Oversampling is ordered before SMOTE.

The results for PART on rare testing examples are given in Table 3.9. Depending on the classifier, SMOTE or SPIDER seem to be the best choice in this case. The ranking for PART is presented in Fig. 3.6b – SMOTE is on the first position ex aequo with NCR. For 1NN the results are similar, while for J48 SPIDER is on the first position. RBF again behaves differently – SPIDER (4.2) $\succ$ SMOTE (3.5) = RO (3.5) $\succ$ NCR (2.7) $\succ$ None (1.1).

For outlier testing examples (in the datasets presented Table 3.10), SMOTE performs the best for all the classifiers. The ranking for PART is given in Fig. 3.6c ($CD = 2.15$). J48 and 1NN produce the same order with slightly different values. For RBF, Oversampling again performs quite well: SMOTE (4.2) $\succ$ RO (3.4) $\succ$ SPIDER (3.3) $\succ$ NCR (2.9) $\succ$ None (1).

Table 3.9: PART and preprocessing: local accuracies recorded on rare testing examples [%]

Dataset	None	RO	NCR	SM	SP
haberman	20.6	49.0	48.4	62.6	64.5
cmc	34.9	40.4	56.1	41.4	45.1
breast-cancer	26.7	28.7	59.3	35.3	44.7
cleveland	22.2	22.2	33.3	22.2	22.2
glass	25.0	25.0	45.0	37.5	35.0
hsv	0.0	30.0	0.0	20.0	20.0
abalone	12.4	37.1	26.5	52.1	48.8
postoperative	8.0	18.0	42.0	6.0	32.0
solar-flare	32.0	58.0	66.0	52.0	60.0
transfusion	21.2	42.4	31.2	62.4	58.8
yeast	20.0	42.0	12.0	38.0	24.0

Table 3.10: PART and preprocessing: local accuracies recorded on outlier testing examples [%]

Dataset	None	RO	NCR	SM	SP
cmc	19.1	24.0	28.0	25.5	30.2
breast-cancer	11.7	18.3	33.3	20.0	26.7
cleveland	16.7	11.1	37.8	21.1	10.0
glass	28.0	16.0	48.0	52.0	32.0
hsv	4.0	4.0	12.0	16.0	8.0
abalone	10.4	27.7	16.6	41.5	39.1
postoperative	5.7	5.7	28.6	22.9	14.3
solar-flare	2.4	16.5	12.9	12.9	27.1
transfusion	1.6	22.9	4.9	45.3	49.4
yeast	2.0	7.0	9.0	26.0	13.0

Preprocessing	Avg. rank
NCR	4.3
SPIDER	3.8
SMOTE	3.4
RO	2.0
None	1.3

(a) Border testing examples

Preprocessing	Avg. rank
SMOTE	3.7
NCR	3.6
SPIDER	2.9
RO	2.5
None	2.1

(b) Rare testing examples

Preprocessing	Avg. rank
SMOTE	4.3
SPIDER	3.5
NCR	3.4
RO	2.2
None	1.5

(c) Outlying testing examples

Figure 3.6: Rankings of preprocessing methods used with PART, depending on the nature of the dataset (based on accuracies on testing examples of a given type).

3.7 Conclusions

In this experimental study we were interested in the data properties of imbalanced datasets which influence the recognition of the minority class. We analysed four types of minority examples – safe, borderline, rare and outlier examples. We proposed the method for identification of these examples in the real-world, multidimensional data, based on the analysis of the local neighbourhood of learning examples [100]. Although it is quite simple, its results were validated on the artificial datasets. The results on the real-world datasets were confirmed (where it was possible) by the visualisation methods, based on the MDS and t-SNE projections of the multidimensional dataset into two dimensions. Changing the parameters of the proposed method such as the size of the neighbourhood, as well as using a kernel density approach, did not influence too much the results.

The use of the proposed method on 21 real-world datasets has led to the following conclusions:

- The datasets can be of different nature. Most datasets contain all types of examples, but in different proportions. Most of them do not contain many safe examples.

- Outlier examples can constitute an important part of the minority class – there are some datasets where they even prevail in the minority class. Therefore, one should be cautious with considering all of them as noise and applying noise-handling methods such as relabelling or removing these examples from the learning set. In general, distinguishing between noise and outliers in the minority class is an important, but challenging issue. We do not consider it in this thesis, however it would be an interesting topic for a future research.

- The imbalance ratio and the size of the data are not as influential as the above distribution types. Using the simple imbalance ratio to differentiate the data as in [132, 10] or the size of data [9] is not sufficient to explain the differences in the classification performance according to our experiments. For instance, datasets with a low imbalance ratio, e.g. car (3%) or ecoli (10%), are easier to learn than transfusion (23%). Similarly, large datasets are often more difficult than the small ones – compare, e.g., the results of abalone (over 4000 examples) with acl (less than 150 examples). Analysing the types of examples in class distribution provides information more relevant to the classification performance. Our observations are partly consistent with some earlier works with artificial datasets. In [44, 60, 102] it has also been shown that the imbalance ratio is not the main source of difficulty. However, these earlier works did not attempt to analyse real-world datasets.

- The four analysed data characteristics (safe, borderline, rare and outlier examples) differentiate the performance of classifiers. In general, safe datasets are easy to learn for all the classifiers. Datasets with a lot of borderline examples are more difficult, however the RBF and SVM classifiers work well on these datasets. Rare and especially outlier examples are extremely difficult to recognize. PART, J48 and sometimes 1NN may classify them but at a very low level. SVM and RBF are very sensitive to these data. Analysing the relationship between the distribution of the dataset and the optimal parameter setting of the SVM and RBF classifiers would be an interesting study, however it is outside the scope of this thesis as we are interested more in studying the data distribution and the rule-based classifiers.

- Finally, the competence of preprocessing methods has been analysed. In general, they can improve the recognition of the minority class examples by 10-30%. NCR (representative of informed undersampling) is better for safe and borderline examples, while SMOTE and SPIDER (informed oversampling and hybrid approach) are more accurate on rare examples. SMOTE is the best approach to improve the recognition of outliers. However, SMOTE can at the same time deteriorate too much the recognition of the majority class. All these informed sampling methods are significantly better than simple Random Oversampling for kNN, tree- and rule-based classifiers. For RBF, on the other hand, Oversampling often performs better.

- Our results often confirm the results of the related works conducted on artificial datasets. For instance, similarly to [43] we have observed that for datasets with more difficult distributions (i.e. with a lot of rare and outlier examples), a more local kNN (1NN) performs better compared to 3NN. Our results confirm also the hypothesis from [63], in which 1NN performed better that SVM on difficult distributions with a lot of outliers. Concerning the results of preprocessing methods, we have also observed that for difficult data distributions SMOTE can deteriorate the majority class too much and, perhaps, introduce overlapping of classes

– which was shown in [9]. Our observation that preprocessing methods usually improve the recognition of the minority class by no more that 30% is consistent with the results presented in [10]. Finally, the observation that informed resampling is better than simple resampling follows the conclusions from [9, 20]. Van Hulse et al in [132] gave contradictory recommendations in favour of simple random oversampling. However, in their experiment they do not take into account the distribution of the datasets, so we think that their conclusions may be a result of averaging over datasets having quite different characteristics.

- Our analysis of the common data distribution patterns carried out on a benchmark of imbalanced datasets can also contribute to the development of methods dedicated for class imbalance. First, the observation that safe examples are uncommon in most of the imbalanced datasets strengthens the need for taking into account the data distribution factors while developing new classifiers or new preprocessing methods dedicated for class imbalance. Then, our study shows that borderline examples appear in most of the datasets and often constitute more than a half of the minority class, so we might conclude that even though standard learning techniques can recognize them to some extent, concentrating on these examples in the methods dedicated for class imbalance may be profitable, as it should bring substantial improvements in the recognition of almost all imbalanced datasets. Finally, we could observe that rare and outlier examples are not only extremely difficult for most of the learning methods, but they are often numerous in the imbalanced datasets. Therefore, developing new methods which focus on these examples would be beneficial, especially taking into account the fact that the existing preprocessing methods do not bring enough improvement on these examples. These guidelines can help in the construction of the more effective informed preprocessing methods, in choosing the appropriate sampling techniques used in the ensembles of classifiers, or in the construction of the single learning algorithms. We will take these conclusions into account while proposing our new rule classifiers in the subsequent Chapters of this thesis.

Chapter 4

Learning Rule Classifiers from Imbalanced Data

In this chapter we discuss rule learning in the context of class imbalance. It contains a brief review of the standard rule learning algorithms, followed by a discussion of the limitations of the techniques used in these algorithms which may hinder learning from imbalanced datasets. We then review the existing modifications of the learning algorithms dedicated for class imbalance. We also take a closer look at classification strategies. We show that, to construct an efficient classifier for imbalanced data, a proper classification strategy is as important as the rule induction technique itself. Finally, we present several works which represent bottom-up rule learning and using single instances in a hybrid representation with rules. Although these approaches were not created for class imbalance, in our opinion they have several characteristics useful for dealing with such data. This chapter serves as an introduction to Chapter 5 in which we introduce our new rule learning algorithm, BRACID, dedicated for imbalanced datasets.

4.1 Basic Notations

The definition and notation of a learning example and a decision rule was given in Section 1.1. Let us now introduce some basic concepts of rule properties.

When the rules are induced for a class Y_j, the examples from this class are called *positive examples*. Examples belonging to the remaining classes are called *negative examples* of class Y_j.

A learning example is *covered* by a rule $P \mapsto Q$ if its description (values of condition attributes) satisfies the elementary conditions or a rule. A set of learning examples covered by a rule is called a *cover* $[P]$ of a rule. A rule is *certain* (also called *discriminant* or *consistent*) if it covers only the positive examples of class Y_j, i.e., $[P] \subseteq Y_j$. Rules which cover not only the positive examples, but also a limited number of negative examples are called *possible* (*partially discriminant* or *uncertain*) rules [36].

During the search for a set of rules, which describes the best the learning problem and will form the final classifier, candidate rules have to be created and evaluated to decide whether they should be included in the final rule set. Nearly all the rule learning techniques use the rule evaluation measures to guide the search, which estimate the generality and certainty of a rule [36].

Generality is usually measured in terms of *support* or *coverage*. The support of a rule R, denoted as $sup(R)$, is equal to the number of learning examples satisfying the condition and the decision parts of a rule (P and Q respectively), i.e.

$$sup(R) = |P \cap Q|$$

where $|.|$ is a cardinality of a set.

Coverage of a rule, called also *rule strength*, is defined as its support divided by a number of positive examples, i.e.

$$cov(R) = \frac{|P \cap Q|}{|Q|}$$

The certainty of a rule is usually measured in terms of $confidence$, which estimates the certainty of assignment to a decision class indicated by the rule. It is defined as

$$conf(R) = \frac{|P \cap Q|}{|P|}$$

This measure is also known as *certainty factor*, *accuracy* or *discrimination level*.

Note that when a rule covers a small number of examples, this measure is not very robust in that one additional example in $|P \cap Q|$ or in $|P|$ will change this measure significantly. Therefore, some authors introduce the corrections to this measure to estimate the confidence in a more robust way. Laplace accuracy [23] is defined as

$$Laplace(R) = \frac{|P \cap Q| + 1}{|P| + k}$$

where k is the number of classes. In this way, a certain rule in the two-class problem ($k = 2$) covering only one positive example will get an estimation of $2/3$, while certain rules covering more examples will asympotically go to one.

M-estimate [33] is another measure, which introduces a correction as

$$M\text{-}estimate(R) = \frac{|P \cap Q| + m * p_i}{|P| + m}$$

where p_i is the a-priori probability of class indications Q in the dataset. The m parameter allows to tune this measure with respect to the data and problem characteristics (usually more noise requires larger m).

Rules which are not certain, i.e. they cover the examples from different classes, can be treated as *deterministic* and output the most probable class. We can also use such rule as a probabilistic one to output a probability distribution over all classes rather than making a single categorical prediction [36].

Another group of measures is used to assess the gain achieved when refining rule R into R' (obtained by adding an elementary condition to P). The *accuracy gain* is defned as the difference in rule accuracy between two rules $R = P \mapsto Q$ and $R' = P' \mapsto Q$.

$$Accuracy\_gain(R', R) = conf(R') - conf(R)$$

Information gain is a logarithmic modification of accuracy gain, i.e.

$$Information\_gain(R', R) = log_2\ conf(R') - log_2\ conf(R)$$

Entropy gain is defined as

$$Entropy\_gain(R', R) = Ent(R') - Ent(R)$$

where $Ent(R)$ measures how well the rule R discriminates the classes:

$$Ent(R) = \sum_{i=1}^{k} -\frac{|P \cap Q_i|}{|P|} log_2\ \frac{|P \cap Q_i|}{|P|}$$

where k is the number of classes and $|P \cap Q_i|$ is the number of examples from class Q_i covered by conditions in P.

Let us note that the two properties of a rule (generality and certainty) are complementary – the most certain rules are usually more specific, while very general rules usually cover also some negative examples. Thus, it is important to control the trade-off between these two measures in the induction process. For instance, when two rules are equally certain, the more general rule should be favoured. The above measures do not take it into account. Therefore, weighted variants are often used, which trade off certainty and generality. The examples are *weighted accuracy gain* (WAG) and *weighted information gain* (WIG) measures [36].

$$Weighted\_accuracy\_gain(R', R) = \frac{sup(R')}{sup(R)} \times (conf(R') - conf(R))$$

$$Weighted\_information\_gain(R', R) = \frac{sup(R')}{sup(R)} \times (log_2\ conf(R') - log_2\ conf(R))$$

There are also other, more complex measures – their review can be found, e.g., in [5].

4.2 Standard Approaches to Rule Learning

Many algorithms have been proposed to induce rules from examples in a standard classification perspective without class imbalance. We describe here the general techniques used in these solutions and present in more detail some algorithms which will be used in the experimental evaluation as comparative approaches. For a more comprehensive review of the current state of the art in rule induction see, e.g., [38, 36, 127].

First algorithms come from Michalski's proposal of the AQ-family [88]. They are based on a *sequential covering* search technique (also called *conquer-and-divide* or *separate-and-conquer*) to find a minimal set of rules which covers the dataset. The covering algorithm generates sequentially the rules for each class independently. The pseudo-code of the sequential covering algorithm for a given class is presented in Algorithm 4.1. It starts with an empty set of rules and it repeatedly generates new rules with a `FindBestRule` procedure, until a stopping criterion is met, e.g. all positive examples of a given class are covered. Once a rule is added to the set of rules, all positive examples covered by this rule are deleted from the current set of considered examples [36].

Algorithm 4.1 *Sequential covering search procedure*

```
Procedure LearnOneClass (Learning examples E,  Class Y)

 1. Let RULE_LIST be an empty list.
 2. Let E_P be a set of positive examples in E.

 3. while E_P is not empty do
 4.    Let BEST_RULE be FindBestRule(E,Y).
 5.    Add BEST_RULE to RULE_LIST.
 6.    Remove from E_P examples covered by BEST_RULE.
 7. end while

 8. Return RULE_LIST.
```

In AQ, the rule generation in `FindBestRule` is *example-driven*, i.e. each rule is constructed from a *seed* learning example. It is called a *specific-to-general* search. A limited set of rules characterizing the seed is found and the "best" rule is added to the final ruleset. Popular successors of AQ are PRISM, LEM2 [45], MODLEM [118], ELEM2 [4] or CN2 [23].

CN2 is (similarly to AQ) a sequential covering algorithm, however, the rules in `FindBestRule` are found in the *general-to-specific* (top-down) direction. A construction of a rule begins with the most general (empty) rule, which is repeatedly specialised with new conditions as long as it still covers negative examples (or until other stopping criterion is met). It is also called a *generate-and-test* approach and is said to be more robust to noise than the *example-driven* approach. Rules in CN2 are specialized using the *beam search*, i.e. rather than considering a single candidate at each search step, it keeps track of the k best candidate rules. As a rule evaluation measure, it uses the Laplace correction.

MODLEM algorithm also follows a sequential covering schema and generates a minimal set of rules using a *generate-and-test* approach. Its specific property concerns direct processing of numerical values of attributes (without pre-discretization), missing values and imperfect descriptions of examples. It uses an Entropy measure to evaluate candidate rules. MODLEM will be presented in more detail in Section 7.3.

While classifying the new, unseen examples, rule sets are either ordered and the first matched rule indicates a decision, or the rules are undordered and a new example is tried against each rule. In this case, special strategies for solving conflicts between matched rules (or when no rule matched the example) have to be applied. We will discuss the classification strategies in more detail in Section 4.3. CN2 algorithm was first designed to induce the ordered set of rules and was then modified to produce also the unordered sets of rules. MODLEM algorithm produces unordered sets of rules.

Some other rule learning algorithms are based on the decision trees, such as C4.5rules [109] or PART [37]. They both produce ordered sets of rules and are based on the elements from the C4.5 decision tree learner. In C4.5rules, an unprunned decision tree is first built using Quinlan's C4.5 algorithm. Then, each path from the root to the leaf in the decision tree is turned into a single rule. Finally, pruning is performed to remove redundant conditions in the rules. Repeated and redundant rules are also removed if it improves a quality evaluation measure.

PART is a separate-and-conquer rule learner proposed by Eibe and Witten. In each iteration, a partial C4.5 decision tree is built and the "best" leaf is turned into a rule. This algorithm also performs an intensive post-pruning.

Other algorithms which intensively prune the ruleset are IREP [39], Grow [24] and RIPPER [25]. RIPPER (Repeated Incremental Pruning to Produce Error Reduction), an optimized version of IREP, is a sequential covering, general-to-specific rule learner. In this approach, a learning set is divided into *growing* set and *pruning* set (in the proportion 2:1). The classes are ordered by size, from the smallest to the largest, and the rules are built starting from the smallest class. RIPPER produces an ordered set of rules, and for the dataset with n classes, it induces rules only for $n-1$ classes. For the remaining (largest) class, a default rule is added at the end of the rule list. The algorithm builds the rules using the growing set, and immediately after a rule is built it tries to simplify it, evaluating it on an independent pruning set (so called *grow-and-prune* approach). After a ruleset is constructed, an additional optimization postpass is performed on the ruleset to further reduce its size and improve its fit to the training data. A combination of cross-validation and minimum-description length techniques is used to prevent overfitting.

Finally, note that rules can also be induced by other approaches, not based on the sequential covering technique. Using this technique often leads to so-called minimal set of rules, which

may contain a limited number of rules having interesting evaluation measures. In the so-called descriptive perspective of knowledge discovery [119], the researchers look for more rules satisfying certain requirements. Some of these algorithms, as BRUTE or EXPLORE, will be discussed in Section 4.5. For a brief review of other approaches, see [126].

4.3 Classification Strategies

Prediction of a class label for an unseen example is based on matching its attribute description to the condition parts of induced rules. When an algorithm produces an ordered set of rules, the first rule that completely matches the example indicates a class label. If no rule matches the example, a special default rule is used, which usually indicates the majority class.

When an unordered set of rules is used, the new example's description is tried against each rule in the set. In this case, conflict situations may occur when the description of a new example matches many rules from different classes or when it does not match any rule. Therefore, specialized *classification strategies* are necessary to solve the conflict situations and decide which class should be assigned to the example. Let us now briefly review the most popular classification strategies proposed in the literature for the algorithms producing unordered sets of rules.

Grzymala's LERS strategy
This strategy was originally introduced by Grzymala in [46]. It is based on a *voting* of matched rules according to their supports $sup(R)$. The total *support* for a class Y is defined as:

$$sup(Y) = \sum_{i}^{m} sup(R_i)$$

where R_i is a matched rule that indicates Y and m is the number of these rules. A new example is classified to the class with the highest total support. In case of no-matching, so called *partial matching* is considered where at least one of rule conditions is satisfied by the corresponding attributes in the new example's description x. The matching factor *match(R,x)* is introduced as a ratio of conditions matched by the object x to all conditions in the rule R. The total support is modified to

$$sup(Y) = \sum_{i}^{p} match(R, x) \times sup(R_i) \tag{4.1}$$

where p is the number of partially-matched rules.

Nearest rules strategy
This strategy, introduced by Stefanowski in [118], is also based on the idea of voting with rule supports. However, a rule support is calculated in a different way – if the rule covers examples from different classes, then their numbers are included in the total supports for each class Y. Then, the main difference lies in solving a no-matching case by means of so-called *nearest rules* instead of partially matched ones [118]. These are rules nearest to the object description with respect to the *valued heterogeneous metric* HVDM (see Section 2.4 for details). A coefficient expressing rule similarity (complement of the calculated distance) is used instead of matching factor in the Equation 4.1 and again the strongest decision class Y wins.

Default rule strategy
In this strategy a default rule is used in case of no-matching. It usually assigns the example to the largest class. In case of multiple matching, it uses the same solution as Grzymala's strategy. This strategy is used e.g. in a version of CN2 producing the unordered set of rules [23].

Single best rule strategy
Then, some strategies solve conflict situations by choosing a single rule according to its quality measure, e.g. *m-estimate* or Laplace accuracy. In case of *m-estimate*, the multiple matching is solved by selecting one rule in the conflict set with the highest value of the *m-estimate*. In case of no-matching, a default rule indicating the majority class is usually applied. In [101], it was modified to use partial matching in case of no-matching. More precisely, for each rule the *m-estimate* is calculated for the rule with the condition part reduced to matched conditions only. Then, a classification decision is taken according to one partially matched rule with the highest *m-estimate*.

Discrimination measure strategy
The four above strategies are the most popular in the rule-based classifiers. There have been also some alternative strategies proposed, which use a different rule quality measure. For instance, Aijun An proposed in [3] to use a rule quality measure called *measure of discrimination* for the ELEM2 rule learner. It is defined as

$$DM(R) = log\frac{P(R|Y) \times (1 - P(R|\neg Y)}{P(R|\neg Y) \times (1 - P(R|Y))}$$

where P denotes probability, R refers to a rule and Y to the decision class. For more technical details of estimating probabilities and adjusting this formula to prevent zero division, see [3]. Its interpretation says that it measures the extent to which rule R discriminates between positive and negative examples of class Y. Inside the classification strategies it is used in similar formulas for decision scores as in the Grzymala's strategy – the only difference concerns putting $DM(R)$ in place of $sup(R)$.

4.4 Limitations of Standard Approaches

Most of the rule induction algorithms described above share a number of problems when it comes to learning from imbalanced datasets. The most comprehensive and systematic study was presented in [139]. Although it concerns data mining in general, many of the observations are also true for rule approaches.

Top-down induction technique
As it was discussed in Section 4.2, rules are most often induced in a top-down (general-to-specific) manner. The top-down technique should favor general rules and avoid overfitting. This is often referred to as maximum-generality bias – when a learner decides to create a rule that covers a subset of training examples, it selects the most general set of conditions that covers those examples but no other. As a result, a top-down induction technique works well for so-called large disjuncts but it has difficulties with identifying the small disjuncts [55]. Rare examples, which are typical for the minority class, may depend on the conjunction of many conditions, therefore strategies which examine the conditions one-by-one in isolation may not guide the search in the proper direction [139]. This is especially true for the minority examples, which often form small disjuncts and may be overwhelmed by the surrounding majority examples.

Improper evaluation measures used to guide the search
A choice of the best condition which should be added to a rule in a given iteration depends on the evaluation measure, which typically tries to assess the accuracy and generality of the rule (e.g. Entropy measure, Laplace accuracy, support and confidence measures – see Section 4.1). Due to rarity of the minority examples, their impact on the accuracy and generality of a rule is much smaller than for common (majority) examples [139]. As a result, the search for the best condition

will be guided mostly by the majority class examples, neglecting the minority class (see, e.g., experiments in [22, 76, 6]).

Greedy, sequential covering technique

Nearly all popular rule induction algorithms employ a sequential covering search technique, in which the examples covered by the rules are removed from the current set of considered examples. Removing these examples during training partitions the space of examples into smaller and smaller groups and changes the descriptive statistics for the training set. As a result, rules generated in further iterations heavily depend on the previous rules and the examples they cover. Moreover, due to a too small number of examples used to induce the last rules, these rules may not be statistically significant. Data fragmentation is problematic especially for the minority examples, which are already sparse and have an intrinsic difficulty in being covered by statistically meaningful rules.

Biased classification strategies

As described in Section 4.3, in learning algorithms inducing unordered sets of rules, classification strategies are needed to solve the conflicts between the rules, when either no rules match a classified example, or many rules representing different classes match to it. They usually measure either the generality or accuracy of the rule and are based on voting of rules with weights depending on their evaluation measures. As minority class rules are usually more specific and supported by fewer examples, they can be characterised by worse values of evaluation measures than rules for the majority class. It can cause a classification bias toward the majority class (see e.g. empirical studies conducted in [141, 48]).

For instance, when we analyse the classification strategies presented in Section 4.3, we can observe that especially in case of no-matching, some of these strategies seem to be biased too much towards the majority classes. A default rule strategy, which in case of no-matching assigns an example to the majority class may be prone to give incorrect decisions by assigning the examples to the majority classes too often. M-estimate is also not well suited for such data, as it uses a class probability in the denominator (see Section 4.1 for its definition). As a result, minority rules (for which this probability is low) with the same accuracy as majority rules are characterized by worse values of this evaluation measure. Laplace accuracy may also discriminate the minority class, as the correction lowers the value of this measure significantly for rules with low support, while for rules with a high support the role of the correction is negligible. As minority rules usually have smaller support than majority rules, they will be affected more by the correction and receive lower evaluations. Grzymala strategy uses the rule supports of *all* partially matched rules, which may also favour the majority classes. Using a more local classification strategy, such as nearest rules, may be better for the minority class examples. Finally, as discrimination measure uses probabilities which are independent of class cardinalities, it may be less biased toward the majority classes.

In [101], these strategies have been compared using the MODLEM algorithm. The experimental results have shown that indeed a default rule and m-estimate were the worst strategies for the minority class, discrimination measure performed the best while nearest rules strategy was slightly better than Grzymala strategy, although the differences (apart from the discrimination measure) were not statistically significant. The discrimination measure was, however, too much biased towards the majority class, so it was also not a satisfying solution.

4.5 Review of Existing Modifications Dedicated for Class Imbalance

Several approaches have been proposed to deal with the above problems. They work on different stages of rule induction and usually aim to solve only one or few of these problems. We will now review them in groups, as shown on Figure 4.1.

Applying less greedy search technique

Using less greedy techniques of rule induction aims at finding more rules for the minority class, and/or improving their values of evaluation measures. This helps to find rules for small disjuncts and to increase the chance of classifying new examples as minority ones. The less greedy search is usually performed only for the minority class. The majority class is either not learnt at all (in case of one-class learning algorithms [148, 111]) or the rules for this class are learnt using a separate induction technique.

RLSD [148] is a one-class learning algorithm that learns only rules for the minority class. It initially generates one rule for each training example and gradually generalizes them. To reduce the number of obtained rules, it employs a sophisticated multi-phase approach based on precision, accuracy and F-measure. During classification, if a new example satisfies any rule, it is classified as a minority example, otherwise it is assigned to the majority class. RLSD was used to find patterns for fraudulent cases in law domain.

BRUTE [111], another one-class learning algorithm, also performs a more exhaustive search looking for accurate minority rules. The algorithm was succesfully applied in a Boeing manufacture design and was able to find small disjuncts of information that other algorithms were not able to locate.

An extention of the EXPLORE algorithm [126, 48] performs a less greedy search for the minority rules, looking for all rules that satisfy a certain threshold for rule support. Rules for the majority examples are induced with a standard sequential covering procedure. As a result, a set of rules for the minority class is more numerous and rules have on average better evaluation measures, which helps to outvote the majority rules during the classification of the unseen examples.

Changing the post-pruning phase

Other solutions try to improve the generality of the minority rules, concentrating on the post-pruning phase of rule induction. In IDL [103], a scheme of weighting the minority examples using a local neighbourhood is proposed. Weights are determined with the aim of maximizing the AUC measure. They are used as rule evaluation measures to decide if pruning should be performed. The idea is to prune only the rules with local neighourhood belonging to the same class.

Changing the classification strategy

The *strength* of minority rules (referring to the rule support measure) can be improved in yet different way, by modifying the classification strategy. For instance, Grzymala in [47] introduces a constant coefficient called a *strength multiplier* which is used to multiply the support of minority rules in a voting phase during classification in case of conflict situations. A value of a multiplier is optimized for a given dataset to maximize an aggregated measure of Sensitivity and Specificity.

Another point of view is taken in [13], where the classification strategy is modified in the ensemble of rule classifiers based on the Ivotes learning scheme. In this proposal, the component classifiers which are not sure about their prediction can abstain from voting for the class of a new example. In this way, only the classifiers which are competent for the new example (e.g. because they have learnt from the data sample representing the subspace of an attribute space close to the classified example), participate in the decision making. It was shown that using abstaining in the

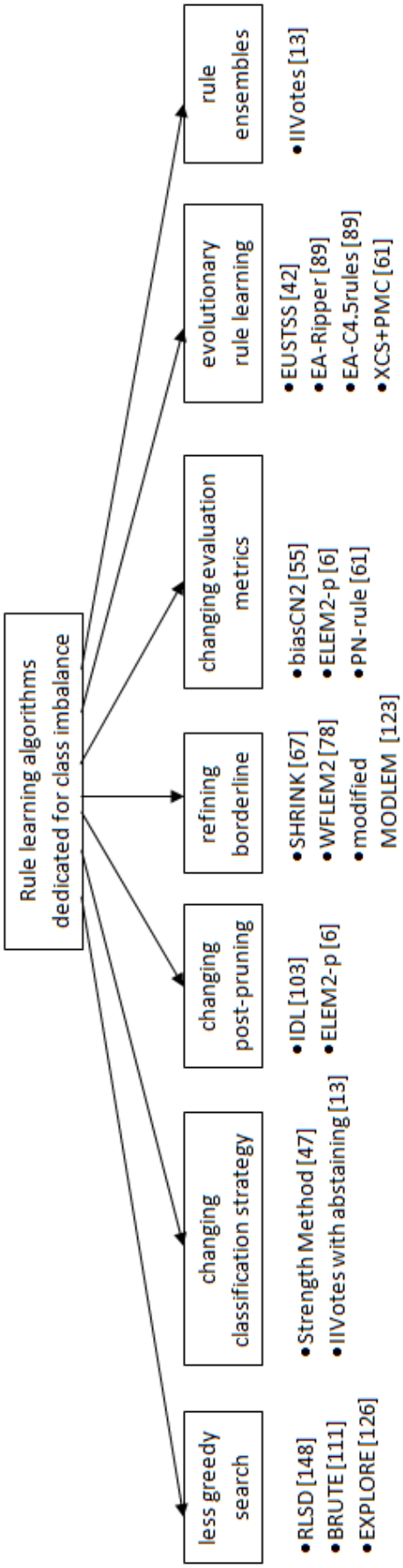


Figure 4.1: Classification of rule learning algorithms modifications

IIVotes makes the classifier more sensitive to the minority class. A similar approach also helped to improve the bagging classifier [101].

Using more appropriate evaluation measures

Some solutions try to deal with a learning bias of classifiers caused by the used evaluation measures, which favor the majority class and can fail to find the rules for small disjuncts (characteristic for minority classes). Holte, Acker and Porter in [55] change the bias of a CN2 algorithm. Original CN2 uses a maximum-generality bias when evaluating rules in the induction process, i.e. it selects the smallest subset of conditions to cover a particular set of training examples using the entropy measure. In the proposal of Holte *et al.*, the maximum-generality bias is used only for large disjuncts, while for small disjuncts a more specific bias is used. More precisely, when a rule is created for a particular (small, e.g. lower than 5 examples) set of training examples S and the maximally general subset of conditions G covering S was found, it is additionally extended by *all* other conditions that cover this subset of examples and meet additional requirements. The additional requirement verifies if the analysed condition does not cover too many examples from the other class.

In [6], the authors consider the ELEM2 sequential covering algorithm and modify its measure evaluating candidate rules. They analyse 11 different measures and show that the recognition of a minority class strongly depends on the particular measure. Furthermore, they focus on the post-pruning phase and propose to prune only the minority rules to obtain stronger rules, while leaving the majority rules unprunned.

Using more appropriate evaluation measures is used also in the PN-rule algorithm [61]. This approach was motivated by an observation that missing the rare cases is a result of optimizing precision and recall simultaneously. PN-rule consists of two phases. The first phase focuses on recall and finds strong rules, even if they are not highly accurate, called P-rules. In the second phase, precision is optimized by finding "exceptions" (rules covering false positives, called N-rules) for each P-rule from the first phase.

Refining rules for the borderline between the classes

This class of approaches concentrates on the borderline between the classes, where the examples from both classes overlap. Most algorithms assign this region to a majority class, because due to the sparseness of the minority examples, majority class usually prevails in the overlapping region. Some algorithms handle this region in a different way. SHRINK [67] finds rules only for the minority class, and labels the mixed regions as positive, no matter if the minority examples dominate in the region or not. There are also proposals to detect a boundary region between classes in a preprocessing phase and relabel all majority examples into minority ones in this region before learning [123]. Another algorithm proposed in [78], WFLEM2, is a rule learner based on rough sets and fuzzy theory. Briefly speaking, it creates weighted fuzzy aproximations of lower and upper bounds of the classes to balance the accuracy of the majority and minority classes in the overlapping regions.

Using ensembles of rule classifiers or genetic algorithms

Finally, to complete the review of the existing works on rules and class imbalance, let us mention a few proposals which combine rules with other paradigms, such as ensemble classifiers and evolutionary programming. There are some works where rules are used inside an ensemble of classifiers to deal with the imbalanced data. For example, in [13] the authors use an ensemble of rule classifiers, which is based on an Ivotes learning scheme [15]. Apart from using abstaining in classification (as described in the paragraph concerning modifications of the classification strategy), a selective preprocessing of examples using SPIDER method is performed to make the ensemble

more sensitive to the minority class.

In [42], an evolutionary algorithm EUSTSS is used to properly undersample the imbalanced training set to improve the performance of a tree- or rule-based classifier. The search space consists of all subsets of a training set. For a given subset, a C4.5 decision tree or a PART rule learner is used to induce a classifier. It is then evaluated by a G-mean measure, which serves as a fitness function.

In [89], a hybrid approach using a set of rule classifiers and an evolutionary algorithm is proposed. In this approach, several balanced datasets with all minority class cases and a random sample of majority class cases are fed to classic learning systems that produce rule sets. The rule sets are then combined to create a pool of rules and an evolutionary algorithm is used to build a final classifier from this pool. In this work, two such algorithms are proposed: EA-Ripper and EA-C4.5rules. Evolutionary algorithms are used also in [106]. Here, the authors propose a rule induction evolutionary algorithm XCS+PMC (XCS with mechanisms Protecting the Minority Class) which self-adapts depending on the imbalance level detected during learning. For instance, it adjusts a population size according to the imbalance ratio, to guarantee that the algorithm is initially supplied with enough rules and that the genetic search will pressure toward the recognition of the minority class.

4.6 Bottom-up Rule Induction and Hybrid Representations

In imbalanced data, rule learning algorithms suffer from the fragmentation problem and small disjuncts problem. Here we discuss two directions which could decrease the negative impact of these factors: using single instances in a hybrid representation with rules and inducing rules from single instances in a bottom-up way. Although, to the best of our knowledge, these paradigms were not yet used in the class imbalance setting, they share some characteristics which might be useful for such data. In this section we also describe the RISE algorithm, which applies both these approaches. Our algorithm, introduced in the next chapter, is insipired by its basic idea.

Instance-based learning (IBL) is a complementary induction paradigm to rule-based learning (RBL) and it is based on the classification according to the similarity of a new example to its local neighbours. In comparison to RBL, this ”lazy” learning paradigm can handle more complex, non-linear frontiers and it can work locally with fewer learning examples, making it less sensitive to class imbalance. However, opposite to rule learners, it is more sensitive to noise and irrelevant attributes. While rules usually represent a maximum-generality bias good for large disjuncts, IBL can be seen as a representative of a minimum-generality bias, suitable for small disuncts. There are some works which aim to combine both paradigms to create a general description in regions where the examples form large disjuncts (using a maximum-generality bias of rules) and in the regions of small disjunts, they exploit good properties of IBL (using its minimum-generality bias). Ting proposes such a hybrid approach in [129]. He first uses a decision-tree (C4.5) to determine if an example is covered by a small or large disjunct. If the example is covered by a large disjunct, then the tree is used to classify the example; otherwise an instance-based classifier is used.

We think that a hybrid use of both complementary paradigms is a good direction for learning with class imbalance. As it was pointed out in [44], some classifiers are less affected by overlapping, noise, small disjunct and imbalance, depending on their local or global nature. However, although the aim of Ting’s solution is to identify the small disjuncts without degrading the recognition of large disjunts, it can still suffer from the data fragmentation and improper bias, as it uses a top-down rule induction technique. We think that an opposite technique, called bottom-up (or specific-to-general), is more appropriate for learning rules from imbalanced data. Bottom-up

techniques start from the most specific rule that covers a single example and then generalise this rule until it cannot be further generalised without covering the negative examples (or until other stopping criterion is met). In this process, some examples may remain not generalized to rules and may be treated as maximally specific rules, leading to a transparent unification of RBL and IBL approaches. Bottom-up search seems better suited for situations where fewer examples are available [36], although it tends to build larger sets of rules and is more susceptible to noise.

Following these motivations for building hybrid rule and single instances representation by means of bottom-up rule induction, we identified in the literature the most related algorithm called RISE [32]. Although it has not been considered for class imbalance, we think that some of its solutions could be a good inspiration.

RISE algorithm

In RISE, a rule is represented as a conjuction of conditions. Conditions on symbolic attributes have a form of $attribute = value$ pairs, and conditions on numeric attributes are represented as closed intervals ($lower\_bound <= x <= upper\_bound$).

RISE starts from building an initial set of rules which is equal to the whole set of training examples. Each learning example is treated as a maximally specific rule (i.e. it contains conditions on all attributes and conditions on numeric attributes are degenerated, that is $lower\_bound = upper\_bound$). Unlike conventional rule induction algorithms, RISE does not construct one rule at a time, but induces all rules in parallel. Also, it does not evaluate each rule separately, but in the context of the rule classifier as a whole. In consequent iterations, rules are gradually generalized until no improvement in the overall accuracy of a rule set is obtained. Accuracy of a set of rules is calculated using a specific *leaving-one-out* procedure – see details in [32].

Generalization of a rule is done by generating the Most Specific Generalization (MSG) to the closest example of the same class, not already covered by this rule. MSG consists in dropping the nominal attributes in case they are different for the rule and example, and broadening the boundaries of intervals for conditions on numerical attributes to cover the nearest example. If during this generalization two rules become identical, one of them is dropped. An important feature of RISE is that when the closest example is selected for MSG, the choice is done from all the learning examples, even if they are already covered by a different rule. This prevents the data fragmentation problem caused by a sequential covering strategy.

Finally, a classification strategy consists in selecting the nearest rule. If several rules are in a conflict set (either because more rules cover the classified example and the distance equals 0, or because no rule covers the example, but several rules are equally distant) one rule is chosen based on the Laplace measure which estimates the confidence of a rule on a specifically chosen set of covered learning examples.

In [32], RISE was compared experimentally with IBL and three rule learning algorithms (PEBLS, CN2 and C4.5rules) on 30 datasets, using the total accuracy. According to the Wilcoxon test, RISE was significantly more accurate (with respect to the global accuracy) than all other algorithms.

Other hybrid algorithms

There are also other algorithms which induce the rules in a bottom-up manner. One of them is EACH algorithm [113] which generates hyperrectangles from examples. However, it can deal with numerical attributes only, and it generates a different representation than a set of unordered rules, as hyperrectangles can be nested inside each other, providing a hierarchy of rules and exceptions. INNER algorithm [79] is an attempt to deal with RISE's drawback of inducing too many rules – it randomly selects a subset of examples and generates rules "strategically placed in decision regions"

to treat them as the representatives of subconcepts in a class. As a result, it does not cover all the learning examples. Finally, FCLS [147] algorithm is a bottom-up modification of the AQ-family, which combines rules and examples to deal with the small disjuncts problem. Its drawback is that it uses the separate-and-conquer strategy of its AQ ancestors.

Chapter 5

BRACID: A Comprehensive Approach To Learning Rules From Imbalanced Data

5.1 Motivations

In Section 2.1 we have discussed the characteristics of data distribution in imbalanced data which hinder learning, such as overlapping of examples from both classes in the borderline region, small disjuncts or outlying/noisy examples. The experiments carried out in Chapter 3 showed that imbalanced datasets with complex data characteristics are difficult for most of the classifiers, including the rule-based ones. As pointed out in Section 4.4, there are several problems on the algorithmic level in rule learners which may be responsible for the degradation of perfomance when rule-based classifiers are learnt from imbalanced data. Let us recall that they include greedy search, sequential covering technique, improper evaluation measures or biased classification strategies. Although some modifications of rule-based algorithms have been proposed to deal with the above problems, the review of these approaches (presented in Section 4.5) shows that they address rather a single or at most a few of these problems. For example, some of the algorithms modify only the greedy search technique (e.g. RLSD [148]) or change the maximum-generality bias (e.g. a modification of CN2 algorithm [55]). The ELEM2 modification [6] works in two areas – evaluation measure used to guide the search and post-pruning phase. In our opinion, such "selective" approaches are not satisfactory and cannot handle sufficiently the difficulties of class imbalance. Therefore, in this Chapter we introduce a new rule induction algorithm, BRACID [99], which tries to deal with more of these problems – more precisely with *all* the main drawbacks of rule learners described in Section 4.4. It also makes use of the knowledge about the types of examples, which were discussed in Chapter 3.

We have decided to choose an integrated representation of rules and single instances (see motivations discussed in Section 4.6). Other crucial assumptions include:

- using a less greedy bottom-up induction of rules from single examples with the specific generalization by looking for the nearest examples to the rule,
- a new evaluation of a generated rule with respect to the recognition of imbalanced classes,
- proposing the new classification strategy with the nearest rule,
- a special treatment of the borderline and noisy examples.

These assumptions will be described in detail in the next sections.

5.2 Notation and Basic Concepts

The BRACID name is the acronym of **B**ottom-up induction of **R**ules **A**nd **C**ases for **I**mbalanced **D**ata. Before describing it, let us introduce some basic concepts.

Learning examples

A definition of a learning example has been given in Section 1.1. BRACID can handle attributes defined either on numeric or nominal scales. In a current form, it works with two-class problems of which one is a minority class (K_{min}) and the other one is the majority class (K_{maj}). For problems with more classes, all the examples from classes other than a selected minority class are merged into a single majority class.

Representation of rules

A rule R is represented similarly to the definition given in Section 1.1. In BRACID, there is at most one elementary condition per single attribute. For nominal attributes it is a single equality test of the form $(x_i = v_{ij})$ where x_i represents the i-th attribute characterizing an example and v_{ij} is a single value from its domain. Conditions for numeric attributes are represented as closed intervals $(v_{i,lower} \leq x_i \leq v_{i,upper})$, where $v_{i,lower} \leq v_{i,upper}$ are values belonging to the domain of the attribute.

In BRACID, examples can be treated as maximally specific rules containing conditions built on all attributes, where the intervals are degenerated to a single point $(v_{i,lower} = v_{i,upper})$.

Rule seed

As a rule in BRACID is induced in a bottom-up direction, by generalizing from a single example, we introduce the term *seed of rule R* to denote the learning example used for creating a maximally specific rule in the first iteration of BRACID.

Moreover, each learning example, in particular a seed of the rule, can be labelled by an extra *tag* expressing its type with respect to the characteristics of its local neighbourhood. Generally speaking, we distinguish between SAFE and UNSAFE examples. Safe examples are the ones which are correctly classified by their k-nearest neighbours. The misclassified examples are called unsafe examples. This distinction follows the typology introduced in [124]. Among the misclassified examples we distinguish *noisy* majority (and *outlier* minority) examples if all neighbours belong to the opposite classes; otherwise they are treated as *borderline*.

Distance measure

To determine the neighbours, either when performing a bottom-up generalization of a rule to the nearest example or when classifying new examples with the nearest rule strategy, we need to calculate the distance between the examples (or between the rule and the example). Following the discussion in Section 2.4, we have decided to use the *Heterogenous Value Difference Metrics* (HVDM). On nominal attributes, the distance between two examples and between the rule and the example is calculated according to the HVDM definition, given in Section 2.4. For numeric attributes, the calculation of the distance between rule x_i and example y_i had to be slightly modified, as a rule does not take a single value on the numeric attribute, but a pair $v_{i,lower}$, $v_{i,upper}$. Here, the attribute distance is defined as

$$d_i(x_i, y_i) = \begin{cases} 0 & if \quad v(x)_{i,lower} \leq y_i \leq v(x)_{i,upper} \\ \frac{y_i - v(x)_{i,upper}}{x_{max} - x_{min}} & if \quad y_i > v(x)_{i,upper} \\ \frac{v(x)_{i,lower} - y_i}{x_{max} - x_{min}} & if \quad y_i < v(x)_{i,lower} \end{cases}$$

5.3 Algorithm Description

A pseudo-code of the main procedure of BRACID is presented in Algorithm 5.1. While discussing the code we will refer to the critical factors mentioned in Section 4.4.

Using a bottom-up induction technique
We think that using a specific-to-general direction of rule induction can facilitate covering the subparts of the minority class which can be interpreted as small disjuncts. Furthermore, leaving some examples ungeneralized to rules can be profitable for rare examples and difficult (non-linear) decision boundaries.

Thus, we start from creating an initial set of the most specific rules `RS`, in which each rule corresponds to a single learning example (Algorithm 5.1, line 1). Then, in the main loop (lines 7-34) the algorithm considers each rule as a candidate for generalization in a bottom-up way. More precisely, in a given iteration the algorithm looks for the nearest examples (using the procedure `FindNeighbours`), which are not already covered by the rule and are from the same class. Depending on the class K of an example and its type (so-called `TAG` determined before the main loop in line 4), either one generalization to the nearest neighbour is considered, or k nearest examples are taken into account. This is done in procedures `AddOneBestRule` and `AddAllGoodRules`, which will be discussed further in the "Facing borderline examples" description. In these procedures, a generalized rule is temporarily added to a rule set RS and its influence on the F-measure is estimated (see "Evaluation metrics" description for details of the evaluation technique). If the generalization of this rule results in an improvement (or at least in no decrease) of the classification performance, the rule is stored in RS and the procedures return a flag `IMPROVED = TRUE`; otherwise the generalization is discarded and flag `IMPROVED = FALSE` is returned. If during the generalization process two rules become identical, one of them is dropped (line 31). The procedure is repeated until no rule in RS could be acceptably generalized (line 35). Let us note that generalizations which do not change the F-measure are also accepted, to promote more general models.

Algorithm 5.1 *BRACID - main procedure*

```
BRACID(Set of Examples ES)
 1 RS = ES              #initialize Rule Set RS with ES
 2 SEED = ES            #set seed examples for RS as ES
 3 FINAL_RULES = ∅      #rules not generalized in next iterations
 4 Calculate TAGS       #tag examples as SAFE or UNSAFE
 5 ITERATION = 0
 6 Flag IMPROVED        #TRUE means that generalization of rule R
                                  better than R was found
 7 NEIGHBOURS = ∅       #set for storing the nearest examples
 8 F = Evaluate(RS)     #Evaluate RS with leaving-one-out procedure
                                  using F-measure
 7 Repeat
 8   For each rule R ∈ RS and R ∉ FINAL_RULES         #main loop
 9      If K[R] = K_min           #a block for minority class rules
10      |  NEIGHBOURS = FindNeighbours(k,R)
        |                #find k nearest examples to R, such that:
        |                   R does not cover any NEIGBOURS[i] and
        |                   K[NEIGHBOURS[i]] = K[R]
11      | If TAGS[SEED[R]] = SAFE
```

```
12      |   IMPROVED = AddOneBestRule(NEIGHBOURS,R,RS, F)
13      | Else                              #seed tagged as UNSAFE
14      |   IMPROVED = AddAllGoodRules(NEIGHBOURS,R,RS, F)
        |
15      | If IMPROVED = FALSE
16      |   If ITERATION ≠ 0       #do not extend if "outlier"
17      |      Extend(R)
18      |      FINAL_RULES = FINAL_RULES ∪ R
19      |  Else                          #a block for majority class rules
20      |    If TAGS[SEED[R]] = SAFE
21      |         n = 1             #analyse only one neighbour
22      |    Else n = k             #analyse k neighbours
23      |    NEIGHBOURS = FindNeighbours(n,R)
24      |    IMPROVED = AddOneBestRule(NEIGHBOURS,R,RS,F)
25      |    If  IMPROVED = FALSE
26      |       If ITERATION = 0          #Treat as noise:
27      |           RS = RS \ R           #Remove rule R
28      |           ES = ES \ SEED[R]     #Remove seed of R
29      |       Else FINAL_RULES = FINAL_RULES ∪ R
31      | If IMPROVED = TRUE and R is identical to another rule in RS
32      |   Delete R from RS
33      | ITERATION ++
34 Until IMPROVED = FALSE for all R ∈ RS
35 Return RS
```

Generalization of a rule to the nearest example

Generalization of a rule is done using the `MostSpecificGeneralization` procedure (Algorithm 5.2). For nominal attributes, MSG consists in dropping the condition on the attribute in case the rule and example have different values on it (lines 4-5). For numerical attributes, the boundaries of intervals in a rule's condition ($R_{i,lower}$, $R_{i,upper}$) are minimally broadened to cover the example (lines 6-9).

Algorithm 5.2 *MostSpecificGeneralization procedure*

```
MostSpecificGeneralization(Example Neighbour, Rule R)

 1 For each Attribute X_i
 2    If condition on X_i is missing in R
 3       Do nothing
 4    Else if X_i is nominal and Neighbour_i ≠ R_i
 5       Remove condition on X_i from R
 6    Else if X_i is numeric and Neighbour_i ≥ R_{i,upper}
 7       R_{i,upper} = Neighbour_i
 8    Else if X_i is numeric and Neighbour_i ≤ R_{i,lower}
 9       R_{i,lower} = Neighbour_i
```

Less greedy search
When the nearest example is chosen for the MSG generation, it is selected from the whole learning set – examples covered by rules are neither removed, nor their weight is diminished in any way. As a result, the algorithm does not suffer from the data fragmentation in subsequent iterations, which could occur for the sequential covering.

Evaluation measure used to guide the search
To decide if the MSG generalization of a rule should be accepted, the influence of this generalization on the whole set of rules `RS` is estimated. Evaluating the rule set with global accuracy (as in RISE) is biased towards the majority class. In BRACID, on the other hand, we want to take class imbalance into account and focus more on the minority class. Thus, we choose the F-measure, which aggregates Recall and Precision measures. Both these measures are defined with respect to the positive (minority) class (see Section 2.2), which makes the classifier more "sensitive" to the minority class examples.

F-measure for a current rule set is estimated using a specific leaving-one-out procedure, originally proposed in RISE. Each learning example is classified by its nearest rule. Based on the classification predictions, a confusion matrix is calculated. When classifying a learning example, a rule for which this example is a seed is left out, unless it already covers other examples as well.

A calculation of a confusion matrix can be done efficiently – when a new MSG is evaluated, only this rule is matched against all examples, to check if it wins any that it did not before (i.e. it is closer to the example than a previously winning rule). If the decision for a newly won example has changed, the confusion matrix is updated.

Hybrid representation
Let us notice that the generalization of an example is accepted and included in the rule set only if it satisfies the leaving-one-out evaluation procedure. Otherwise the example remains ungeneralized as a maximally specific rule.

Facing borderline examples
To make BRACID more sensitive to the overlapping (boundary) regions, we use the information about the nature of examples to perform different actions in the consistent (*safe*) and in the overlapping (*unsafe*) regions. We assign tags (`SAFE` or `UNSAFE`) to all learning examples (line 4 of the Algorithm 5.1) based on the class labels of the nearest neighbours, as described in Section 5.2.

The BRACID algorithm processes rules differently depending on the tag and on the class of its seed example. For the `SAFE` examples from the majority class, we assume that the rule is created in the consistent majority region which is sufficiently represented in the learning set. Therefore, for these rules we analyse only the MSG to a single nearest neighbour (lines 20-21). For `UNSAFE` majority examples, we assume that this example could be inside the overlapping region, which should be more carefully analysed. So, we allow these rules to analyse the MSGs to k nearest neighbours, and to choose the best one according to the F-measure evaluation (lines 22-24), using `AddOneBestRule` procedure presented in Algorithm 5.3.

Algorithm 5.3 *AddOneBestRule procedure*

```
AddOneBestRule(Set NEIGHBOURS, Rule R, RuleSet RS, Evaluation F)

 1 BEST_F = F                    #F-measure evaluation of RS
 2 BEST_G = R                    #best generalization of R

 3 For each Neighbour in NEIGHBOURS
```

```
 4    G = MostSpecificGeneralization(Neigbour, R)
 5    TMP_RS = ( RS \ R ) ∪ G
 6    TMP_F = Evaluate(TMP_RS) #evaluate TMP_RS
                                by calculating influence of G
 7    If TMP_F ≥ BEST_F         on confusion matrix and F-measure
 8      BEST_F = TMP_F
 9      BEST_G = TMP_G

10 If BEST_G ≠ R               #better generalization was found
11    RS =  ( RS \ R ) ∪ BEST_G
11    R = BEST_G
12    F = BEST_F
13    Return TRUE

14 Else return FALSE
```

Minority examples are treated in a different way. For `SAFE` examples, we assume that the minority class is always underrepresented in the data, even in the consistent regions. Therefore, for these examples we also allow to analyse the MSGs for k nearest neighbours, and choose a single best generalization (lines 11-12). In case of `UNSAFE` examples, on the other hand, we assume that they should be additionally strengthened as they are located in the boundary region between the classes and could be overwhelmed by the majority class examples. Thus, we assume that an `UNSAFE` minority example can be generalized more than once. Having its k-nearest neighbours, we can add to the rule set *all* the generalizations, which do not harm the F-measure (procedure `AddAllGoodRules` in line 14). `AddAllGoodRules` is done in a greedy manner, by analysing the neighbours starting from the nearest one. The first MSG which does not harm the F-measure estimate replaces the original rule in RS, while the MSGs to the following neighbours (estimated with respect to the updated RS) are added to RS.

Facing noisy and outlying examples

Noisy majority examples, present inside the minority class regions, may hinder the induction of general minority rules. BRACID has an embedded mechanism for detecting and dealing with such examples. If a maximally specific rule representing a single majority example cannot succesfully generalize to any of its neighbours, we assume that it represents a noisy example. Otherwise, the learning set would possess at least one similar majority example, as we assume that this class is well represented in the dataset – see discussion in Chapter 3. Therefore, such maximally specific majority rules are removed from a set of rules. Additionally, the corresponding learning example (seed) is removed from a learning set, because it may disturb the evaluation of a confusion matrix for the nearby minority rules and prevent them from generalizing in this direction (lines 26-28 in Algorithm 5.1).

In case of the analogous situation for the minority class rules (i.e. when the maximally specific minority rule cannot be generalized to any of its neighbours), the rule and its corresponding seed are not removed, because we assume that such an outlying example may belong to a valid sub-concept of this class, which is just not sufficiently represented in the learning set (see discussion in Chapter 3). Our experiments will also show, that for some datasets such maximally specific minority rules prevail in the final ruleset, so their removal might seriously harm the performance. In the experimental setup we will analyse how many of these examples are labeled as outliers also by our method described in Chapter 3.

Facing the underrepresentation of the minority class

As minority class is often underrepresented in the data, its examples are also more sparsely disposed in the attribute space than the majority examples. As a result, the decision boundary is often shifted too close to the minority class. Thus, we have decided to extend the boundaries of minority rules. When there is no neighbour of the same class, towards which the rule can be successfully generalized, BRACID performs the `Extend` procedure on the rule and adds it to the `FINAL_RULES` set (lines 17-18 in Algorithm 5.1) .

The `Extend` procedure (Algorithm 5.4) processes only the conditions in R on numerical attributes (lines 3-4) and allows to extend the intervals towards the surrounding majority examples. This is done by choosing k nearest examples from the opposite (majority) class (line 1). For each attribute's left and right boundary separately, the closest (not covered – line 6 and 9) neighbour is selected (line 5 and 8), and the interval is extended to half of the distance between the rule boundary and the neighbour's value on this attribute (lines 7 and 10).

What is important, the `Extend` procedure is not performed on maximally specific rules representing single examples which could not be generalized to any of its neighbours (line 16 in Algorithm 5.1). We assume that such examples may be outliers and we do not want to amplify such regions.

Algorithm 5.4 *Extend procedure*

```
Extend(Rule R)

 1  OPPOSITE_NEIGHBOURS = FindNeighbours(k,R)
                    #find k nearest examples to R, such that:
                       R does not cover any NEIGBOURS[i] and
                       K[NEIGHBOURS[i]] ≠ K[R]
 2 For each Attribute X_i
 3    If X_i is nominal or condition on X_i is missing in R
 4       Do nothing

  #extend left boundary towards the nearest neighbour
 5    Find arg min_k(R_{i,lower} − OPPOSITE_NEIGHBOURS[k]_i)
 6           such that R_{i,lower} − OPPOSITE_NEIGHBOURS[k]_i > 0
 7    R_{i,lower} = 0.5 * (R_{i,lower} - OPPOSITE_NEIGHBOURS[k]_i)

  #extend right boundary towards the nearest neighbour
 8    Find arg min_k(OPPOSITE_NEIGHBOURS[k]_i − R_{i,upper})
 9           such that OPPOSITE_NEIGHBOURS[k]_i − R_{i,upper} > 0
10    R_{i,upper} = 0.5 * (OPPOSITE_NEIGHBOURS[k]_i − R_{i,upper})
```

5.4 Evaluation of Computational Costs

An important question determining the usability and applicability of BRACID, is whether this bottom-up, less-greedy induction of rules is much more costly than standard greedy sequential covering algorithms. As the general loop of BRACID is partly analogous to that of RISE, we can make use of its cost evaluation [32]. In the worst case, when in each iteration only a single rule is generalized on only one condition, the complexity was shown to be $O(e^3a^2)$ where e is the number of examples, and a is the number of attributes (see [32] for details). Most often, many rules will

be generalized in parallel and more than one condition will be processed in each iteration. For comparison, a complexity of CN2 algorithm is estimated as $O(be^2a^2)$, where b is the beam size.

From many elements which differ BRACID from RISE, the most costly is using k neighbours instead of a single rule/example. Since in one iteration BRACID allows to analyse k generalizations to a rule, and to produce k rules from one example, the above estimation should be multiplied by a constant value of k^2. However, this worst case is very unlikely, as it would happen only if all learning examples were the minority unsafe examples. Let us remark that in our experiments, BRACID's time was comparable to that of RISE.

5.5 Classification Strategy Based on the Nearest Rule

Using rules and single examples induced by BRACID to classify new coming examples is another non-trivial issue. As we discussed in Section 4.3, algorithms inducing unordered sets of rules require special classification strategies to solve conflict situations of ambiguous matching of the new example's description to multiple rules from different classes or non-matching to any rule. The behaviour of different strategies in face of imbalanced data was analysed in Section 4.4. It showed that classification strategy can have an important impact on the performance of the classifier, and that some strategies may be biased toward the majority class.

Yet another issue is that BRACID produces both rules and single examples. Having such a combined, dual knowledge representation, we have decided to make the classification decision on the basis of the *local neighbourhood* of the new example, as it is often done in the instance-based learning. In other words, we look for a rule or a single example which is the nearest to this example. This idea seems to be a natural extension of the k-NN principle and it is also consistent with BRACID's internal procedures for a rule generalization. Additionally, the nearest rule strategy may reduce the impact of the global domination of majority rules in the rule set. It also diminishes the role of very general rules, for which the quality measures are estimated basing on the example distributions in the regions distant from the classified example. So, we think that such a local strategy may be less biased towards the majority class. Let us also recall that the similar nearest rule strategy was also successfully used in the earlier works of Stefanowski [116, 117] as well as in the related RISE algorithm [31]. However, in these works the authors considered the overall classification accuracy only and did not take into account class imbalance.

To calculate the nearest rule to the classified example, we apply the same HVDM measure as in BRACID (see Section 5.2). However, even assuming that we look for the first nearest rule only, it may happen that more rules are equally distant from the classified example, causing ambiguity. Such a situation may occur either when several rules cover the example, i.e. their $distance = 0$, or when no rule covers it, but several rules are equally distant from the example with $distance > 0$. These are conflict situations if the rules represent different classes. Let us stress that in the preliminary experiments we observed that such a situation may hold for about 20% of cases.

Such conflict situations could be solved in several ways, by taking into account additional measures characterizing the equally distant rules. For instance, Domingos in RISE [31] proposed to choose the rule with the highest value of accuracy calculated with Laplace accuracy [104], estimated on the very specific choice of so-called winning examples. However, we have observed that it did not work properly with respect to measures suitable for class imbalance. We also checked that nearly all rules induced by RISE were approximately equally certain (with respect to the confidence measure). In BRACID the situation is analogous. So, confidence-based measures might not sufficiently discriminate the rules. Additionally, using the Laplace accuracy favors the majority rules (we have discussed in Section 4.4 and we will show it on a toy example). This is

why we come back to rule support measures. Although majority rules are globally characterized by the higher support, focusing on a local neighborhood of the classified example should reduce the risk of the domination of majority class rules over the minority class rules.

In our classification strategy the decision how to classify a new example e is made according to the sum of supports for all equally distant rules R. The total support for class K_i and example e is defined with the following expression:

$$sup(K_i, e) = \sum_{rules\ for\ K_i\ equally\ distant\ to\ e} sup(R).$$

Example e is assigned to the class K_j for which the total support is the largest.

Summing the supports for all the equally distant rules may additionally help the minority class as BRACID generalizes more rules for the unsafe examples of this class in the difficult, overlapping regions (see Section 5.3).

Let us show it on a toy example. Figure 5.1 presents a possible conflict of rules when a minority example (marked with ?) is classified. Minority class examples are marked with black circles. Let us assume that all 4 rules (rectangles) are equally distant from the classified example. Using accuracy as a rule quality measure would result in a random selection of one rule, as all 4 rules are 100% confident. Selection of the single, strongest rule would assign the example to the majority class. Laplace measure also favors stronger (usually majority) rules – it would give $\frac{10+1}{10+2} = 0.92$ estimation for the majority rule, and $\frac{5+1}{5+2} = 0.86$ for the best minority rule. Summing the supports of all equally distant rules can result in a correct classification of this example.

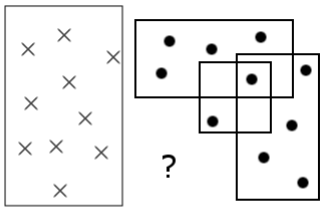


Figure 5.1: An example of a conflict situation while using the nearest rule classification strategy

Chapter 6

BRACID – Experimental Study

This Chapter presents the experimental study concerning BRACID. The aim of the experiments is to evaluate the classification abilities of the BRACID classifier in presence of class imbalance. First, we want to analyse an impact of BRACID's components on its final performance. Then, we compare it with other standard rule induction classifiers. Although we could expect some improvements, we want to see how much one can gain using BRACID instead of well known rule-based approaches. We will also verify if BRACID is better than its related "parent" approaches, i.e. k-NN and RISE. Finally, we will compare BRACID against some methods dedicated to deal with class imbalance – MODLEM with a modified classification strategy, and PART with two preprocessing methods. We compare the classifiers using the evaluation measures suitable for class imbalance. Additionally, we compare the structure of a rules produced by BRACID and other classifiers producing unordered sets of rules. To determine the area of competence of BRACID, in the separate experiment we analyse its performance on the testing examples labelled using the procedure introduced in Chapter 3. This will let us estimate how efficiently BRACID deals with safe, borderline, rare and outlying minority examples.

6.1 Experimental Setup

The experiments are carried out on 22 imbalanced datasets. 20 of them come from the UCI repository, while *abdominal-pain* and *scrotal-pain* datasets are real-world retrospective medical datasets from prof. W.Michalowski and the MET Research Group from the University of Ottawa [143, 86]. The datasets represent a wide range of domains, imbalance ratios (from 3% to 35%), sizes (from 100 to over 4000 examples) and attributes (purely nominal, purely numeric and mixed). These datasets were often used in other experimental studies with related methods for class imbalance and they appeared to be difficult for the learning classifiers (see e.g. [132]). For the datasets with multi-class domains, we selected the smallest class as a minority class, and aggregated the remaining classes into one majority class. Let us notice that for some of these datasets the class imbalance ratio is rather low (e.g. *pima* or *ionosphere*). However, according to our analysis with the labelling method proposed in Chapter 3, they are also characterized by other influential factors as overlapping of decision classes or presence of noisy or rare examples which is consistent with the assumptions behind our approach. Table 6.1 summarizes the main characteristics of the datasets.

The performance of all compared classifiers is evaluated by three measures (considered also in Chapter 3): Sensitivity of the minority class and two aggregating measures – G-mean and F-measure (see their definitions in Section 2.2). We choose G-mean as it has a good intuitive meaning and expresses a trade-off between Sensitivity and Specificity. Furthermore, we think that it is important to analyse an additional measure which is not directly optimized in BRACID. Let

Table 6.1: Basic characteristics of datasets

Dataset	No of examples	Minority class size	Imbalance ratio [%]	No of attributes (numeric)	Minority class name
abalone	4177	335	8.02	8 (7)	0-4 16-29
abdominal-pain	723	202	27.93	13 (0)	positive
balance-scale	625	49	7.84	4 (4)	B
breast-cancer	286	85	29.72	9 (0)	rec-events
breast-w	699	241	34.47	9 (9)	malignant
car	1728	69	3.99	6 (0)	good
cleveland	303	35	11.55	13 (6)	positive
cmc	1473	333	22.61	9 (2)	long-term
credit-g	1000	300	30.00	20 (7)	bad
ecoli	336	35	10.42	7 (7)	imU
flags	194	17	8.76	29 (2)	white
haberman	306	81	26.47	3 (3)	died
hepatitis	155	32	20.65	19 (6)	die
ionosphere	351	126	35.89	34 (34)	bad
new-thyroid	215	35	16.28	5 (5)	hyper
pima	768	268	34.89	8 (8)	diabetes
postoperative	90	24	26.66	8 (0)	S
scrotal-pain	201	59	29.35	13 (0)	positive
solar-flare	1066	43	4.03	12 (0)	F
transfusion	748	178	23.80	4 (4)	yes
vehicle	846	199	23.52	18 (18)	van
yeast	1484	51	3.44	8 (8)	ME2

us also recall that we resign from analysing the ROC curves and calculating AUC measure, as the chosen rule classifiers give deterministic predictions while the way of calculating AUC reflects better the performance of classifiers with probabilistic outputs – see a quite similar discussion in [137] and other arguments in Section 2.2.

All the experiments were run with a stratified 10-fold cross-validation repeated 5 times for a better reproducibility of results and to reduce a possible variance of estimating the average of the measures.

In the additional experiments, we compare BRACID and selected rule classifiers with respect to the structure of the induced set of rules and to the average values of rule evaluations measures such as support and average number of rules.

6.2 Studying the Role of BRACID's Components

First, we evaluate the influence of BRACID's components on its final classification abilities. More precisely, we study the impact of:

- new classification strategy described in Section 5.5 (called in this experiment component C),
- removal of noisy majority examples (component N),
- the use of the `Extend` operator (component E).

The final classifier is called in this experiment BRACID-N-E-C beacuse it uses all three components. The version which does not extend the minority rules is called BRACID-N-C etc. A version without the C component uses a classification strategy coming from the RISE algorithm – based on the Laplace accuracy instead of the support.

Fig. 6.1 shows how these three components influence the Sensitivity measure. We present only a representative subset of analysed datasets – the behaviour on the remaining datasets was comparable. A single group of 4 bars refers to one dataset. Analysing the bars in a group from the leftmost bar (referring to the most simplified algorithm) to the rightmost bar (referring to the final algorithm with all the components), one can notice that adding the components improves the Sensitivity. Using a classification strategy better suited for class imbalance results in the highest increase of Sensitivity. Removing the noise in the majority class and extending the minority rules brings further improvements.

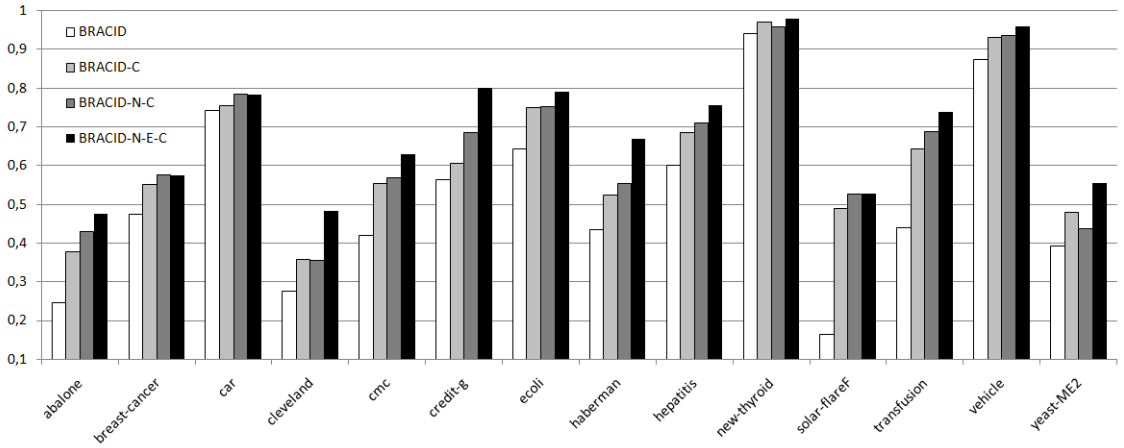


Figure 6.1: Influence of BRACID's components on Sensitivity measure

As all three components were created with a view to improve the recognition of a minority class, they may cause a decrease of a recognition of a majority class. Fig. 6.2 (presenting values of G-mean for all the datasets) shows however, that this aggregated measure also improves from the left to the right bars. It indicates that these components do not deteriorate the majority class too much. We also calculated the similar results for the F-measure and the conclusions were the same.

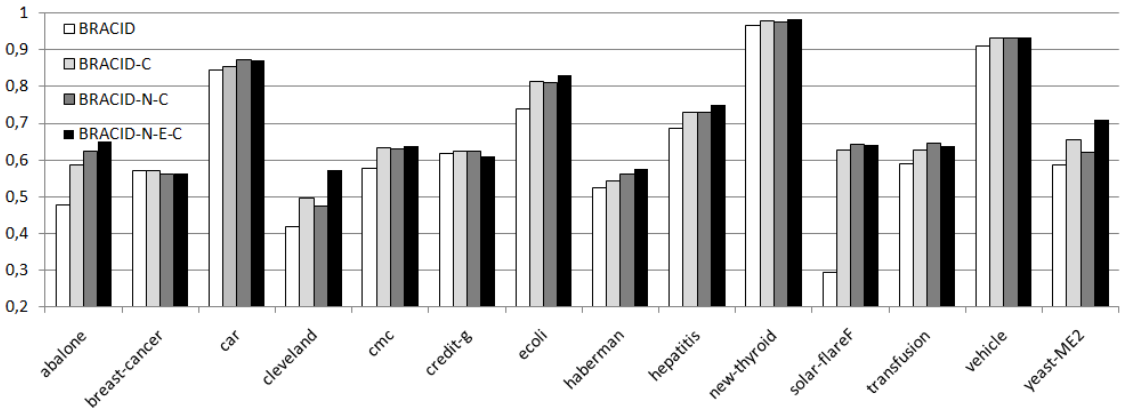


Figure 6.2: Influence of BRACID's components on G-mean measure

Finally, we analysed how these components affect the average rule support in the minority class. In Fig. 6.3 we present this measure for BRACID with N and E components and for the algorithm without these components. Component C operates only in the classification phase and it does not influence the induction of rules, so it is not included in this figure. It can be observed

that removing the noisy majority examples and using the `Extend` operator helps to create stronger rules for the minority class. It is worth mentioning here that BRACID-N-E increases also the average support of the majority rules, but it is rather a by-product of removing maximally specific majority rules (by the N component) which decrease the average value for the remaining rules.

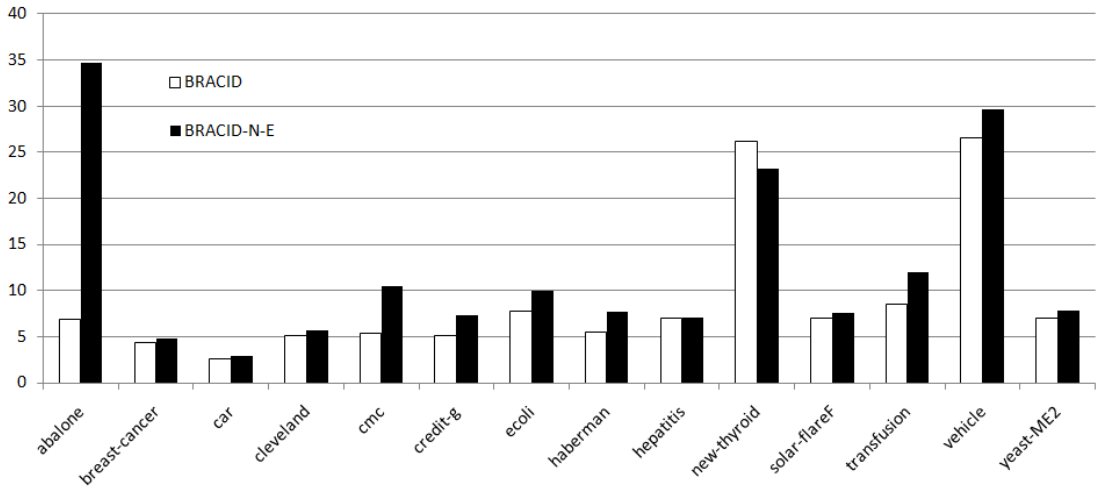


Figure 6.3: Influence of BRACID's components on the average rule support for the minority class

6.3 Comparison of BRACID with Standard Rule Classifiers

In this section we compare the classification performance of BRACID against 4 very popular rule algorithms: CN2 [23], PART [37], RIPPER [25] and C45rules [109] – they were presented in Section 4.2. We also compare it to the MODLEM [118] algorithm, as it was already used for imbalanced data (for example in [124, 125]) and its extention MODLEM-C will be used in the experiments presented in the next section.

CN2 is run with Laplace accuracy as an evaluation measure and beam size = 5. It produces the unordered set of rules with the classification strategy based on voting with rule supports in multiple matching and a default rule in case of non-matching. RIPPER is run with standard parameters (including rule pruning) and its typical classification strategy using an ordered list. PART, C45rules and MODLEM are also used with standard parameters, however without prunning. MODLEM is used with the standard Grzymala classification strategy [46]. BRACID is parameterized only with the neighbourhood size. We tested values 3, 5 and 7, which are often used in the neighbourhood-based methods (such as k-NN) and in the preprocessing approaches. Although all three values have led to comparable results, $k = 5$ has been slightly better than the other two. Therefore, we present the results for $k = 5$ only.

As BRACID produces a hybrid instance and rule representation, we have also decided to compare it against a typical k-NN algorithm representing instance-based learning (with k=5, to stay with the same value as in BRACID) and to the RISE algorithm which is the most related hybrid algorithm.

In case of CN2, RISE, MODLEM and C45rules algorithms, the original authors' implementations were used[1]. All other implementations come from the WEKA library. BRACID was

[1]CN2 is available at `http://www.cs.utexas.edu/users/pclark/software/` , RISE at `http://homes.cs.washington.edu/~pedrod/` and MODLEM at `http://www.cs.put.poznan.pl/jstefanowski/`; a code of C4.5rules was attached to a book [109].

Table 6.2: Sensitivity for BRACID compared against standard algorithms

Dataset	BRACID	RISE	kNN	C45rules	CN2	PART	RIPPER	MODLEM
abalone	**0.474**	0.128	0.137	0.339	0.160	0.188	0.184	0.245
abdominal-pain	**0.782**	0.711	0.775	0.695	0.658	0.726	0.602	0.657
balance-scale	**0.565**	0.000	0.004	0.018	0.018	0.000	0.000	0.000
b-cancer	**0.572**	0.356	0.261	0.330	0.276	0.411	0.288	0.319
breast-w	**0.989**	0.959	0.968	0.917	0.886	0.947	0.896	0.887
car	0.781	0.596	0.031	0.753	0.544	**0.900**	0.530	0.787
cleveland	**0.483**	0.147	0.042	0.175	0.000	0.252	0.163	0.085
cmc	**0.631**	0.293	0.308	0.404	0.096	0.377	0.071	0.256
credit-g	**0.801**	0.359	0.371	0.373	0.260	0.477	0.213	0.365
ecoli	**0.790**	0.505	0.578	0.597	0.185	0.420	0.445	0.400
flags	**0.840**	0.020	0.000	0.308	0.000	0.250	0.190	0.000
haberman	**0.669**	0.224	0.181	0.244	0.184	0.334	0.180	0.240
hepatitis	**0.757**	0.487	0.475	0.358	0.050	0.457	0.417	0.383
ionosphere	**0.976**	0.902	0.629	0.837	0.779	0.840	0.818	0.824
new-thyroid	**0.980**	0.928	0.867	0.850	0.866	0.933	0.855	0.812
pima	**0.875**	0.551	0.558	0.507	0.408	0.591	0.377	0.485
postoperative	**0.577**	0.147	0.000	0.000	0.017	0.103	0.037	0.033
scrotal-pain	**0.771**	0.544	0.492	0.569	0.432	0.634	0.521	0.547
solar-flare	**0.517**	0.066	0.000	0.148	0.000	0.187	0.010	0.070
transfusion	**0.738**	0.297	0.319	0.386	0.150	0.429	0.088	0.371
vehicle	**0.960**	0.831	0.865	0.867	0.329	0.883	0.874	0.859
yeast	**0.555**	0.245	0.194	0.323	0.000	0.267	0.259	0.189

implemented by us using the WEKA components.

Tables 6.2-6.4 present the average values of Sensitivity, G-mean and F-measure, respectively, for all compared classifiers. In all these tables, for each dataset, we marked with bold fonts the best result.

We use a statistical approach to compare the differences in performance between all classifiers. First, we apply a non-parametric Friedman test to globally compare the performance of 8 different classifiers on 22 datasets (following the recommendations given in [66, 58]). The null hypothesis in this test is that all compared classifiers perform equally well. It uses ranks of all classifiers' results on each of the data sets. The higher rank, the better classifier.

Let us start from analyzing the results for the Sensitivity measure. Friedman statistics for these results gives 89.64 which exceeds the critical value for confidence level 0.05 (equal to 14.07) and we can easily reject (for p much smaller than $\alpha = 0.05$) the null hypothesis saying that all compared classifiers perform equally well. The average ranks of each of the classifiers are the following: BRACID 7.9; PART 6.11; C45rules 5.25; RISE 4.43; KNN 3.59; MODLEM 3.52; RIPPER 3.02; CN2 2.15. Then, we carried out a complete post-hoc analysis of differences between classifiers with a Nemenyi test [58]. The critical value of difference (CD) between the average ranks of two classifiers is 2.23. So, we can claim that Sensitivity of BRACID is significantly better to all other classifiers except PART – where the difference is smaller than CD. Then, we repeat the same testing procedure for G-mean. The Friedman statistics is 76.23 and we can again reject the null hypothesis. The average ranks of the classifiers are the following: BRACID 7.77; PART 5.7; C45rules 5.02; RISE 4.61; KNN 3.81; MODLEM 3.7; RIPPER 3.15; CN2 2.2. A post-hoc analysis leads to similar conclusions – performance of BRACID is significantly better than other classifiers and the difference between it and PART is just near CD. Statistical Friedman test for the F-measure has led us to the same conclusions.

As BRACID is always close to PART, we have decided to use the Wilcoxon signed rank test to

Table 6.3: G-mean for BRACID and standard algorithms

Dataset	BRACID	RISE	kNN	C45rules	CN2	PART	RIPPER	MODLEM
abalone	**0.650**	0.345	0.358	0.568	0.396	0.419	0.421	0.484
abdominal-pain	0.811	0.805	**0.828**	0.784	0.775	0.786	0.748	0.771
balance-scale	**0.567**	0.000	0.009	0.019	0.019	0.000	0.000	0.000
breast-cancer	**0.559**	0.545	0.475	0.486	0.460	0.529	0.485	0.485
breast-w	0.968	0.963	**0.969**	0.929	0.929	0.950	0.928	0.926
car	0.870	0.751	0.079	0.858	0.714	**0.943**	0.711	0.879
cleveland	**0.574**	0.232	0.081	0.259	0.000	0.382	0.258	0.149
cmc	**0.637**	0.507	0.517	0.586	0.258	0.543	0.255	0.472
credit-g	**0.611**	0.540	0.569	0.555	0.469	0.602	0.439	0.563
ecoli	**0.830**	0.638	0.701	0.717	0.284	0.554	0.587	0.568
flags	**0.481**	0.025	0.000	0.339	0.000	0.297	0.216	0.000
haberman	**0.576**	0.375	0.334	0.426	0.345	0.468	0.355	0.401
hepatitis	**0.751**	0.604	0.615	0.508	0.050	0.549	0.504	0.502
ionosphere	0.912	0.**928**	0.780	0.878	0.870	0.888	0.874	0.890
new-thyroid	**0.984**	0.951	0.921	0.901	0.915	0.953	0.911	0.878
pima	**0.712**	0.666	0.681	0.649	0.600	0.679	0.581	0.641
postoperative	**0.345**	0.193	0.000	0.000	0.022	0.133	0.055	0.044
scrotal-pain	**0.731**	0.667	0.661	0.676	0.582	0.707	0.662	0.678
solar-flare	**0.638**	0.135	0.000	0.270	0.000	0.319	0.020	0.126
transfusion	**0.639**	0.507	0.529	0.579	0.342	0.602	0.266	0.529
vehicle	**0.935**	0.895	0.914	0.911	0.513	0.919	0.919	0.916
yeast	**0.709**	0.436	0.341	0.511	0.000	0.420	0.452	0.337

Table 6.4: F-measure for BRACID and standard algorithms

Dataset	BRACID	RISE	kNN	C45rules	CN2	PART	RIPPER	MODLEM
abalone	**0.370**	0.192	0.208	0.393	0.253	0.269	0.282	0.326
abdominal-pain	0.718	0.738	**0.751**	0.713	0.704	0.691	0.681	0.694
balance-scale	**0.198**	0.000	0.007	0.019	0.019	0.000	0.000	0.000
b-cancer	**0.438**	0.426	0.364	0.373	0.335	0.389	0.366	0.351
breast-w	0.947	0.949	**0.957**	0.912	0.915	0.932	0.910	0.910
car	0.730	0.665	0.054	0.766	0.680	**0.895**	0.600	0.866
cleveland	**0.332**	0.169	0.059	0.178	0.000	0.225	0.165	0.103
cmc	**0.444**	0.351	0.358	0.434	0.140	0.361	0.124	0.311
credit-g	**0.527**	0.404	0.449	0.426	0.352	0.471	0.311	0.442
ecoli	**0.601**	0.517	0.592	0.593	0.244	0.450	0.473	0.465
flags	**0.240**	0.012	0.000	0.238	0.000	0.204	0.141	0.000
haberman	**0.442**	0.240	0.214	0.300	0.235	0.349	0.233	0.262
hepatitis	**0.603**	0.489	0.538	0.406	0.100	0.452	0.407	0.423
ionosphere	0.878	**0.913**	0.747	0.847	0.850	0.864	0.848	0.872
new-thyroid	**0.970**	0.947	0.895	0.843	0.906	0.918	0.879	0.848
pima	**0.661**	0.577	0.599	0.567	0.512	0.596	0.484	0.550
postoperative	**0.317**	0.158	0.000	0.000	0.016	0.110	0.043	0.032
scrotal-pain	**0.628**	0.563	0.584	0.578	0.493	0.606	0.570	0.585
solar-flare	**0.284**	0.088	0.000	0.170	0.000	0.177	0.015	0.079
transfusion	**0.468**	0.354	0.385	0.443	0.214	0.462	0.149	0.354
vehicle	0.857	0.855	0.877	0.867	0.433	0.875	0.885	**0.892**
yeast	**0.420**	0.311	0.243	0.352	0.000	0.287	0.286	0.245

get a better insight in the comparison of these two classifiers. In this non-parametric test, the null hypothesis is that the medians of measures for the two compared classifiers on all datasets are equal [66, 28]. The ranks are assigned to the values of differences in performance of a pair of classifiers for each dataset – while in Friedman test the winner is only established for a given dataset, without considering how much one algorithm outperforms the other. The p-values resulting from this test are: Sensitivity 0.00089; G-mean 0.00018. All the p-values support our observation that BRACID is significantly better than any of the compared algorithms, also including PART.

We can also discuss some of these results for particular datasets and measures. BRACID can better recognize the minority class than all other classifiers (Table 6.2). The improvements of Sensitivity, sometimes relatively high, are particularly visible if we compare it with its "parents", i.e. RISE and k-NN. The only exception is the *car* dataset, where PART is the best algoritm – we will analyse this case in more detail in Section 6.5. We can also say that this improved recognition of the minority class does not degrade too much the recognition of the majority class – values of G-mean in the Table 6.3 are higher for BRACID than for other algorithms although some differences are smaller. Only for more balanced datasets (e.g. ionosphere – 35%, breast-w – 34%, abdominal-pain – 28%), which according to the experiments in Chapter 3 have a rather *safe* characteristics, the degradation on the majority class is more serious and it influences the G-mean measure. The same observation refers to the F-measure (Table 6.4).

6.4 Experiments with Approaches Dedicated for Class Imbalance

In the previous experiment we could expect the superiority of BRACID over standard rule classifiers as they are not suited to handle imbalanced data. Thus, we include in the comparison some rule-based methods dedicated for class imbalance. Unfortunately, the access to the most interesting algorithms described in Section 4.5 was impossible (most of these algorithms are not available publicly or their authors do not maintain the software anymore). We received a MODLEM-C implementation[2], which is a generalization of the MODLEM algorithm with a modified Grzymala classification strategy [47] – see also Section 4.5. For each dataset separately, we tested 10 possible values of a strength multiplier (from 1 to 10) and chose the best one (according to F-measure and G-mean). Furthermore, as the original RISE uses a less greedy bottom-up induction technique, we can also treat it as better suited for class imbalance. Therefore we include its results in this comparison as well.

As we have been unable to get access to other rule-based approaches dedicated to class imbalance, we have decided to compare BRACID with a rule algorithm combined with specialized data preprocessing methods. First, we direct our interest to a well known SMOTE algorithm [20] as in many experimental studies it has been evaluated to be one of the most efficient preprocessing methods and it has been often used together with rule or tree classifiers. We combine SMOTE with PART rule induction algorithm, as it is the second-best algorithm from the previous experiment. SMOTE is run with $k = 5$ (this value is used in many experiments with SMOTE; it is also consistent with the neighbourhood size used in BRACID) and oversampling ratio tuned for each dataset separately to balance the distribution between classes.

Finally, as SMOTE can sometimes increase the overlapping in the difficult datasets (see discussion in Section 2.3.1), we include SMOTE-ENN which combines SMOTE with cleaning, to get an even more competitive classifier.

We should stress here that our aim in this part of the experiment is not to generally study the preprocessing methods as they are based on different principles than rule algorithms. We want

[2]We thank Dr Szymon Wilk for providing us his implementation.

Table 6.5: Sensitivity for algorithms dedicated for class imbalance

Dataset	BRACID	RISE	MODLEM-C	PART SMOTE	PART SMOTE+ENN
abalone	0.474	0.128	0.274	0.478	**0.582**
abdominal-pain	**0.782**	0.711	0.753	0.738	0.770
balance-scale	**0.565**	0.000	0.000	0.277	0.443
b-cancer	**0.572**	0.356	0.406	0.426	0.482
breast-w	**0.989**	0.959	0.949	0.969	0.983
car	0.781	0.596	0.787	**0.856**	0.749
cleveland	**0.483**	0.147	0.138	0.290	0.470
cmc	0.631	0.293	0.358	0.490	**0.660**
credit-g	**0.801**	0.359	0.551	0.514	0.668
ecoli	0.790	0.505	0.457	0.780	**0.798**
flags	**0.840**	0.020	0.000	0.190	0.190
haberman	0.669	0.224	0.413	0.728	**0.796**
hepatitis	**0.757**	0.487	0.552	0.543	0.573
ionosphere	**0.976**	0.902	0.900	0.889	0.885
new-thyroid	**0.980**	0.928	0.842	0.940	0.938
pima	0.875	0.551	0.720	0.862	**0.890**
postoperative	**0.577**	0.147	0.283	0.170	0.257
scrotal-pain	**0.771**	0.544	0.692	0.697	0.693
solar-flare	**0.517**	0.066	0.192	0.337	0.494
transfusion	0.738	0.297	0.497	0.591	**0.769**
vehicle	**0.960**	0.831	0.920	0.910	**0.960**
yeast	0.555	0.245	0.209	**0.628**	0.505

to check whether BRACID is not worse or competitive to the well known representatives of these methods.

The results, presented in Tables 6.5-6.7, show as previously Sensitivity, G-mean and F-measure. Comparing the standard MODLEM algorithm with MODLEM-C proves that modified classification strategy helps to deal with imbalanced classes (Table 6.2). Similarly, PART+SMOTE and PART+SMOTE+ENN work better than PART alone. We conducted again the Friedman test for all the classifiers. For all measures we can reject the null hypothesis. Critical values are: Sensitivity 54.94; G-means 34.78; F-measure 27.76. In the post-hoc analysis the critical difference CD is equal to 1.3 (with Nemenyi test).

The average ranks are the following: Sensitivity – BRACID 4.5; PART+SMOTE+ENN 3.84; PART+SMOTE 3.11; MODLEM-C 2.11; RISE 1.43. G-mean – BRACID 4.3; SMOTE+ENN +PART 3.65; SMOTE+PART 2.84; MODLEM-C 2.29; RISE 1.88. F-measure – BRACID 4.22; PART+SMOTE+ENN 3.52; PART+SMOTE 2.88; MODLEM-C 2.25; RISE 2.11W. With the critical difference CD=1.3 we cannot say that BRACID is significantly better than PART+ SMOTE+ENN; however the difference between them is around 1 in favor of BRACID. All other algorithms are ouperformed by BRACID. In all cases RISE is the worst algorithm while MODLEM-C is worse than PART combined with SMOTE.

Again we applied the Wilcoxon signed rank test to verify more deeply the differences between BRACID and PART+SMOTE+ENN. With respect to Sensitivity, BRACID is significantly better than PART+SMOTE+ENN ($p < 0.0308$). For G-mean and F-measure, BRACID is better with $p < 0.043$ and $p < 0.028$, respectively. Determining win-loss between them also shows that BRACID dominates PART+SMOTE+ENN for 14 or 15 datasets, depending on the evaluation measure. It is defeated (for all measures) on only two datasets: *abalone* and *haberman*. To sum up, we can conclude that BRACID's classification performance is comparable or even slightly better than the

Table 6.6: G-mean for algorithms dedicated for class imbalance

Dataset	BRACID	RISE	MODLEM-C	PART SMOTE	PART SMOTE+ENN
abalone	0.650	0.345	0.513	0.643	**0.704**
abdominal-pain	0.811	0.805	0.793	0.790	**0.818**
balance-scale	**0.567**	0.000	0.000	0.346	0.462
breast-cancer	**0.559**	0.545	0.530	0.526	0.540
breast-w	**0.968**	0.963	0.947	0.959	0.962
car	0.870	0.751	0.879	**0.916**	0.842
cleveland	**0.574**	0.232	0.225	0.410	0.565
cmc	**0.637**	0.507	0.544	0.581	0.635
credit-g	0.611	0.540	0.645	0.612	**0.658**
ecoli	**0.830**	0.638	0.633	0.826	0.826
flags	**0.481**	0.025	0.000	0.224	0.224
haberman	0.576	0.375	0.532	**0.608**	0.596
hepatitis	**0.751**	0.604	0.644	0.639	0.656
ionosphere	0.912	**0.928**	0.898	0.876	0.868
new-thyroid	**0.984**	0.951	0.903	0.955	0.955
pima	**0.712**	0.666	0.704	0.681	0.660
postoperative	**0.345**	0.193	0.297	0.158	0.251
scrotal-pain	0.731	0.667	0.729	0.716	**0.732**
solar-flare	0.638	0.135	0.322	0.492	**0.651**
transfusion	**0.639**	0.507	0.579	0.601	0.621
vehicle	0.935	0.895	0.941	0.932	**0.942**
yeast	0.709	0.436	0.370	**0.749**	0.658

Table 6.7: F-measure for algorithms dedicated for class imbalance

Dataset	BRACID	RISE	MODLEM-C	PART SMOTE	PART SMOTE+ENN
abalone	0.370	0.192	0.353	0.350	**0.372**
abdominal-pain	0.718	**0.738**	0.695	0.695	0.733
balance-scale	**0.198**	0.000	0.000	0.131	0.171
b-cancer	**0.438**	0.426	0.390	0.392	0.405
breast-w	0.947	**0.949**	0.925	0.940	0.940
car	0.730	0.665	**0.864**	0.823	0.602
cleveland	**0.332**	0.169	0.157	0.223	0.318
cmc	**0.444**	0.351	0.372	0.386	0.442
credit-g	0.527	0.404	0.524	0.481	**0.539**
ecoli	0.601	0.517	0.512	**0.618**	0.571
flags	**0.240**	0.012	0.000	0.162	0.162
haberman	0.442	0.240	0.370	**0.491**	0.483
hepatitis	**0.603**	0.489	0.535	0.511	0.525
ionosphere	0.878	**0.913**	0.867	0.839	0.826
new-thyroid	**0.970**	0.947	0.869	0.918	0.922
pima	**0.661**	0.577	0.627	0.639	0.630
postoperative	**0.317**	0.158	0.217	0.121	0.178
scrotal-pain	0.628	0.563	0.625	0.608	**0.634**
solar-flare	0.284	0.088	0.193	0.228	**0.319**
transfusion	**0.468**	0.354	0.395	0.445	**0.468**
vehicle	0.857	0.855	**0.902**	0.886	0.874
yeast	**0.420**	0.311	0.263	0.335	0.327

best method for informed re-sampling used together with the most competitive rule algorithm PART.

6.5 Analysis of Rule Sets

To analyse the differences between the induced rule sets for the selected algorithms, we additionally calculate some descriptive statistics of rule evaluation measures such as the average number of rules and their average support for each class separately. These values characterize the differences in the rule induction phase but they may also help to interpret the results of applying the classification strategies. We do not compare the confidence of the rules as all the selected algorithms (with the parameters used, such as disabled pruning in MODLEM), induce nearly equally confident rules. Moreover, we do not present the average length of a rule, as BRACID never drops conditions on numerical attributes while other algorithms use more greedy strategies, so it would be misleading. As PART and C45rules return an ordered set of rules, and RIPPER learns rules for only one class, we have decided to compare the sets of rules for RISE, CN2 and MODLEM only. The results, presented in Table 6.8, include additionally the number of maximally specific rules representing single minority examples in the final BRACID classifier (called instances). This parameter shows how hybrid is the knowledge representation corresponding to the minority class for a given dataset. It could also refer to the difficulty of data – a lot of maximally specific minority rules suggests a limited number of large disjuncts and possible other difficulties with the minority class distribution. To verify this hypothesis, at the end of this experiment we will analyse the types of examples (estimated using our method introduced in Chapter 3) which are the seeds of these rules.

We present the results for 9 selected datasets. The first 5 datasets (over the double horizontal line) represent the standard behaviour of the algorithms, which we also observed for the remaining 13 datasets, so there is no need to include them in this Table. The last 4 datasets represent different untypical situations which we will further discuss. Typically (first 5 datasets), BRACID and RISE generate more rules than CN2 and MODLEM algorithms. This is due to the fact that CN2 and MODLEM represent a maximum generality bias and try to induce a minimal set of rules. BRACID induces more minority rules than RISE, because in the difficult regions it allows to create more rules for unsafe examples from the minority class. However, it is important to note that although BRACID generates much more minority rules, they are characterized by the highest average support comparing to rule supports from other algorithms.

It is interesting to check whether an increase of a number of rules (even if they are strong) is always profitable. For instance, for the *transfusion* dataset, BRACID generates 6 times more minority rules than CN2. However, if we come back to Table 6.2 and analyse the sensitivity measure on this dataset, it can be observed that it improves the recognition of the minority class from 15% to 73%. For *hepatitis*, CN2 covered the minority class with less than four rules (while BRACID with 60), however the recognition for CN2 was 5% and for BRACID – 75%. So, in our opinion the trade-off is worth it.

Under the double horizontal line in Table 6.8 we present 4 datasets for which the results were in a way untypical. For *solar-flare* and *cleveland*, CN2 generated much stronger minority rules than BRACID. An analysis of Table 6.2 shows, however, that these rules were not useful – they could not correctly classify even a single testing example. The same refers to *abdominal-pain* dataset, for which BRACID also generates less rules than CN2. This time, although CN2 still achieves worse performance on Sensitivity, G-mean and F-measure, its results are more comparable to those of BRACID. Let us note that this dataset is more balanced than other datasets (28%), which may be the reason why CN2 (and other learning algorithms, according to the experiments in Chapter 3)

Table 6.8: Average values of evaluation measures for rulesets

Dataset	Classifier	No of rules (MIN)	No of rules (MAJ)	No of instances	Support (MIN)	Support (MAJ)
balance-scale	CN2	39.92	47.18		1.51	42.12
	MODLEM	43.48	48.04		1.02	41.88
	RISE	42.96	104.40		1.31	79.43
	BRACID	65.32	124.04	10.48	7.82	14.83
b-cancer	CN2	22.60	34.88		2.77	6.09
	MODLEM	32.46	36.94		3.04	7.20
	RISE	52.68	73.12		2.45	7.99
	BRACID	64.60	61.54	18.10	4.76	5.78
hepatitis	CN2	3.66	4.14		4.00	15.68
	MODLEM	4.88	5.42		7.78	30.17
	RISE	22.18	47.60		5.12	16.58
	BRACID	60.88	46.54	1.38	7.03	19.57
new-thyroid	CN2	2.70	3.20		17.71	140.63
	MODLEM	2.76	2.54		19.10	133.17
	RISE	9.72	20.98		13.23	112.15
	BRACID	19.18	20.70	0.04	23.18	116.76
transfusion	CN2	23.00	37.24		9.86	17.85
	MODLEM	59.02	63.36		6.32	14.59
	RISE	101.08	110.66		7.86	14.62
	BRACID	146.02	109.06	21.00	11.90	11.06
solar-flare	CN2	11.30	29.02		30.49	59.39
	MODLEM	20.24	18.18		5.55	107.20
	RISE	32.64	48.42		4.10	58.20
	BRACID	34.50	64.08	11.70	7.55	37.14
cleveland	CN2	9.76	13.02		10.64	44.71
	MODLEM	11.82	14.20		2.91	37.33
	RISE	19.10	83.66		4.22	16.02
	BRACID	84.52	81.20	2.50	5.71	17.05
abdominal-pain	CN2	17.98	38.04		17.51	36.48
	MODLEM	41.32	41.52		9.54	35.49
	RISE	57.40	110.44		8.05	14.93
	BRACID	71.44	100.46	4.44	12.90	13.59
car	CN2	30.34	16.00		2.21	215.76
	MODLEM	14.02	12.00		5.07	270.31
	RISE	45.38	328.92		1.85	17.54
	BRACID	35.74	164.14	12.28	2.95	32.29

can learn it reasonably well. The same behaviour was observed for another dataset not included in Table 6.8, *ionosphere*, which is even more balanced (36%). Also, according to our analysis of the data characteristics in Chapter 3, *abdominal-pain* and *ionosphere* datasets are *safe* datasets so they should be easier to learn for standard classifiers such as CN2.

We report the rule statistics also for *car* dataset, to check why BRACID performed on it worse than classical PART algorithm and comparably to MODLEM. According to our analysis in Chapter 3, this dataset has a lot of borderline examples. The analysis of the ruleset suggests additionally, that the distribution of examples in the borderline might be very scattered – almost 35% of the final BRACID classifier corresponds to single cases. As this dataset consists of only nominal attributes, a generalization of a rule to another example in BRACID can be achieved only by dropping the whole condition on a given attribute, which might be difficult in a mixed borderline region without covering too many examples from the other class. As a result, the average rule strength of BRACID

is rather low compared to other algorithms on this dataset. Also, our algorithm did not manage to create many additional rules for the difficult minority class examples, which would satisfy the leaving-one-out evaluation procedure. BRACID created a comparable number of rules to other algorithms, which are comparably strong – which may be a reason why it could not outperform other algorithms.

Finally, we have analysed the seed examples for the maximally specific rules. Using our labelling method from Chapter 3, we wanted to analyse the type of these examples, expecting that they will be mostly rare and outlying examples. For most of the datasets, almost 100% of examples are of type rare or outliers. The exceptions are *flags* dataset (which had only two such rules) and *car* dataset. The detailed results can be found in Appendix A (Table 8).

6.6 Applicability of the Algorithm

In this experiment we want to analyse what is the area of competence for BRACID. As we have shown in Chapter 3, imbalanced datasets can have different characteristics – they can consist of safe, overlapping, rare or noisy/outlying examples. We have also shown that classifiers can reveal different sensitivity to the particular types of examples. Although the experiments carried out in Sections 6.3 and 6.4 have shown that BRACID works well on all imbalanced datasets regardless of their data distribution, we want to observe for which data distributions BRACID is the most well suited, by recording errors made on each type of minority class examples.

To carry out this analysis, we first identify a type of each learning example. Then, in the cross-validation procedure, we record the sensitivity for each type of testing examples separately. Analogously to the experimental setup in Chapter 3, we do not present the results for too small datasets (less than 300 examples) to ensure that there are enough representatives in each category. As some datasets do not have any examples of a particular type or their number is too small (e.g. cleveland, vehicle) – we left the corresponding cells empty. Note that, compared to the experiment in Chapter 3, three additional datasets have been used in this experiment – breast-w, balance-scale and pima. Therefore, we give their labelling results in Table 6.9. In general, breast-w represents safe datasets, balance-scale consists mostly of outliers and pima is a borderline dataset.

Table 6.9: Labelling of datasets

Dataset	S [%]	B [%]	R [%]	O [%]
breast-w	91.29	7.88	0.00	0.83
pima	29.85	56.34	5.22	8.58
balance-scale	0.00	0.00	8.16	91.84

We compare the results of BRACID with two classifiers: PART – a classic rule learning classifier which according to the experiments in Chapter 3 worked well for all types of examples, especially for rare and outlying ones, and PART combined with SMOTE-ENN – a classifier dedicated for class imbalance, which was the most competitive to BRACID in the experimental setup carried out in Section 6.4. Table 6.10 presents these results – for each dataset and the three compared classifiers, local accuracies (sensitivities) on each type of testing examples are grouped together. The best results are marked with bold.

First of all, we can observe that both approaches dedicated for class imbalance improve the recognition of all four types of testing examples compared to PART. The improvements on the *safe* examples are small, because PART alone can already deal very well with these examples (recognizing correctly 80-90% of them). However, BRACID can additionally improve the recognition of these examples, often reaching a hundred percent accuracy.

Table 6.10: Sensitivity on labeled testing examples

Dataset	PART	PART SM+ENN	BRACID	PART	PART SM+ENN	BRACID
		Safe			Border	
abalone	0.743	0.993	**1.000**	0.278	**0.792**	**0.792**
abdominal-pain	0.917	0.932	**0.985**	0.695	**0.836**	0.805
balance-scale						
breast-w	0.975	0.994	**1.000**	0.720	0.907	**0.973**
car	**0.915**	0.891	0.879	**0.911**	0.556	0.704
cleveland				0.450	**0.725**	0.650
cmc	0.631	0.963	**0.983**	0.375	0.746	**0.769**
credit-g	0.657	0.836	**0.979**	0.533	0.720	**0.886**
ecoli	0.840	0.960	**1.000**	0.329	**0.882**	0.847
haberman	0.900	**1.000**	**1.000**	0.482	**0.965**	0.865
ionosphere	0.986	0.986	**1.000**	0.921	0.994	**1.000**
pima	0.865	**0.993**	0.990	0.620	**0.948**	0.938
solar-flare				0.275	0.563	**0.763**
transfusion	0.861	**1.000**	0.994	0.645	**0.916**	0.887
vehicle	0.931	**0.982**	0.978	0.741	0.900	**0.959**
yeast	0.733	0.933	**1.000**	0.500	0.811	**0.900**
		Rare			Outlier	
abalone	0.124	**0.602**	0.555	0.104	**0.443**	0.253
abdominal-pain	0.171	**0.229**	0.105	**0.088**	0.075	0.075
balance-scale	0.000	0.500	**0.800**	0.000	0.440	**0.547**
breast-w						
car	**1.000**	0.833	0.833	0.467	**0.733**	0.400
cleveland	0.222	0.489	**0.644**	0.167	**0.344**	0.333
cmc	0.349	**0.567**	0.520	0.191	**0.378**	0.246
credit-g	0.405	0.653	**0.749**	0.320	0.444	**0.532**
ecoli				0.120	0.240	0.240
haberman	0.206	**0.755**	0.568	0.050	**0.350**	0.267
ionosphere	0.676	0.686	**0.952**	0.375	0.575	**0.875**
pima	0.276	**0.747**	0.720	0.113	**0.513**	0.452
solar-flare	0.320	**0.780**	0.720	0.024	**0.271**	0.212
transfusion	0.212	**0.647**	0.600	0.016	**0.510**	0.473
vehicle						
yeast	0.200	0.480	**0.580**	0.020	**0.180**	0.160

When the *borderline* testing examples are concerned, PART's results are around 60%. BRACID works very well here – it is often better than SMOTE-ENN and it improves the recognition of these examples up to about 90%.

As for the *rare* and *outlying* examples, PART usually cannot recognize more than 20% of them. SMOTE-ENN and BRACID can rise this number to about 60-80% on the rare examples, while on the outliers the improvements are smaller – usually less than 50%. For the rare examples, both approaches work comparably, while on the outliers SMOTE-ENN is a better algorithm. There may be two reasons why BRACID does not concentrate that much on the outlying examples: the `Extend` operator which we have decided to use in the "conservative" way and not apply it for the most specific rules (which often cover outlying examples), and the classification strategy which in case of two equally distant conflicting rules makes a decision based on the rule support (which might be low for rules covering outlying examples). If we would apply the `Extend` operator in a more aggressive way (extending all the rules) and apply a classification strategy more biased towards the minority class, the results on outlying examples might be better. However, in our preliminary experiments we have tested a more "aggressive" use of the `Extend` component, extending all the rules, as well as some classification strategies biased more towards the minority class (e.g. assigning the minority class label whenever any minority rule was in the conflict set) – these approaches indeed improved the recognition of the minority class, but jeopardised the majority class too much.

The good behaviour of SMOTE-ENN on these examples may be a result of introducing artificial minority examples around the outlier examples by SMOTE. Especially for the datasets with high imbalance ratio, where the level of oversampling is high, introducing new examples in these regions may help to recognize the rare and outlier examples. However it may have negative consequences for the majority class.

6.7 Conclusions

The most important features of BRACID, which in our opinion should improve classifiers constructed from imbalanced data, are the following:

- It produces an integrated hybrid representation of rules and instances to use their complementary advantages, i.e., it uses rules to generalize consistent regions and instances to better represent the overlapping regions, rare or outlier examples in the data.
- It induces rules in the bottom-up direction and it does not use a sequential covering technique to prevent the data fragmentation and to better handle possible small disjuncts.
- It uses the F-measure in a leaving-one-out procedure to evaluate and accept these rules that are more capable of recognizing the minority class.
- It uses the local nearest rule classification strategy which diminishes the role of the global domination of the majority rules when making a classification decision.
- It removes noisy examples from the majority classes to prevent the fragmentation of the minority class regions.
- It extends the minority rules and allows to analyze more generalizations to rules in the consistent regions in order to address the problem of under-representation of the minority class in the data.
- It creates more minority class rules in the overlapping regions to decrease the chance of overwhelming the minority class by the majority classes. All these rules are generated from

the actual learning examples. This significantly distinguishes BRACID from the preprocessing methods based on oversampling (e.g., an original version of SMOTE generates quite a large number of artificial examples which could lead to ambiguity either in a human interpretation of a rule or while explaining the classification decisions for new coming examples) or from modified classification strategies using so-called strength amplifiers (MODLEM-C) to *artificially* amplify the importance of minority class in the set or conflicting rules. This property of BRACID is crucial for getting a more comprehensible and transparent knowledge representation.

We have conducted an extensive experimental evaluation on 22 imbalanced datasets, where we have compared BRACID to a number of the state-of-the-art rule classifiers, one instance-based classifier and some approaches dedicated for class imbalance. The main conclusions from these experiments are the following:

- BRACID significantly (with respect to the non-parametric Friedman test and the post-hoc analysis) outperforms other standard rule classifiers, as well as its "parent" approaches – RISE and kNN. Moreover, according to the results of Wilcoxon test, BRACID performs better than the most competitive rule algorithm – PART.
- BRACID is able to better recognize the minority class than other compared algorithms (except for one dataset, car, for which PART is superior);
- The improvement of the sensitivity measure is associated with a limited deterioration of specificity; also global measures as F-measure and G-mean are improved by BRACID. Only for nearly balanced datasets, a slight decrease in F-measure and G-mean has been sometimes observed.
- What is even more important, the classification performance of BRACID (with respect to all measures) has been better than other compared approaches dedicated for class imbalance, including the integration of the PART algorithm with the basic version SMOTE. Only after extending SMOTE by the Edited Nearest Neighbor Rule (ENN), the difference of average ranks between this approach (PART+SMOTE+ENN) and BRACID has become insignificant according to the Friedman test with the post-hoc analysis. The last result is not a drawback as SMOTE + ENN is a specialized informed re-sampling approach and it is a well known, effective solution at the data level. Moreover, BRACID could be seen as better with respect to the additional paired Wilcoxon test and a win-loss analysis.
- BRACID induces a classifier containing more rules, especially for the minority class, compared to classifiers induced by standard rule algorithms. At the same time, the average support of the BRACID rules from the minority class is higher than for other classifiers. Such rules can be effectively applied within the new proposed classification strategy based on the nearest rules.
- When the types of testing examples are concerned, BRACID can improve the recognition of all types of examples, but it handles better the borderline examples and concentrates less on the outlying examples. Therefore, we can conclude that BRACID is particularly well suited for the imbalanced datasets with high overlapping of classes. Let us observe that, according to the results presented in Chapter 3, borderline examples are the most common type of examples in the imbalanced datasets. For the datasets with a lot of outlying examples, PART combined with SMOTE-ENN might be worth considering (due to the effect of SMOTE discussed in the previous Section).

Chapter 7

ABMODLEM: Addressing Imbalanced Data with Argument-based Rule Learning

7.1 Motivations

In this Chapter we direct our interest to using the expert's knowledge in learning rules from imbalanced data. One of the problems with class imbalance is that the minority class is underrepresented in the data. What is more, it may be further separated into smaller subconcepts or the examples from both classes may overlap. In such case, it may be difficult for standard, automatic learning algorithms to find meaningful rules for this class, having a good interpretation for an expert. If a subconcept is represented by only few minority examples, there may exist several possible rules which cover it, but not all of them will be consistent with the user's expectations and the expert's knowledge concerning the domain of application. As it was pointed out in [139], incorporating knowledge in the learning process is especially useful when rare classes/cases are present, since the user may have domain knowledge that can aid in the search process – for example, he may help to distinguish the features that are useful for predicting rare, but important, cases [139]. It can lead to the induction of rules which will be consistent not only with the learning examples, but also with the domain knowledge. Furthermore, it could lead to better classification abilities.

Using the *background knowledge* in the induction of rules has already been considered by many researchers, although not in the context of class imbalance – see, e.g., generalizations of AQ algorithms [87], and in particular rule approaches in Inductive Logic Programming systems [73]. For other rule-based approaches proposed in the context of data mining see, e.g, a review [64]. In these approaches it is assumed that the experts express their *"global"* knowledge, valid for the whole domain of application. For example, the constraints given by an expert can concern the relation between the attributes, which has to be true for all the data. The learning algorithm must take these constraints into account during the entire induction process. However, expressing such "global" knowledge is often difficult for humans.

Recently, Bratko *et al* introduced in [93, 94] a new paradigm called *argument based machine learning* (ABML) which enables expressing the domain knowledge in a more natural way. A key concept of their approach is to let the expert annotate some of the learning examples. An expert can describe reasons for assigning the example to a given class, which are called *arguments*. This approach uses a *"local"* expert's knowledge which concerns only specific situations and can be valid for limited, chosen examples rather than for the whole domain [93]. Note that the idea of explaining the decisions for the selected examples may be well suited for handling rare and outlier minority examples. The reasons for assigning such example to the minority class may be very specific and atypical, so expressing the "global" knowledge for such examples, valid in the whole domain, might

not be feasible. Also, this kind of explanation is natural for such domains as justification of cases in law, discussing circumstances of making some decisions in finance or medicine, which are often characterised by class imbalance.

A somewhat related approach was proposed by Plaza and Ontanon in the context of multiagent inductive learning, where agents learn rules from separate data samples and argumentation of learning examples is used as a communication framework between the agents to share their local knowledge and reach a consensus on a final definition of the problem [105]. However, in this framework no external expert knowledge is used – the arguments passed to another agent are generated automatically from the agent's local sample. The results show that this distributed learning protocol allows to construct a classifier which has a comparable performance to a classifier obtained using the classic, centralized approach.

Although the ABML framework can be used with any algorithm, it is the most suitable for rule induction, as both induced hypotheses and arguments are represented in the same language and the influence of expert's arguments is explicitly visible in the induced rules [94]. The ABML has been originally implemented as an extension of a CN2 algorithm (called ABCN2 – *argument based* CN2 [93]) and it has been succesfully applied to the problems of justification of cases in law, loan policy, chess strategy and medical treatment, see [146, 96, 95].

Let us recall that these works were not carried out in the context of class imbalance and they have been evaluated using the standard measures such as the total accuracy. However, we think that argument-based framework is very well suited for handling imbalanced data as the expert can help identify the features that are useful for predicting difficult minority cases.

In this Chapter we want to verify whether the ABML framework can be used to improve rule learning from imbalanced data. We expect that the expert's annotations might help to create better rules for difficult (e.g. rare and outlier) minority examples, leading to a better recognition of this class. We have decided to use the ABML framework with other learning algorithm than CN2 – MODLEM [9] – which is more suited to deal with numerical and imperfect data and which was already used in the context of class imbalance (see Chapter 6). Also, according to the experiments in Chapter 6, MODLEM was a better classifier than CN2 on imbalanced datasets. Although our generalization, called ABMODLEM, is inspired by the paper [94], we have to consider new methodological aspects. First of all, it is necessary to introduce another measure for evaluating candidate elementary conditions to be added to a rule, which allows to obtain more general rules, in particular the ones covering argumented examples. Secondly, while classifying new objects with rules, a new classification strategy is required which takes into account that rules induced from argumented examples are usually supported by fewer examples than non-argumented rules.

In this study, we concentrate in particular on the *identification of examples to be argumented* by an expert, as it is a crucial issue for the effects of ABML. In problems described with a relatively small number of examples, an expert could know them all and determine the necessary examples manually. However, for larger problems or carrying out experiments with many datasets, it may be impossible for humans and a more automatic support for selecting the examples is necessary. In order to identify them Bratko *et al.* already proposed to focus attention on the "problematic" examples (for which the arguments would be likely to improve learning), which were understood as examples incorrectly classified by a set of rules induced from plain examples without any arguments [93]. They introduced an iterative approach, which includes selecting and argumenting only one example at a time, and re-learning the entire set in each iteration. The example is identified as the most often misclassified example in an internal k-fold cross validation technique used with the rule induction by the classic version of CN2. Although it was applied in some experiments, e.g., [93, 95], this technique may still select quite high a number of critical examples with the same number of

misclassifications, which leads to ambiguity. Moreover, for large datasets the iterative procedure is too computationally costly and requires a tiresome co-operation with an expert. Therefore, we suggest that it is more reasonable to construct an approach for selecting the sufficient number of examples in one step. We use some inspirations from active learning [85], where an analysis of difficulties with classifying the examples is used to select the smallest subset of unlabelled learning examples for which the class labels should be produced. It uses an ensemble of classifiers and the idea of disagreement of its answers. Our hypothesis is that the method should select in one step a sufficient number of examples which, after being argumented, provide a substantial improvement of the recognition of the minority classes. At the same time, this improvement should not come at a too high cost of the majority class recognition, improving also the aggregation measures such as G-mean, F-measure, or even total classification accuracy.

The other related research problem concerns studying the influence of selecting examples to be argumented on the recognition of particular classes. Most of the learning algorithms dedicated for class imbalance face a trade-off between the recognition of minority and majority classes – they usually can improve the recognition of the minority class, but at the cost of deteriorating the majority class – see, e.g., the experimental results in Chapters 3 and 6. In argument-based learning, explaining the examples from the given class may not only influence induction of rules from this class but it may also affect other classes. Therefore, we want to check whether an appropriate selection of examples for argumentation can improve the recognition of minority classes without deteriorating the recognition of the majority classes.

7.2 Notation and Basic Concepts

We briefly present the concepts of ABML that are necessary for introducing our proposal. We follow the most related works of Bratko, Mozina *et al* and their notations [94], presented in a most complete form in a Ph.D. dissertation of Martin Mozina, [92]. It is assumed that some of the learning examples are enhanced by partial justifications given in a form of arguments. Each argument is attached to a single learning example only, while one example can have several arguments. There are two types of arguments; *positive arguments* are used to explain why a learning example is assigned to a given class, and *negative arguments* are used to explain why it should not belong to a given class. Examples enhanced with arguments are called *argumented examples.*

The task of learning with argumented examples is given as:

Definition 7.1 *Given examples + supporting arguments for some of the examples, find a hypothesis (set of rules) that explains the examples using given arguments.*

Argumentation

Similarly to the definition given in Section 1.1, learning examples are represented as $(\mathbf{x}, y)$, where $\mathbf{x}$ is a vector of *attribute-value* pairs called a description of the example, and $y \in Y$ is a value of a class label. Additionally, in ABML some of the examples are enhanced by arguments. An argumented example AE is denoted as a triple

$$AE = (\mathbf{x}, y, Arguments)$$

where $\mathbf{x}$ and y are defined as in standard examples. *Arguments* is a set of arguments $Arg_1, ..., Arg_n$, where an argument Arg_i has one of the following form:

Positive argument defined as y *because of Reasons*

Negative argument defined as *y despite Reasons*

Reasons are expressed as conjunctions of *attribute-value* expressions r_i which take a form similar to elementary conditions used in the syntax of a rule. So, for nominal attributes $x \in X$ an elementary condition is of a form $(x_i = v_{i,j})$ and for numerical attributes of $(x_i \; rel \; v_{i,j})$ where $v_{i,j}$ is a value of attribute x_i and *rel* stands for an operator $<, \leq, =, \geq, >$.

A set of rules is said to explain the examples using given arguments, when there exists at least one rule for each argumented example that contains at least one positive argument in its condition part, and when it does not contain any of the negative arguments.

Argumentation leads to redefining the idea of covering examples by rules, which is called *argument-based covering*. We say that a rule *AB-covers* an argumented example if it satisfies the requirements given in Definition 7.2.

Definition 7.2 *Rule R AB-covers an argumented example AE if:*
(1) all the conditions in R are true for the description of AE,
(2) condition part of R is consistent with at least one positive argument of AE,
(3) it is not consistent with any of the negative arguments of AE,
where consistency means that the condition part of R contains elementary conditions r_i in the same form as expressions in reasons *or r_i are their generalizations (i.e. a condition r_i is a superset of the analogous expression in reasons).*

To illustrate the above mentioned concepts, let us analyse a toy example. Assume that we have a medical information for 3 patients of a hospital. Patients' state is described by temperature, presence of stomach ache, blood test results and blood pressure. For each patient, a decision was taken whether to admit him to a hospital or not.

Patient	Temperature	Stomach ache	Blood test result	Blood pressure	Admitted
Johns	high	no	bad	normal	yes
Biggle	normal	no	bad	v.high	yes
Perkins	high	yes	good	normal	no

If one wants to discover a rule explaining the decision on admitting a patient to the hospital, a typical rule induction algorithm could produce the following rule:

if (stomach ache = no) then (admitted = yes),

which covers all the positive examples of this decision. On the other hand, this rule contradicts the common sense.

A physician asked to explain why Johns was admitted to hospital could explain it as (giving a positive argument):

"Johns was admitted to hospital because his body temperature = high"

He could also explain the decision by giving a negative argument:

"Johns was admitted to a hospital despite stomach ache = no"

Notice that this argumentation is "local" – the physician claims that Johns was admitted to hospital because of the temperature, but he does not claim that all patients with high temperature are automatically admitted (as is the case of Perkins). This argumented example is formally denoted as:

$$
\begin{aligned}
AE = (&\boldsymbol{x} = \{Johns,\ high,\ no,\ bad,\ normal\},\\
&y = yes,\\
&Arguments = \{y = yes\ because\ temperature = high,\\
&\qquad\qquad y = yes\ despite\ stomach\ ache = no\})
\end{aligned}
$$

This argumentation can be used to direct the induction process. It shows that the temperature was an important factor when the decision for Johns was taken, and at the same time lack of stomach ache was not relevant for this decision. Therefore, in a construction of a rule AB-covering this example, the temperature attribute should be favoured (to preserve consistency with the positive argument), while stomach ache attribute should be neglected (to prevent consistency with the negative argument). An algorithm using this argumentation would therefore induce a rule:

$$if\ (temperature = high)\ and\ (blood\ test\ results = bad)\ than\ (admitted = yes)$$

which in turn is a rule cosistent with a common sense and the expert's knowledge. Also, note that arguments impose constraints over the space of possible hypotheses, thus they can reduce the search complexity.

Rule Learning with Arguments

A general framework for argument-based rule learning was proposed in [94]. The pseudo-code is given in Algorithm 7.1. First, arguments given for examples are evaluated as simple rules, and argumented examples are sorted from the most to the least "general" ones, according to a chosen evaluation measure. Then, an algorithm induces rules consistent with argumented examples – if necessary, it adds additional elementary conditions to the expressions from the arguments. Starting from the most general argumented examples helps to induce rules which cover not only the argumented examples, but possibly also some other non-argumented examples. Rule induction follows a typical sequential covering schema [23] – when the first rule is generated, all examples covered by it are removed from consideration and the algorithm looks for the next best rule until the set of examples to be AB-covered is not empty. As after the induction of rules from arguments some non-argumented examples may still remain not covered, the following rules are induced with the standard learning procedure `LearnOneClass` (which usually looks for the best rules in a sequential covering schema as, e.g,. in classic CN2 [94]).

This framework is general enough to be used with any rule learning algorithm to evaluate the argumented examples (line 3 of the algorithm) and to induce the rules covering non-argumented examples (line 13), provided that it is enhanced by the possibility of inducing rules AB-covering argumented examples (line 6). As this framework assumes that rules are induced for every class separately, it can be used for both binary and multiclass problems. Bratko *et al* propose to use the enhanced version of the well known CN2 algorithm [23] (shortly called ABCN2). The authors modified the beam search procedure such that it guarantees to AB-cover the given argumented example. They also replaced the original Laplace rule evaluation measure [23] with the *extreme value correction* method (EVC) based on the *m-estimate* [29], as the Laplace measure proved to underestimate the quality of rules induced from arguments. Finally, the authors noticed that the rules learnt from arguments usually have a too small impact while solving ambiguous situations in classifying unseen objects. Following this observation, a new classification strategy was introduced in ABCN2, in which each class is represented by only one rule with the best quality according to EVC [94].

Algorithm 7.1 *General framework for ABML*

```
Procedure ABLearnOneClass (Learning examples ES,  Class Y)

 1.Let RULE_LIST be an empty list.
 2.Let AES be the set of examples that have arguments.
 3.Evaluate arguments (as if they were rules) and sort examples
            according to the evaluation of their best argument.

 4.while AES is not empty do
 5.    Let AE1 be the first example in AES.
 6.    Let BEST_RULE be ABFindBestRule(ES,AE1,Y).
 7.    Add BEST_RULE to RULELIST.
 8.    Remove from AES examples AB-covered by BEST_RULE.
 9.end while

10.for all RULE in RULE_LIST do
11.    Remove from ES examples covered by RULE.
12.end for

13.Add rules obtained with LearnOneClass(ES,Y) to RULE_LIST.
```

An important question in ABML is how to identify the examples for argumentation if the expert does not directly choose them. The discussion of Bratko *et al* proposal is shifted to Section 7.5, where our methods are also presented.

7.3 Algorithm Description

We incorporate the ABML paradigm inside MODLEM algorithm, originally introduced by Stefanowski in [118]. Similarly to CN2, it also follows a sequential covering schema and generates a minimal set of unordered rules. It iteratively searches for the best rule for a given class, removes all covered positive examples from the learning set and continues the procedure until all the examples from that class are covered. The process is repeated for each decision class. A construction of a single rule starts from finding the best condition according to an evaluation measure, and continues by adding new conditions until a stopping criterion is met. The specific property of MODLEM consists in direct processing numerical values of attributes (without pre-discretization) and missing values. It can also be adopted to handle inconsistent or noisy examples either by rule pruning or by rough approximations. Details of MODLEM can be found in [118, 119, 120]. Yet another reason for choosing this algorithm is that it has already been studied in the context of imbalanced data [124, 125].

Our algorithm, called ABMODLEM, uses the general framework of ABML, given in Algorithm 7.1. To induce the rules covering non-argumented examples (with `LearnOneClass` procedure – line 13), we use the standard MODLEM procedure. Key difference between ABMODLEM and ABCN2 is another way of constructing the best rule (line 6). This procedure is presented below as Algorithm 7.2.

Construction of a Rule from Arguments

To find the best rule that AB-covers an argumented example, an initial rule is built from each positive argument (to assure coherence with at least one positive argument given for the example) – i.e. a conjunction of elementary expressions is taken as a candidate for a condition part (line 3 in Algorithm 7.2). If the stopping criterion is not met (e.g. the rule still covers some negative examples from other classes), additional conditions are iteratively added to the rule condition part by the `ABFindBestCondition(a,S,AE,RULE)` procedure.

This procedure finds the best condition by comparing candidate conditions for each attribute, assuring the coherence with arguments and evaluating them with respect to a chosen measure (which will be described in the "Rule Evaluation Measure" paragraph). The construction of elementary conditions depends on the type of the attribute:

1. For nominal attributes, as the rule must cover the argumented example, the condition must take the form *attribute* = *value*, where value comes from the description of the argumented example. If adding this condition to the rule will cause the consistency with any negative argument for the considered example, this condition is skipped.

2. For numerical attributes, conditions are in form of $x_i > v_i$, $x_i < v_i$, $x_i \geq v_i$ or $x_i \leq v_i$. For a particular v_i, direction of the relation is chosen so that the condition covers the argumented example. To choose the best v_i, candidate thresholds are built between the values present for the attribute in the learning set (which discriminate examples from different classes). For each candidate threshold, an appropriate condition is built and temporarily added to the rule. If it does not violate any negative arguments, a new candidate rule is evaluated using an evaluation measure and the best condition is chosen.

Algorithm 7.2 *Induction of the best rule from an argumented example*

```
Procedure ABFindBestRule (Not_covered_examples ES,
                          Argumented_Example AE, Class Y)

 1.LET BEST_RULE be an empty rule.
 2.foreach Positive_argument Arg for AE
 3.    Let RULE be a conjunction of reasons from Arg.
 4.    Let S be a set of objects in ES covered by RULE
 5.    while (S contains negative examples)     #not from Y
 6.        Let BEST_CONDITION be an empty elementary condition
 7.        foreach attribute A in Attributes
 8.            Let NEW_CONDITION be ABFindBestCondition(A,S,AE,RULE)
 9.            if (NEW_CONDITION is better then BEST_CONDITION)
10.               BEST_CONDITION = NEW_CONDITION
11.        end foreach
12.        Add BEST_CONDITION to RULE
13.        Update S
14.    end while
15.    Remove redundant conditions from RULE
16.    if (RULE is better than BEST_RULE)
17.        BEST_RULE = RULE
18.end foreach
```

Rule Evaluation Measure

The arguments should lead to the induction of general rules covering many examples from the learning set, not only the specific example that was argumented. As Entropy or Laplace measure originally used in MODLEM [120] resulted in too specific argumented rules, we looked for other evaluation measure which would be computationally simpler than the extreme value correction technique [29]. We chose a *Weighted Information Gain* (WIG) inspired by Quinlan's proposal from FOIL algorithm, which according to [38] favors more general rules. Its definition was given in Section 4.1. In our earlier study [98] we carried out several experiments considering other evaluation measures, which showed that this measure was the best in generating more general rules.

7.4 Classification Strategy

A proper classification strategy is an important issue in argument-based learning with unordered set of rules. Let us observe that the argumented rules are usually built for more difficult or specific examples (we will discuss it in Section 7.5), so they might be more specific and supported by a smaller number of learning examples. Standard classification strategies, such as Grzymala strategy used in pure MODLEM, voting with m-estimate or Laplace accuracy (their definitions were given in Section 4.3), may discriminate the rules covering a small number of examples (see discussion in Section 4.4). As a result, they may underestimate the argumented rules in a conflict set when classifying a new object, which will be outvoted by stronger and more general non-argumented rules. The argumented rules should receive more attention as they were partly supervised by the expert and refer to special decision cases.

This problem was already noticed by Bratko *et al* in [94] and they proposed to solve it by choosing a single rule according to a new quality measure, which was estimated by their own method of extreme value correction [29]. However, calculating its parameters is quite sophisticated. We looked for simpler methods that would increase the role of argumented rules, but would still have a good intuitive meaning and could be consistent with the strategies already applied in MODLEM. As a result, we have decided to follow the inspirations coming from some earlier works on adapting Grzymala's strategy to class imbalance, which were based on increasing artificially the rule support for minority class, see [47, 48]. Its experimental evaluation showed that such modification improved the recognition of the minority class, see e.g [48]. In this thesis, we have not considered the classification strategies similar to the nearest rules used in BRACID, but we also focus on reducing the effect of outvoting the minority class rules by the too strong majority class rules. Therefore, in [98] we proposed for ABMODLEM to use the *Average Strength Strategy*, which aims to balance the influence of argumented and non-argumented rules while classifying new objects. In this strategy, for each class of rules in the conflict set the *support* is calculated according to the formula:

$$SUP(Y_k) = \sum_{R \in R_k} sup(R) \times MA(R)$$

where $SUP(Y_k)$ is the total support of decision class Y_k; $sup(R)$ is a support of rule R (number of examples covered by it) and R_k is a set of rules for class Y_k being in the conflict set. This formula is calculated for each class and the new object is assigned to the winner – the class with its greatest value. With MA factor equal to 1, this formula would be equivalent to Grzymala strategy [46]. In the *Average Strength Strategy*, the support of each *argumented rule* is multiplied by a *strength* factor MA which is defined as in Def. 7.3:

Definition 7.3

$$MA(R_{arg}) = \frac{\bar{R}_n}{\bar{R}_{arg}}$$

$$\bar{R}_n = \frac{\sum_{R \in R_n} sup(R)}{|R_n|}$$

$$\bar{R}_{arg} = \frac{\sum_{R \in R_{arg}} sup(R)}{|R_{arg}|}$$

R_{arg} is a set of rules induced from argumented examples only; R_n stands for a set of rules induced from non-argumented examples; $sup(R)$ is a support of rule R and $|.|$ is a cardinality of a set. If rule R is induced from non-argumented examples, then its $MA(R) = 1$. In this way, the importance of weaker rules induced from argumented examples is amplified and they can contribute more to a final classification decision.

We also tested other simpler strategies, including an Arbitrary Strategy, which assumed even higher level of credibility for the rules induced from argumented examples. In this strategy, if any argumented rule matched the classified object, the classification was performed according to the voting of solely argumented rules in the conflict set. However, this strategy performed worse than the above described proposal, showing that strong, confident rules induced without the help of the expert should also have a high level of credibility.

7.5 Identification of Examples for Argumentation

Selecting appropriate examples to be argumented by an expert is a crucial issue for the success of argument based rule induction, both in the class imbalance context and in the standard setting (with balanced classes). "Easy" examples, which represent a part of the concept definition supported by many learning examples, will probably be a seed for strong rules correctly built by an induction algorithm itself, and thus do not need to be argumented. One should rather select difficult examples corresponding to more difficult regions of the concept, such as regions under-represented in the learning dataset (common in imbalanced data). Such examples can lead to unintuitive rules (e.g. because they were built using weaker confidence estimates) which may decrease the accuracy of a rule classifier. The open problem is, how to select such examples from a dataset.

The simplest solution is to leave a choice of these examples to an expert who knows very well the domain of a problem at hand. Such a co-operation was presented in some real case studies with ABCN2 [94, 96, 146]. Let us recall that giving arguments to all the learning examples is not feasible in practice because it requires definitely too much effort for humans [94], so an expert has to focus on the limited number of examples. For problems described with a relatively small number of examples, an expert could have enough knowledge about all of them and should determine the necessary examples. However, such an approach is not feasible for larger problems. Demanding from the expert not only to explain some examples, but also to scan through the whole learning set to decide which examples should be chosen is rather non-realistic. What is more important, an example that is perceived by an expert as difficult may be easy for the classifier and conversely. Therefore, there is a need for a more automatic solution suggesting the expert which examples are possible candidates for argumentation.

Following the above motivations we think (similarly to Bratko [94]) that the expert should turn his attention to these examples which are frequently misclassified when used as testing ones in a validation technique for a rule induction. Such understanding of difficult examples has also occurred in the earlier research on instance based algorithms or active learning.

7.5.1 An Iterative Approach to Finding Misclassified Examples

Bratko *et al* introduced in [94] a simple method of automatic selection of examples (without considering the class imbalance problem). Its schema is given below.

1. Learn a rule set from plain examples without arguments.
2. Find the most critical example and present it to the expert. If a critical example cannot be found, stop the procedure.
3. Expert explains the example; the explanation is encoded in arguments and attached to the learning example.
4. Learn rules with ABML from the learning set extended with a new argumented example.
5. Return to step 2.

Before the induction process, a standard CN2 algorithm is used within the k-fold cross validation procedure repeated n times (in this way each example is evaluated n times). Test examples are then evaluated with respect to a chosen measure – according to [94] it is the number of misclassifications of the example. The example that was misclassified in most of the cases is chosen as the example that needs to be argumented. If there are several such examples, then the algorithm picks one at random. Then, ABCN2 algorithm is used to induce a new set of rules. The procedure is repeated until no critical examples are found or a satisfactory value of the classification accuracy is achieved. We see the following drawbacks of this approach:

(i) For large datasets, it is computationally costly and requires a time-consuming co-operation with an expert, who has to wait after each argument for the classifier to re-learn.

(ii) We carried out some preliminary experiments with this method with $k = 10$ and $n = 10$. During the identification of the first difficult example, for many examples the number of misclassifications was equal to n (sometimes the number of such examples was at least 10% of the whole set). Picking one of these examples at random may not lead to a choice of the most relevant example, in particular if one is interested in imbalanced data (we will show it in Chapter 8).

Following this criticism, we introduce in Sections 7.5.2 and 7.5.3 two different one-phase approaches in which an expert gives argumentation for all the required examples at once.

7.5.2 One-phase Cross Validation Approach

This approach is a simple modification of Bratko *et al.* procedure described in the previous section. Our proposal also uses k-fold cross validation repeated n times and testing examples are evaluated according to a number of misclassifications of the example. However, instead of picking one example at a time, the whole set of examples with the maximal number of misclassifications is presented to the expert.

Although the co-operation with an expert is now reduced to one step, this approach "inherits" the drawback (ii) – for some difficult datasets, too many examples ranked with the same number of errors could be identified (we will show some numbers in detail in the experimental study). In this case we still have to co-operate with the expert on selecting a smaller number of examples (using his domain knowledge) or perform a random selection. How much to reduce the set of identified examples is an open question. For datasets used in the further experiments we established that

staying with a small percentage of the data size (less than 2% for datasets bigger than 1000) was sufficient. Additional experiments also showed that increasing it did not bring substantial improvements with respect to the classification abilities. We denote this method as *CV* (abbreviation from the term *cross validation*).

7.5.3 One-phase Disagreement Approach

As the *CV* approach could still identify too many equally misclassified examples, we looked for another solution that could reduce a number of identified examples and focus on the most critical ones. Our proposal is inspired by a method called *Query by Committee* used in the framework of *Active Learning*, see e.g. [85]. A committee is an *ensemble of classifiers* of the same type (in our case of MODLEM classifiers), where each classifier is trained on a different subset of a learning set (like in bagging).

The key issue in active learning from partially labelled examples is also selecting examples – however it concerns unlabelled examples and the task is to direct their smallest subset to an oracle with an ask for providing their class labels. In most active learning methods these examples are chosen among the most uncertain ones, for instance with respect to the uncertainty of decisions made by an initial classifier trained on a small subset of a learning set and then used to classify the rest of the unlabelled examples.

In our approach we adapt the idea of selecting critical examples based on the largest *disagreement* between the component classifiers in an ensemble as to the predicted label for the classified (testing) example. As a measure of disagreement we choose a *margin measure* proposed in [1]. It is defined as the difference between the number of votes in the committee for the most predicted class and that for the second predicted class. Examples with the smallest margins are considered as the most uncertain and difficult for classification – so the disagreement can serve as a measure for identification of examples to be argumented. To be more selective in choosing difficult examples, we use the generalized version of margins, which takes probability distributions of class predictions instead of votes (following inspirations from [85] and a good earlier experience when using it in another active learning system [122]).

Within our proposal, we make only one iteration of this idea (in original Query by Bagging several iterations are possible), using all the examples from the learning set. After all classifiers have learned by bootstrap sampling [14] from the appropriate subset of learning examples, a measure of disagreement between the classifiers is calculated for the rest of examples. To be more consistent with the *CV* approach, the whole process is repeated n times, each time a set of examples is differently distributed among classiffiers.

In our first approach (further called *DoC*, short for *Disagreement of Classifiers*), the examples most frequently identified as difficult (uncertain) in n iterations are chosen for argumentation. More precisely, in each iteration we sort the examples according to their margin measure, and mark as difficult the first 10% of a learning set. We have also considered to sum the margin measures from n iterations for each example instead of applying a *cutoff* at 10% of the dataset, as the latter approach looses some information and the choice of 10% may seem arbitrary. However, sorting the examples according to their sum of margin measures did not lead to a clear threshold after which the remaining examples should be considered uncertain. It was also not more effective in the preliminary experiments.

We will show in the next section that *DoC* method selects a substantially smaller number of examples than *CV* method, maintaining similar classification abilities. However, we noticed that sometimes all the selected examples came from the minority class. As a result, argumentation

was too much biased toward the minority class at the expense of the other classes, which could downgrade the overall performance.

Due to this observation we propose a modified solution, which could be more suitable for imbalanced data. To assure that examples from other classes are also argumented, we choose all the most frequently identified examples, and in case they come from the minority class only we add a few additional examples most frequently identified among the remaining classes. We propose to add such a number of examples, that the ratio of selected examples is inversely proportional to the proportion of classes in the learning set. For instance, if the class imbalance ratio is 1:9 between the minority and the rest of the classes, the selected examples should reflect the proportion 9:1. This approach will be called *DoC-b* (*balanced DoC*).

Chapter 8

ABMODLEM: Experimental Study

In this experimental study we want to evaluate the effect of argumentation on the recognition of classes, focusing our interest on the minority class in imbalanced datasets. First, we compare ABMODLEM with its basic, non-argumented origin MODLEM, to evaluate if argumentation together with other modifications (concerning evaluation measure and classification strategy) influences the structure of the rule set (number of rules, average length of the condition parts) and its classification abilities. Analogously to the experimental evaluation carried out for BRACID, as the evaluation measures we use the F-measure, G-mean and Sensitivity (see their definition in Section 2.2). For the purpose of calculating G-mean and F-measure, multi-class problems are transformed to the binary case by aggregating all the majority classes into a negative one.

The other aim of the experiment is to compare different methods of selecting the critical examples which should be explained by an expert. We want to analyse how the choice of these examples influences the recognition of particular classes – both majority and minority ones. We will also verify if the disagreement-based approaches (DoC and DoC-b) can reduce a number of argumented examples compared to the CV method, maintaining a comparable (or even better) classification performance. While DoC method can favour too much the minority class, we will also check if keeping the balance of selected examples between the classes (DoC-b) can help to maintain a sufficient accuracy in the majority class.

Finally, we carry out an experimental study evaluating the scalability of argument-based approaches, to verify if for large datasets this approach would not require too many argumented examples to bring any classification improvement.

8.1 Datasets and Argumentation

For the experiments, datasets have to be extended by the expert's annotations. Bratko *et al* in their papers [146, 96] presented the results of co-operation with real experts from law or medicine. As in our study we were unable to co-operate with such experts, we have decided to use imbalanced datasets from the UCI repository. Due to the considerable effort required to provide the reliable annotations, the number of datasets used in the experiments had to be limited. We chose the datasets with intuitive domains, for which we were able to provide the reliable argumentation on our own.

The following datasets from the UCI repository were chosen: ZOO – describing species of animals with descriptive attributes, German Credit – representing bank credit policy, Car – evaluation of the quality of cars and Cmc – representing the choice of contraceptive method of a woman based on her demographic and socio-economic characteristics. All these datasets contain numerical attributes (for which MODLEM or ABMODLEM are well suited) and are characterized by dif-

ferent imbalance ratios. Although Car and German Credit datasets have a lower imbalance ratio, according to our analysis in Chapter 3, most of the examples in these datasets are borderline, rare or outlying examples. As a result, they are difficult for the learning algorithms – see experiments in Section 6.3 comparing BRACID with standard rule classifiers. Let us recall that they are also often used in the related works on class imbalance (e.g. in [132]). The basic characteristics of the datasets are given in Table 8.1.

Dataset	No. of examples	No. of attributes (numerical)	Minority class [%]	Domain
ZOO	100	17(1)	reptile (5%)	type of animal
Car	1728	6(2)	good (4%)	car evaluation
Credit	1000	20(7)	bad (30%)	admission of credit
Cmc	1473	9(6)	l-term (22%)	use of contraception

Table 8.1: Characteristics of argumented datasets

To identify the examples for argumentation we use our three approaches described in Section 7.5 – *CV* (Cross Validation Approach), *DoC* (Disagreement of Classifiers) and *DoC-b* (balanced DoC). We do not compare our methods with the original interactive method used in ABCN2, because they are based on different philosophies. The interactive procedure is continued until satisfactory results are obtained, and in theory by adding more iterations, one can constantly improve the quality of a classifier. Therefore, it would be difficult to compare the results of the methods because one would have to determine in advance when the iterative procedure should be stopped.

Our argumentation is based on common or encyclopaedic knowledge. However, for some difficult examples we additionally induced rules by other rule induction algorithms (such as PART, C45rules) and analysed the syntax of condition parts for the strongest and most accurate patterns covering these examples.

8.2 Experimental Setup

To the best of our knowledge, the ABCN2 implementation is not publicly available for the moment of writing this thesis. Therefore, we conduct the experiments using only our ABMODLEM implementation (implemented in Java using classes from the WEKA platform). Estimations of all evaluation measures are carried out by means of stratified 10-fold cross-validation repeated 10 times. To verify if the differences on a single dataset for a given pair of classifiers are statistically significant, we perform a corrected, one-tailed paired t-test with $\alpha = 0.05$ on each dataset [28]. Argumentation for a given example is used only if the example belongs to a training part. If an argumented examples is located in the testing part, it is treated as a plain non-argumented example. This means that for a set of argumented examples, only 90% of it is used for learning in each fold on average. Minority class labels in Tables 8.2-8.6 are underlined. For two datasets, the next small class has comparable cardinality so we also mark these classes, to observe if their accuracy can be also improved by argumentation.

ZOO

Algorithm	Total.acc	Mammal	Bird	Reptile	Fish	Amph.	Insect	Inv.
MODLEM	0.89	1.0	1.0	0.0	1.0	0.25	0.98	0.80
ABM1	0.96	1.0	1.0	0.60	1.0	0.75	1.0	0.90
ABM2	0.97	1.0	1.0	0.60	1.0	0.75	1.0	1.0

Car

Algorithm	Total.acc	Unacc	Acc	Good	Vgood
MODLEM	0.91	1.0	0.79	0.35	0.56
ABM1	0.91	1.0	0.81	0.42	0.58
ABM2	0.95	0.99	0.95	0.50	0.65

Credit

Algorithm	Total.acc	Bad	Good
MODLEM	0.73	0.32	0.90
ABM1	0.73	0.37	0.89
ABM2	0.75	0.38	0.92

Cmc

Algorithm	Total.acc	L-term	Other
MODLEM	0.75	0.25	0.90
ABM1	0.74	0.28	0.88
ABM2	0.76	0.30	0.90

Table 8.2: Total classification accuracy and accuracies in particular classes. ABM1:ABMODLEM with Entropy evaluation measure and Grzymala classification strategy, ABM2:ABMODLEM with WIG measure and Average Strengths strategy. Minority classes are underlined.

Dataset	Arguments	No. of rules	Rule length
ZOO	no	14	1.31
	yes	9	2.19
Car	no	148	4.72
	yes	149	4.74
Credit	no	172	4.09
	yes	123	3.91
Cmc	no	258.97	7.34
	yes	240.23	5.72

Table 8.3: Average characteristics of induced rule sets

8.3 Evaluation of ABMODLEM Components

Table 8.2 presents a total classification accuracy as well as the local accuracies in particular classes for pure MODLEM, ABMODLEM used with standard Entropy rule evaluation measure and Grzymala's classification strategy (denoted as ABM1) and ABMODLEM used with WIG evaluation measure and our Average Strength classification strategy (denoted as ABM2). We skip the results of comparing other configurations (they were presented in [98]). All the results were obtained with the same set of arguments (selected using CV method). One can notice that both ABMODLEM versions improved the total classification accuracy and the recognition of particular classes compared to MODLEM with no argumentation. While majority classes were improved only slightly or their recognition stayed at the same level (only ABM1 decreased it a little for Credit and Cmc datasets), the improvements were always observed for the minority classes – e.g. from 0 to 60% and from 25 to 75% for the minority classes in ZOO. The differences were statistically significant according to the t-test – e.g. for MODLEM vs ABM1, p-value on sensitivity for was 0.003 for Cmc, 0.0002 for Credit and smaller for other datasets. ABM2 could further increase the accuracy, especially by raising the recognition of minority classes (except for ZOO where they stayed at the same level; on other datasets, p-values for ABM1 vs ABM2 on sensitivity were: 0.019 for Credit, 0.008 for Cmc, 1.6E-06 for Car). This configuration (WIG evaluation measure and Average Strengths classification strategy) will be used in all the subsequent experiments.

The best classification performance of ABM2, especially when the minority classes are concerned, could be mainly due to the classification strategy used. We analysed that the average support of non-argumented rules was much higher than the support of rules induced from arguments – the ratio ranged from 4 (in Credit) to 40 (in Car). Therefore, without amplifying the importance of argumented rules in the voting classification strategy, they do not have the chance to sufficiently impact the classification. This is particularly important for the minority classes, as argumentation is provided mostly for the difficult minority examples.

We also analysed the influence of argumentation on the structure of induced set of rules – the average length of each rule (measured in the number of conditions) and the total number of rules (see Table 8.3). ABMODLEM (using argumentation) generated a slightly smaller set of rules (except for Car data) than MODLEM (with no argumentation).

8.4 Evaluation of Identification Methods

In Table 8.4 we present the number of examples selected for argumentation by three one-phase methods: cross-validation method (CV), disagreement of classifiers method (DoC) and balanced disagreement of classifiers (DoC-b). We should remark that in the Cmc dataset, DoC method initially identified 38 examples which could be argumented – however when we introduced the arguments for a part of them, the remaining ones could be explained using the same argumentation (they represented the same reasoning pattern) – so we could stay with a smaller number. It is clearly visible that in comparison with CV, both DoC methods can substantially decrease the number of identified examples. Let us recall that in order to keep the balance of examples between the classes, a DoC-b method has to add some additional (majority) examples.

As the CV method sometimes identified too high a number of examples (more than 100 in case of Credit and Cmc), we decided to make this procedure semi-automatic and manually select only a subset of examples. We were choosing the examples for which we were able to provide a reliable argumentation, true for more than one example (to let the rules generalize over other, possibly also non-argumented examples). Moreover, among these examples we were trying to maintain the same proportion of classes as in the whole identified set. To estimate the correct number of examples,

ZOO

Method	Total.args	% of examples
CV	10	10
DoC	3	3
DoC-b	4	4

Car

Method	Total.args	% of examples
CV	50	4.3
DoC	13	1.1
DoC-b	16	1.3

Credit

Method	Total.args	% of examples
CV	127	12.7
DoC	11	1.1
DoC-b	16	1.6

Cmc

Method	Total.args	% of examples
CV	137	9.3
DoC	16	1
DoC-b	20	1.3

Table 8.4: A number of examples identified for argumentation by each method

we used a simple heuristics. We were starting from a small number of examples, argumenting them, and adding iteratively new examples until the G-mean measure did not further improve. For instance, for Credit dataset we increased the number of examples by two until 21 when the G-mean reached 59%, because argumenting more examples did not bring further improvement.

Tables 8.5 and 8.6 present the classification results of using ABMODLEM with different identification methods. Table 8.5 summarises the number of arguments used on average per fold (see Section 8.2 for explanation) and the classifier performance evaluated with respect to three measures: total accuracy, F-measure and G-mean calculated for the selected minority classes. Table 8.6 gives details of classification accuracy for each class. As can be observed, argumentation always improves the performance. The paired t-tests confirm that the differences are statistically significant – e.g. tests comparing DoC-b method with non-argumented MODLEM give the following p-values on accuracy: 5.1E-15 for ZOO, 8.3E-09 for Car, 1.0E-05 for Credit and 0.0004 for Cmc. Although ABMODLEM with *CV* argumentation usually achieves the best total accuracy, both *DoC* methods lead to quite comparable results by using only about half as many arguments. *DoC* method usually gets the best performance with respect to the minority class (Table 8.6), which is sometimes reflected also in the values of F-measure and G-mean (Table 8.5, e.g. for Credit with p-values on test comparing CV and DoC equal to 1.3E-13 on G-mean and 3.2E-07 on F-measure).

ZOO

Method	Args per fold	Total.acc	F-measure	G-mean
no args	0	0.89	0	0
CV	5.4	0.97	0.66	0.77
DoC	2.7	0.95	0.65	0.94
DoC-b	3.6	0.96	0.69	0.96

Car

Method	Args per fold	Total.acc	F-measure	G-mean
no args	0	0.91	0.23	0.57
CV	29.7	0.95	0.47	0.69
DoC	11.7	0.91	0.37	0.79
DoC-b	14.4	0.92	0.39	0.79

Credit

Method	Args per fold	Total.acc	F-measure	G-mean
no args	0	0.73	0.41	0.53
CV	18.9	0.76	0.48	0.59
DoC	9.9	0.71	0.53	0.65
DoC-b	14.4	0.75	0.48	0.59

Cmc

Method	Args per fold	Total.acc	F-measure	G-mean
no args	0	0.75	0.31	0.47
CV	36.1	0.77	0.37	0.52
DoC	14.5	0.74	0.36	0.53
DoC-b	18.1	0.76	0.36	0.51

Table 8.5: Classification results depending on the identification method

However, *DoC* method often increases the recognition of minority class at the expense of the majority classes. One can also notice that the *DoC-b* method does not put such emphasis on one class only but it achieves more balanced results. While using this method, the recognition of minority class is also increased (although not as much as for *DoC*), but it maintains better recognition of majority classes. As a result, it gives a quite comparable performance to *CV* method (with respect to F-measure and G-mean, it is almost the same for Credit and Cmc, slightly worse on Car and slightly better on ZOO), while using definitely fewer argumented examples. We additionally inspected the distribution of argumented examples in classes for *CV* method, to see if it does not favor any of the classes. It was analogous to the distribution of *DoC-b* method – the minority class received much more arguments than the majority classes.

ZOO

Algorithm	Mammal	Bird	Reptile	Fish	Amph.	Insect	Inv.
no args	1	1	0	1	0.25	0.98	0.8
CV	1	1	0.6	1	0.75	1	1
DoC	1	1	0.94	1	0.75	0.98	0.64
DoC-b	1	1	0.96	1	0.75	0.97	0.73

Cas

Algorithm	Unacc	Acc	Good	Vgood
no args	1	0.79	0.35	0.56
CV	1	0.95	0.5	0.65
DoC	0.99	0.72	0.69	0.62
DoC-b	0.99	0.79	0.67	0.61

Credit

Algorithm	Bad	Good
no args	0.32	0.9
CV	0.38	0.92
DoC	0.55	0.78
DoC-b	0.38	0.91

Cmc

Algorithm	L-term	Other
no args	0.25	0.9
CV	0.3	0.9
DoC	0.33	0.85
DoC-b	0.29	0.9

Table 8.6: Local accuracy in each class depending on the identification method

8.5 Scalability of the Algorithm

Finally, in the last experiment we analysed the scalability of the argument-based rule learning approach, to verify if for large datasets this method would not require too many argumented examples to bring any improvement. In Table 8.7 we summarize the result of this experiment with using the DoC-b method for the largest dataset – Car. We were stepwise increasing the number of argumented examples, preserving in each set the proportion of argumented examples in classes according to the DoC-b method, and recorded the total classification accuracy and sensitivity on the minority class (*good*). The changes of sensitivity are presented in Table 8.7. The total accuracy stabilized at 0.92 after explaining 5 examples.

We can observe that increasing the number of argumented examples improves the classification performance, however after some steps (when about 16 examples are explained with arguments), the results become stable. We hypothesize that the method does not require to increase the number

Total.args	0	2	5	9	12	15	16	19	22
good	0.35	0.42	0.44	0.47	0.55	0.62	0.67	0.68	0.68

Table 8.7: Sensitivity on Car dataset depending on the number of argumented examples (DoC-b method)

of argumented examples linearly with the size of the dataset, but that a reasonable small number of arguments (in this case less than 1%) is sufficient to improve the evaluation measures.

8.6 Conclusions

In this Chapter we carried out a series of experiments with ABMODLEM on 4 specially argumented datasets from UCI repository. Let us summarize the main conclusions from these experiments:

1. Including argumentation in the learning process always led to the improvement of minority classes. Introducing a new evaluation measure (WIG) and a classification strategy (Average Strength Strategy) resulted in further improvement. For instance, in ZOO data the recognition of the *reptile* minority class increased from 0% to 96%; in Car for *good* class the improvement was equal to 15%; and to 6% for the *bad credits* class in German Credit. On the one hand, this could be because arguments are often created for difficult, misclassified examples that come from minority classes in the original set. On the other hand, the observed improvement in the minority classes did not decrease too much the recognition of the majority classes (no decrease for ZOO , Cmc and Credit, a small percentage only for Car).

2. As a result, using the argumentation always improved also the total accuracy (see Table 8.2) comparing to standard, non-argumented MODLEM. The differences were statistically significant according to the statistical test used on each dataset. This is a particularly interesting characteristics of using an argument-based approach for imbalanced data, as most solutions dedicated for class imbalance (including the preprocessing approaches and the BRACID algorithm) usually improve the recognition of minority classes at some expense of the majority classes, which often leads to the deterioration of the total accuracy.

3. In order not to limit the comparison to the MODLEM algorithm, we conducted additional experiments with (non-argumented) Ripper and C45 algorithm (WEKA implementations). For ABMODLEM the overall accuracy was still better (from 2 to 12% improvement for RIPPER and from 4 to 15% for C45, depending on the data) and it worked even better for imbalanced classes (improvement by up to 75%).

4. Another contribution involves selecting the examples suggested to the expert for explanation. We compared our methods of identification of examples (*CV*, *DoC* and *DoC-b*) and showed that both *DoC* methods select a substantially smaller set of examples than the *CV* method. The *CV* method is in fact semi-automatic, because in case of large datasets a number of identified examples is too high and they have to be further selected (either randomly or manually by an expert).

5. The new-proposed disagreement-based methods outperform the *CV* method, because with selecting a smaller number of examples, they can reach the same level of accuracy.

6. When the dataset is imbalanced, the *DoC* method favors the minority class and often does not select any examples from the majority classes. As a result, this method outperforms

other methods when the minority class recognition is concerned. However, argumenting only the examples from the minority class has a negative impact on the recognition of the majority class. As a result, this method is recommended when the minority class recognition is the most important. For balanced datasets, the *DoC* method should select the examples from both classes and avoid the problem of deterioration in one of the classes.

7. When the minority class is not the only priority and maintaining the recognition of the majority classes is also important, the *DoC-b* method helps to preserve a high accuracy in the majority classes by argumenting also some examples from them.

8. Compared to BRACID, this approach can be an interesting alternative when a more conservative approach is needed, which can bring (smaller) improvement on the minority class without deteriorating the majority classes. Also, BRACID seems to concentrate more on improving the recognition of the borderline examples, while ABMODLEM could be suited mostly for rare and outlier examples.

9. The argument-based rule learning approach is scalable, as with the growing size of the dataset, the number of examples which should be explained by an expert to give some improvement remains at a reasonable level – at the beginning the performance measures improve quickly with every argument, and after some time the results plateau, so there is no need to argument more examples. The ”saturation” can be reached faster when crucial examples are selected for argumentation at early stages.

10. Finally, concerning the structure of rulesets, ABMODLEM produced slightly smaller sets of rules than MODLEM. These conclusions are consistent with the results obtained by Bratko and Mozina with their ABCN2 algorithm.

In conclusion, our study shows that argument-based learning can be seen as a new valid method for improving rule classifiers learnt from imbalanced data. What is particularly important, with a proper control of choice of examples which should be argumented, argument based learning can improve the recognition of the minority class without deteriorating the recognition of majority classes – which is a limitation for most of the existing solutions dedicated for class imbalance. This approach is especially useful when the decisions taken have to be easily explicable and verifiable, as it uses a rule representation which is natural for humans and argumentation leads to induction of rules more consistent with the expert knowledge. It could be applicable also for large datasets, as explaining even a few crucial examples can bring a substantial improvement in both the classification performance and the interpretability of rules.

Let us also observe that argument-based framework and the automatic methods selecting difficult examples which should be consulted with an expert, might be also considered as a useful tool to distinguish between the outlying and noisy examples. This idea was also promoted in [41] where the authors suggested that such examples should be shown and analyzed by an expert; examples representing real noise should be eliminated from the training set, whereas outliers should be kept as rules representing valid exceptions.

Our identification method, e.g. DoC method based on the disagreement of classifiers, will most probably identify in the first place such lonely examples. We could easily extend the argumentation framework to let the expert decide wheter it is a rare, but important case which should be explained, or if it is a noisy observation which should be discarded (or corrected, e.g. relabeled, and added again to the dataset). For this purpose, we could adapt an extention proposed by Bratko in [95], where the expert could say ”I do not know” if he was unable to explain the example, because it was unintuitive or noisy. In this case the example was skipped and another example was presented

to the expert to obtain his argumentation. Distinguishing between noise and outliers in the class imbalance setting is an important issue, difficult to solve by automatic approaches – see discussion of approaches to handling noise in imbalanced datasets in Section 3.2.

Chapter 9

Summary and Conclusions

This thesis concerned the problem of learning rule classifiers from imbalanced data. As stated in Section 1.3, our goal was to analyse factors on data-level and on algorithmic-level which make learning rules from imbalanced data difficult; based on these observations, we wanted to introduce new rule learning techniques, which are more efficient than the existing solutions in terms of performance measures dedicated for class imbalance. In our opinion, this goal has been achieved. To support this claim, we present below the summary of the main contributions.

Study of data-level sources of difficulty

We have presented a literature review summarizing the current understanding of the imbalanced learning problem (Chapter 2). We share the claim of many researchers that the class imbalance ratio is not the main and only source of difficulty – the mutual position of examples has also a crucial impact on learning from imbalanced data. In Chapter 3, we have experimentally analysed these data factors, but from a different research perspective than the existing studies. First, contrary to the studies in, e.g., [132, 10], we have assumed that data factors such as overlapping of classes or noise are more influential than the size of the dataset or imbalance ratio. We have also focused on the data factors which in our opinion have not received enough attention in the literature concerning class imbalance – i.e. *rare* examples and *outlier* (minority) examples, which in our opinion should not be treated as noise. Second, we claimed that although the existing experimental studies already give interesting conclusions about the impact of different data factors on the classifiers and preprocessing methods, there is a lack of methods which would help to estimate the occurence of these factors in the real-world datasets. As a result, their conclusions cannot be easily applied in the real-world settings.

Therefore, in Chapter 3 we wanted to carry out the analysis of data factors in the real-world imbalanced datasets. First, we have used two visualisation methods (multi-dimensional scaling and t-SNE) to estimate the occurrence of these four types of examples in the real-world, multidimensional data. Then, to automatically identify the types of examples in the data, a new method has been introduced which is based on the analysis of the local neighbourhood of the learning examples, to identify four types of minority examples: *safe*, *borderline*, *rare* and *outlier* examples. We have decided to model the neighbourhood using k-nearest neighbour approach and HVDM distance measure, basing the choice of k and of HVDM on the literature studies. For comparison, we have also tested other types of neighbourhood.

Using these methods, we have analysed a collection of imbalanced datasets from the UCI repository and showed that the datasets are of different nature and that usually the distribution of examples is very complex. The comparison of several popular classifiers showed that they are sensitive to the types of examples in a different degree. Safe examples are easy to recognize by

most classifiers, however this type of examples in uncommon in the imbalanced datasets. Borderline examples are observed in many datasets and they can constitute more than a half of the majority class. They are more difficult for all the classifiers, but SVM and RBF work better than other compared classifiers on these examples. Rare and outlier examples, although not that numerous in the datasets, can represent 20-30% of the minority class. They are extremely difficult for all the classifiers but, contrary to borderline examples, PART, J48 and sometimes 1NN are less sensitive to these examples than SVM and RBF.

Preprocessing methods also behave differently depending on the type of examples. Undersampling methods (NCR) seem to be better for borderline examples, while for rare and outlier examples, it is better to use oversampling (SMOTE, SPIDER). According to our experiments, informed resampling works better than simple random resampling for most classifiers. Only for RBF we have observed that random oversampling works better compared to other classifiers.

Such a study, analysing several data factors at once in the real-world datasets, has not been carried out in the literature. The detailed conclusions from these experiments can be found in Chapter 3. The dataset analysis with our labelling method can be used to:

- point out the most promising directions for the development of methods dedicated for class imbalance,
- analyse the area of competence of the existing and newly-proposed learning methods,
- suggest an appropriate learning method for a given problem at hand.

Study of algorithmic-level sources of difficulty

We have conducted a comprehensive analytical study of the techniques used in standard rule learning algorithms. We have discussed such techniques as sequential covering induction technique, measures used to evaluate rules and classification strategies and showed how they may be implicitely biased towards the majority classes. We have also carried out a broad review of literature proposals modifying the rule learning algorithms for the class imbalance setting. To the best of our knowledge, such a broad review has not been presented yet in the literature.

Bottom-up induction of Rules And Cases for Imbalanced Data

Based on the theoretical analysis presented in Chapters 2-4, we have proposed a new rule learning algorithm, BRACID (Chapter 5). Compared to the existing methods dedicated for class imbalance, BRACID addresses more comprehensively the problems on both data level and algorithmic level. The existing methods usually address only a single (or few) of these problems. The main characteristics of BRACID include using a hybrid representation of rules and instances, changing the greedy sequential covering, top-down induction technique, using a less biased evaluation measures, applying a more local classification strategy and different processing of examples depending on their type. BRACID has been compared to several standard rule learning algorithms as well as to the solutions dedicated for class imbalance in Chapter 6. The results showed that BRACID can significantly improve the recognition of the minority class compared to other approaches. BRACID works well with all types of minority examples distinguished in Chapter 3, but it is especially well suited for the borderline (and partly rare) examples.

Using expert argumentation for learning rules from imbalanced data

The experiments carried out in Chapters 3 and 7 showed that the safe examples are easy to learn by most standard learning methods. The recognition of borderline examples can be improved if

the methods dedicated for class imbalance are used, e.g. by our BRACID algorithm. However, rare and outlier minority examples are extremely difficult for most of the automatic learning methods. Even if some methods (such as SMOTE) can sometimes improve the recognition of these examples, it comes at a cost of deteriorating too much the majority class recognition. In Chapter 7, we hypothesized that identifying such difficult examples and explaining why they are assigned to a particular class could be done in co-operation with the domain expert. To the best of our knowledge, using the expert knowledge in the context of class imbalance has not been considered yet in the literature. We adapted the paradigm of argument-based learning (originally formulated by Bratko and Mozina in [94]) for the imbalanced domain, in which an expert can give additional arguments for the selected difficult examples, explaining the decision taken for it. Our important contribution is the introduction of a new method, which automatically selects a small number of most critical examples which should be explained by an expert. The experimental evaluation of our proposal, called ABMODLEM, showed that it can improve the recognition of the minority class (Chapter 8). What is more important, this improvement does not come at a cost of degrading the majority class recognition, which is a problem for most automatic approaches. With a proper selection of examples which should be explained by an expert, a trade-off between the recognition of the minority and majority classes can be controlled, according to the preferences of the user.

The results presented in this thesis have either been already published or are currently under review in the journals from the field of machine learning and artificial intelligence. Their list is included in Appendix B.

Additionally, as a part of this thesis, several algorithms have been designed and implemented. They include the implementation of BRACID and ABMODLEM within the WEKA framework; the labelling method has been implemented as a WEKA filtering method. This software will be made available as the open source projects.

Finally, there are still some interesting questions we would like to study in the future:

Using BRACID when multiple minority classes are present. ABMODLEM can be used when there are several minority classes in the dataset. In this case, arguments for these classes have to be introduced. In the current version of BRACID, only one minority class has to be distinguished, on which the learning algorithm will concentrate. It would be interesting to introduce a possibility to focus on several minority classes at once. Multiple minority classes are observed, e.g., in medical problems, where apart from a single minority class corresponding to a rare illness, there may be another class, represented by slightly more examples, but equally important. Therefore, it should not be considered as a majority class.

Comparison of the rulesets induced by BRACID and PART+SMOTE+ENN. In the experiments carried out in Chapter 6, the most competitive method to BRACID was PART used with SMOTE+ENN. We hypothesise that the additional argument in favor of BRACID is that its rules are based solely on the actual examples from the learning set, while rules induced with the help of SMOTE may depend mostly on the artificial examples. We plan to verify this hypothesis experimentally and evaluate the structure of rules for these two classifiers in terms of the examples used to build the particular rules. In general, an analysis of how much the results of a classifier depend on the artificial examples introduced by SMOTE, is an iteresting research topic.

Using ABMODLEM to distinguish true noise from outliers. As already discussed in Section 8.6, argument-based framework used with our method for identification of critical examples might be be used to let the expert distinguish if a difficult example is a true noise, which should be removed from the training set, or if it is an outlier important for the definition of a class, which should be kept. Distinguishing noise from outliers in the class imbalance setting in an important research challenge and to the best of our knowledge, no satisfactory methods have been proposed to address this issue.

Studying the impact of the classification strategies on rule classifiers. We have discussed in Section 4.4 that classification strategies are often biased towards the majority classes. The experimental evaluation of the components both in BRACID and in ABMODLEM showed that changing the classification strategy was responsible for an important improvement in the performance of the final classifier. We could even say that not adjusting the classification strategy undermined the improvements introduced in the previous phases of constructing a classifier. Therefore, we plan to carry out a more comprehensive study comparing the strategies in the class imbalance setting, and possibly propose new, more sophisticated strategies which could be used in learning rules from imbalanced datasets.

Bibliography

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proc. of 15th Int Conf. on Machine Learning*, pages 1–10, 2004.

[2] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML'04)*, pages 39–50, 2004.

[3] A. An. Learning classification rules from data. *Computers and Mathematics with Applications*, 45:737–748, 2003.

[4] A. An and N. Cercone. ELEM2: A learning system for more accurate classifications. In *Proceedings of the 12th Conference on Advances in Artificial Intelligence*, pages 426–441, 1998.

[5] A. An and N. Cercone. Rule quality measures for rule induction systems: Description and evaluation. *Computational Intelligence*, 17(3):409–424, 2001.

[6] A. An, N. Cercone, and X. Huang. A case study for learning from imbalanced data sets. In *Proceedings of the 14th Canadian Conference on Artificial Intelligence (AI2001)*, pages 1–15, 2001.

[7] D. Anyfantis, M. Karagiannopoulos, S. B. Kotsiantis, and P. E. Pintelas. Robustness of learning techniques in handling class noise in imbalanced datasets. In *Proc. of AIAI'07*, pages 21–28, 2007.

[8] R. Barandela, J.S. Sanchez, V. Garcia, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, (36):849–851, 2003.

[9] G. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[10] G. Batista, D. Silva, R. Giusti, V. Souza, and R. Prati. An experimental design to evaluate class imbalance treatment methods. To appear in the proceedings of ICMLA'12, Workshop on Class Imbalances, 2012.

[11] G. Batista and D. F. Silva. How k-nearest neighbor parameters affect its performance. In *Proc. of Argentine Symposium on Artificial Intelligence, Mar del Plata, Argentina*, pages 1–12, 2009.

[12] Ch. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[13] J. Blaszczynski, M. Deckert, J. Stefanowski, and Sz. Wilk. Integrating selective preprocessing of imbalanced data with ivotes ensemble. In *Proceedings of the RSCTC'10 Conference*, volume 6086 of *LNAI*, pages 148–157. Springer Verlag, 2010.

[14] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[15] L. Breiman. Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1-2):85–103, 1999.

[16] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

[17] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-level-SMOTE: Safe level synthetic over-sampling technique for handling the class imbalanced problem. In *Proc. PAKDD 2009*, volume 5476 of *Springer LNAI*, pages 475–482, 2009.

[18] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 164–168. AAAI Press, 1998.

[19] N. V. Chawla. Data mining for imbalanced datasets: An overview. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 853–867. Springer, 2005.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Int. Res. (JAIR)*, 16:321–357, 2002.

[21] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *Proc. of Principles Knowl. Discov. Databases*, pages 107–119, 2003.

[22] D. A. Cieslak and N. V. Chawla. Learning decision trees for unbalanced data. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, ECML PKDD'08, pages 241–256, Berlin, Heidelberg, 2008. Springer-Verlag.

[23] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[24] W. W. Cohen. Efficient pruning methods for separate-and-conquer rule learning systems. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 988–994, 1993.

[25] W. W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123, 1995.

[26] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.

[27] W. C. de Leeuw and R. van Liere. Visualization of multidimensional data using structure preserving projection methods. In *Data Visualization: The State of the Art'03*, pages 213–224, 2003.

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

[29] J. Demsar, M. Mozina, I. Bratko, and J. Zabkar. Why is rule learning optimistic and how to correct it. In *Proc. of 17th European Conference on Machine Learning (ECML 2006)*, Berlin, Springer-Verlag, pages 330–340, 2006.

[30] M. Denil and T. P. Trappenberg. A characterization of the combined effects of overlap and imbalance on the SVM classifier. *CoRR*, pages 1–1, 2011.

[31] P. Domingos. The RISE system: Conquering without separating. In *Proceedings of 6th IEEE International Conference on Tools with Artificial Intelligence*, pages 704–707. IEEE Computer Society Press, 1994.

[32] P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168, 1996.

[33] S. Dzeroski, B. Cestnik, and I. Petrovski. Using the m-estimate in rule induction. *Journal of Computing and Information Technology*, pages 37–46, 1993.

[34] T. Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.

[35] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Min. Knowl. Discov.*, 1(3):291–316, 1997.

[36] P. A. Flach and N. Lavrac. Rule induction. In M. Berthold and D.J. Hand, editors, *Intelligent Data Analysis: An Introduction*, pages 229–267. Springer, 2003.

[37] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In *Proceedings of the 15th Int. Conf. on Machine Learning*, pages 144–151, 1998.

[38] J. Furnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

[39] J. Furnkranz and G. Widmer. Incremental reduced error pruning. In *Proceedings of the Int. Conf. on Machine Learning*, pages 70–77, 1994.

[40] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(4):463–484, 2012.

[41] D. Gamberger, R. Boskovic, N. Lavrac, and C. Groselj. Experiments with noise filtering in a medical domain. In *Proc. of 16 th ICML*, pages 143–151. Morgan Kaufmann, 1999.

[42] S. Garcia, A. Fernandez, and F. Herrera. Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Applied Soft Computing*, 9:1304–1314, 2009.

[43] V. Garcia, R. A. Mollineda, and J. S. Sanchez. On the k-nn performance in a challenging scenario of imbalance and overlapping. *Pattern Anal. Appl.*, 11(3-4):269–280, 2008.

[44] V. Garcia, J. Sanchez, and R. Mollineda. An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In *Proc. of CIARP'07*, volume 4756 of *LNCS*, pages 397–406, 2007.

[45] J. W. Grzymala-Busse. LERS – a system for learning from examples based on rough sets. In R. Slowinski, editor, *Intelligent Decision Support*, pages 3–18. Kluwer Academic Publishers, 1992.

[46] J. W. Grzymala-Busse. Managing uncertainty in machine learning from examples. In *Proceedings of the 3rd International Symposium in Intelligent Systems*, pages 70–84. IPI PAN Press, 1994.

[47] J. W. Grzymala-Busse, L. K. Goodwin, W. J. Grzymala-Busse, and X. Zheng. An approach to imbalanced data sets based on changing rule strength. In *Proceedings of Learning from Imbalanced Data Sets, AAAI Workshop at the 17th Conference on AI*, pages 69–74, 2000.

[48] J.W. Grzymala-Busse, J. Stefanowski, and Sz. Wilk. A comparison of two approaches to data mining from imbalanced data. In *Proceedings of the KES 2004 – 8th Int. Conf. on Knowledge-based Intelligent Information & Engineering Systems*, volume 3213 of *LNCS*, pages 757–763. Springer, 2004.

[49] H. Guo and H. L. Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM SIGKDD Explor. Newsl.*, 6(1):30–39, 2004.

[50] H. Han, W. Wang, and B. Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proc. of ICIC*, volume 3644 of *Springer LNCS*, pages 878–887, 2005.

[51] P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, (14):1968, 515–516.

[52] H. He, Y. Bai, E. A. Garcia, and S. Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proc. Int. J. Conf. Neural Networks*, pages 1322–1328, 2008.

[53] H. He and E. Garcia. Learning from imbalanced data. *IEEE Transactions on Data and Knowledge Engineering*, 9(21):1263–1284, 2009.

[54] S. Hido and H. Kashima. Roughly balanced bagging for imbalanced data. In *Proc. of 8th SIAM Int. Conf. Data Mining*, pages 143–152, 2008.

[55] R. C. Holte, L. E. Acker, and B. W. Porter. Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 813–818, 1989.

[56] N. Japkowicz. Class imbalance: Are we focusing on the right issue? In *Proceedings of 2nd Workshop on Learning from Imbalanced Data Sets (ICML)*, pages 17–23, 2003.

[57] N. Japkowicz, C. Myers, and M. A. Gluck. A novelty detection approach to classification. In *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, pages 518–523, 1995.

[58] N. Japkowicz and M Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[59] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6 (5):429–450, 2002.

[60] T. Jo and N. Japkowicz. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40–49, 2004.

[61] M. V. Joshi, R. C. Agarwal, and V. Kumar. Mining needles in a haystack: classifying rare classes via two-phase rule induction. In *Proceedings of the SIGMOD KDD Conference on Management of Data*, 2001.

[62] M. V. Joshi, V. Kumar, and R. C. Agarwal. Evaluating boosting algorithms to classify rare cases: comparison and improvements. In *Proc. of First IEEE International Conference on Data Mining*, 2001.

[63] T. M. Khoshgoftaar and J. Van Hulse. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Enginnering*, 68:1513–1542, 2009.

[64] W. Klosgen. Domain knowledge to support the discovery process. In *Handbook of Data Mining and Knowledge Discovery*, pages 457–461. Oxford University Press, 2002.

[65] W. Klosgen and J. M. Zytkow, editors. *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, 2002.

[66] I. Kononenko and M. Kukar. *Machine Learning and Data Mining.* Horwood Pub., 2007.

[67] M. Kubat, R. Holte, and S. Matwin. Learning when negative examples abound. In *Proceedings of the International Conference on Machine Learning: ECML-97*, number 1224 in Lecture Notes in Artificial Intelligence, Springer, pages 146–153, 1997.

[68] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215, 1998.

[69] M. Kubat and S. Matwin. Addresing the curse of imbalanced training sets: one-side selection. In *Proceedings of the 14th Int. Conf. on Machine Learning*, pages 179–186, 1997.

[70] P. Langley and H. A. Simon. Fielded applications of machine learning. In R. S. Michalski, I. Bratko, and M. Kubat, editors, *Machine learning and data mining*, pages 113–129. John Wiley & Sons, 1998.

[71] J. Laurikkala. Improving identification of difficult small classes by balancing class distribution. Technical report, University of Tampere, 2001.

[72] J. Laurikkala and M. Juhola. Nearest neighbour classification with heterogeneous proximity functions. In A. Hasman, B. Blobel, J. Dudeck, R. Engelbrecht, G. Gell, and H.-U. Prokosch, editors, *Studies in health technology and informatics: medical infobahn for Europe*, pages 753–757. IOS Press, Amsterdam, 2000.

[73] N. Lavrac and S. Dzeroski. *Inductive logic programming – techniques and applications.* Ellis Horwood series in artificial intelligence. Ellis Horwood, 1994.

[74] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction.* Springer, New York, NY, USA, 2007.

[75] C. Ling and C. Li. Data mining for direct marketing problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98).* AAAI Press, 1998.

[76] W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla. A robust decision tree algorithm for imbalanced data sets. In *Proc. of SIAM International Conference on Data Mining, 2010*, pages 766–777, 2010.

[77] X. Y. Liu and Z. H. Zhou. The influence of class imbalance on cost-sensitive learning: an empirical study. In *Proc. of 6th Intl. Conf. on Data Mining*, pages 970–974, 2006.

[78] Y. Liu, B. Feng, and G. Bai. Compact rule learner on weighted fuzzy approximation spaces for class imbalanced and hybrid data. In *Proceedings of the 6th International Conference on Rough Sets and Current Trends in Computing*, volume 5306 of *LNAI*, pages 262–271. Springer-Verlag, 2008.

[79] O. Luaces. Inflating examples to obtain rules. *International Journal of Intelligent Systems*, 18:1113–1143, 2003.

[80] J. Lumijarvi, J. Laurikkala, and M. Juhola. A comparison of different heterogeneous proximity functions and Euclidean distance. *Stud Health Technol Inform*, 107(Pt 2):1362–6, 2004.

[81] T. Maciejewski and J. Stefanowski. Local neighbourhood extension of SMOTE for mining imbalanced data. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pages 104–111. IEEE Press, 2011.

[82] M. A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Workshop on Learning from Imbalaned Data Sets (ICML03)*, 2003.

[83] S. Marcellin, D. A. Zighed, and G. Ritschard. Evaluating decision trees grown with asymmetric entropies. In *Proceedings of the 17th international conference on Foundations of intelligent systems*, ISMIS'08, pages 58–67, Berlin, Heidelberg, 2008. Springer-Verlag.

[84] B. McCane and M. Albert. Distance functions for categorical and mixed variables. *Pattern Recogn. Lett.*, 29:986–993, 2008.

[85] P. Melville and R. J. Mooney. Diverse ensembles for active learning. In *Proc. of 21st Int Conf. on Machine Learning*, pages 584–591, 2004.

[86] W. Michalowski, Sz. Wilk, K. Farion, J. Pike, S. Rubin, and R. Slowinski. Development of a decision algorithm to support emergency triage of scrotal pain and its implementation in the met system. *European Journal of Operational Research*, 43:287–301, 2005.

[87] R. S. Michalski, I. Bratko, and M. Kubat. *Machine Learning and Data Mining: Methods and Applications*. John Wiley and Sons, 1998.

[88] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application in three medical domains. In *Proceedings of 5th National Conference on AI*, pages 619–625. AAAI-Press, 1986.

[89] C. R. Milar, G. Batista, and A. Carvalho. A hybrid approach to learn with imbalanced classes using evolutionary algorithms. *Logic Journal of the IGPL*, 19(2):293–303, 2011.

[90] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1997.

[91] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In *Proc. of International Conference on Machine Learning (ICML)*, pages 268–277, 1999.

[92] M. Mozina. Argument based machine learning. *Ph.D dissertation, University of Ljubljana*, 2009.

[93] M. Mozina and I. Bratko. Argumentation and machine learning. *Research Report: Deliverable 2.1 for the ASPIC project*, 2004.

[94] M. Mozina, I. Bratko, and J. Zabkar. Argument based machine learning. *Aritificial Intelligence Journal*, 171:922–937.

[95] M. Mozina, M. Guid, J. Krivec, A. Sadikov, and I. Bratko. Fighting knowledge acquisition bottleneck with argument based machine learning. In *Proc. of Int. Conference on ECAI 2008*, pages 234–238. IOS Press, 2008.

[96] M. Mozina, J. Zabkar, T. Bench-Capon, and I. Bratko. Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13(1):53–57, 2005.

[97] I. Nabney and P. Jenkins. Rule induction in finance and marketing. *Expert Systems*, 10 (3):173–177, 1993.

[98] K. Napierala and J. Stefanowski. Argument based generalization of MODLEM rule induction algorithm. In *Proc. of 7th Int. Conf. Rough Sets and Current Trends in Computing*, volume 6086 of *Springer LNAI*, pages 138–147, 2010.

[99] K. Napierala and J. Stefanowski. BRACID: a comprehensive approach to learning rules from imbalanced data. *Journal of Intelligent Information Systems*, 39(2):335–373, 2012.

[100] K. Napierala and J. Stefanowski. Identification of different types of minority class examples in imbalanced data. In *Proc. of HAIS, Salamanca, Spain*, volume 7209 of *Springer LNCS*, pages 139–150, 2012.

[101] K. Napierala and J. Stefanowski. Modifications of classification strategies in rule set based bagging for imbalanced data. In *Proc. of HAIS, Salamanca, Spain*, volume 7209 of *Springer LNCS*, pages 514–525, 2012.

[102] K. Napierala, J. Stefanowski, and Sz. Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *Proc. of 7th Int. Conf. Rough Sets and Current Trends in Computing*, volume 6086 of *Springer LNAI*, pages 158–167, 2010.

[103] C. H. Nguyen and T. B. Ho. An imbalanced data rule learner. In *Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD05)*, pages 617–624, 2005.

[104] T. Niblett. Constructing decision trees in noisy domains. In *Proceedings of EWSL*, pages 67–78, 1987.

[105] S. Ontanon and E. Plaza. Multiagent inductive learning: an argumentation-based approach. In *Proc. of ICML*, pages 839–846, 2010.

[106] A. Orriols-Puig, D. E. Goldberg, K. Sastry, and E. Bernado-Mansilla. Modeling XCS in class imbalances: population size and parameter settings. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO, pages 1838–1845. ACM, 2007.

[107] R. C. Prati, G. Batista, and M. C. Monard. Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Proceedings of MICAI'04*, pages 312–321, 2004.

[108] R. C. Prati, G. Batista, and M. C. Monard. Learning with class skews and small disjuncts. In *Proc. of SBIA'04*, pages 296–306, 2004.

[109] J.R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[110] B. Raskutti and A. Kowalczyk. Extreme re-balancing for SVMs: a case study. In *Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning*, 2003.

[111] P. Riddle, R. Segal, and O. Etzioni. Representation design and brute-force induction in a boeing manufacturing design. *Applied Artificial Intelligence*, (8):125–147, 1994.

[112] J. Saez, M. Luengo, J. Stefanowski, and F. Herrera. SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 2012.

[113] S. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:251–276, 1991.

[114] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, 40(1):185–197, 2010.

[115] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Commun. ACM*, pages 1213–1228, 1986.

[116] J. Stefanowski. Classification support based on the rough sets. *Foundations of Computing and Decision Sciences*, 18:371–380, 1993.

[117] J. Stefanowski. Using valued closeness relation in classification support of new objects. In T.Y. Lin and A.M. Wildberg, editors, *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*, pages 324–327. Simulation Council Inc., 1995.

[118] J. Stefanowski. Rough set based rule induction techniques for classification problems. In *Proceedings of 6th European Congress on Intelligent Techniques and Soft Computing*, volume 1, pages 109–113, 1998.

[119] J. Stefanowski. Algorithms of rule induction for knowledge discovery (in Polish). *Habilitation Thesis published as Series Rozprawy, Poznan University of Technology Press*, 361:18–21, 2001.

[120] J. Stefanowski. On combined classifiers, rule induction and rough sets. *Transactions on Rough Sets*, 6:329–350, 2007.

[121] J. Stefanowski. Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In S. Ramanna, L. C. Jain, and R. J. Howlett, editors, *Emerging Paradigms in Machine Learning*, volume 13 of *Smart Innovation, Systems and Technologies*, pages 277–306. Springer Berlin Heidelberg, 2013.

[122] J. Stefanowski and M. Pachocki. Comparing performance of committee based approaches to active learning. In *Recent Advances in Intelligent Information Systems*, pages 457–470. EXIT, 2009.

[123] J. Stefanowski and Sz. Wilk. Rough sets for handling imbalanced data: combining filtering and rule-based classifiers. *Fundamenta Informaticae*, 72:379–391, 2006.

[124] J. Stefanowski and Sz. Wilk. Improving rule based classifiers induced by modlem by selective pre-processing of imbalanced data. In *Proceedings of the RSKD Workshop at ECML/PKDD*, pages 54–65, 2007.

[125] J. Stefanowski and Sz. Wilk. Selective pre-processing of imbalanced data for improving classification performance. In *Proceedings of the 10th Int. Conf. DaWaK*, volume 5182 of *LNCS*, pages 283–292. Springer, 2008.

[126] J. Stefanowski and Sz. Wilk. Extending rule-based classifiers to improve recognition of imbalanced classes. In Z. Ras and A. Dardzinska, editors, *Advances in Data Management*, volume 223 of *Studies in Computational Intelligence*, pages 131–154. Springer Berlin/Heidelberg, 2009.

[127] P. N. Tan, M. Steinbach, and V. Kumar. Classification: Alternative techniques. In *Introduction to Data Mining*, pages 207–223. Pearson Addison Wesley, 2005.

[128] Y. Tang, Y. Zhang, N. V. Chawla, and S. Krasser. Svms modeling for highly imbalanced classification. *IEEE Trans. System Man and Cybernetics Part B*, 39(1):281–288, 2009.

[129] K. M. Ting. The problem of small disjuncts: its remedy in decision trees. In *Proceeding of the 10th Canadian Conference on Artificial Intelligence*, pages 91–97, 1994.

[130] I. Tomek. Two modifications of CNN. *IEEE Transactions on Systems Man and Communications*, (6):769–772, 1976.

[131] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[132] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th Int. Conf. on ML (ICML)*, pages 17–23, 2007.

[133] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. A novel noise filtering algorithm for imbalanced data. In *ICMLA'10*, pages 9–14, 2010.

[134] S. Verbaeten and A. van Assche. Ensemble methods for noise elimination in classification problems. In *Proc. of 4th International Workshop On Multiple Classifier Systems*, pages 317–325. Springer, 2003.

[135] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proc. of Intl. Joint Conf. on Artificial Intelligence*, pages 55–60, 1999.

[136] S. Visa and A. Ralescu. Issues in mining imbalanced data sets - a review paper. In *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, pages 67–73, 2005.

[137] B. Wang and N. Japkowicz. Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems*, 25(1):1–20, 2010.

[138] X. Wang, H. Shao, N. Japkowicz, S. Matwin, X. Liu, and B. Nguyen. Using SVM with adaptively asymmetric misclassification costs for mine-like objects detection. To appear in the proceedings of ICMLA'12, Workshop on Class Imbalances, 2012.

[139] G. M. Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6 (1):7–19, 2004.

[140] G. M. Weiss and H. Hirsh. Learning to predict rare events in event sequences. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 359–363. AAAI Press, 1998.

[141] G. M. Weiss and F. Provost. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, pages 315–354, 2003.

[142] C. G. Weng and J. Poon. A new evaluation measure for imbalanced datasets. In *Proceedings of AusDM'08*, pages 27–32, 2008.

[143] Sz. Wilk, R. Slowinski, W. Michalowski, and S. Greco. Supporting triage of children with abdominal pain in the emergency room. *European Journal of Operational Research*, 160:696–709, 2005.

[144] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Communications*, 3(2):408–421, 1972.

[145] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *J. Artif. Intell. Res. (JAIR)*, 6:1–34, 1997.

[146] J. Zabkar, M. Mozina, J. Videcnik, and I. Bratko. Argument based machine learning in a medical domain. In *Computational Models of Argument: Proc. of COMMA 2006, (Frontiers in Artificial Intelligence and Applications*, volume 144, pages 725–730. IOS Press, Amsterdam, 2006.

[147] J. Zhang. A method that combines inductive learning with exemplar-based learning. In *Proceedings of the Second IEEE International Conference on Tools for Artificial Intelligence*, pages 31–37. IEEE Computer Society Press, 1997.

[148] J. Zhang, E. Bloedorn, L. Rosen, and D. Venese. Learning rules from highly unbalanced data sets. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM '04)*, pages 571–574. IEEE Computer Society, 2004.

[149] X. Zhu, X. Wu, and Y. Yang. Error detection and impact-sensitive instance ranking in noisy datasets. In *Proceedings of the 19th National Conference on Artifical intelligence*, AAAI'04, 2004.

[150] J.M. Zytkow. Types and forms of knowledge (patterns): rules. In *Handbook of data mining and knowledge Discovery*, pages 51–54. Oxford University Press, Inc., 2002.

Appendix A – Supplementary Tables

This Appendix contains supplementary tables with experimental results for Chapters 3 and 6.

Table 1: Labelling of datasets – $k = 7$

Dataset	S[%]	B[%]	R [%]	O[%]
abdominal-pain	65.84	20.79	7.92	5.45
acl	70.00	27.50	0.00	2.50
new-thyroid	68.57	31.43	0.00	0.00
vehicle	75.88	24.12	0.00	0.00
car	26.09	60.87	13.04	0.00
scrotal-pain	52.54	33.90	10.17	3.39
ionosphere	46.83	31.75	11.11	10.32
credit-g	12.67	66.00	12.33	9.00
ecoli	31.43	60.00	0.00	8.57
hepatitis	34.38	37.50	12.50	15.63
haberman	9.88	53.09	28.40	8.64
breast-cancer	24.71	42.35	23.53	9.41
cmc	18.32	52.25	15.02	14.41
cleveland	0.00	42.86	20.00	37.14
glass	0.00	11.76	58.82	29.41
hsv	0.00	0.00	28.57	71.43
abalone	8.96	17.31	30.45	43.28
postoperative	0.00	50.00	33.33	16.67
solar-flare	0.00	32.56	30.23	37.21
transfusion	20.22	46.63	18.54	14.61
yeast	1.96	50.98	11.76	35.29

Table 2: G-mean [%] of compared classifiers

Dataset	1NN	3NN	J48	PART	RBF	SVM
abdominal-pain	79.8	82.6	78.1	78.6	82.6	79.9
acl	81.2	86.6	89.1	84.8	88.8	87.8
new-thyroid	97.3	93.9	94.3	95.3	99.1	94.3
vehicle	92.1	91.9	91.3	91.9	89.7	96.4
car	7.9	7.9	86.8	94.3	67.9	93.3
scrotal-pain	68.7	72.3	67.2	70.7	74.0	74.7
ionosphere	81.8	79.8	87.6	88.8	92.7	93.2
credit-g	63.7	58.1	59.1	60.2	61.0	65.2
ecoli	66.8	66.3	69.2	55.4	65.7	71.1
hepatitis	56.1	51.5	53.9	54.9	71.9	64.7
haberman	44.6	43.9	53.8	46.8	34.4	3.1
breast-cancer	56.1	47.3	53.1	52.9	56.7	59.0
cmc	53.8	53.0	56.9	54.3	32.2	20.0
cleveland	30.7	22.2	34.3	38.2	16.0	14.1
glass	36.2	20.0	36.2	40.7	29.8	0.0
hsv	0.0	0.0	0.0	2.8	1.3	0.0
abalone	43.2	38.8	53.9	41.9	32.2	1.4
postoperative	5.6	0.0	6.2	13.3	16.5	10.0
solar-flare	17.8	16.6	37.6	31.9	18.8	26.8
transfusion	50.2	53.9	59.9	60.2	54.4	8.6
yeast	58.3	43.8	49.7	42.0	27.1	0.0

Table 3: F-measure [%] of compared classifiers

Dataset	1NN	3NN	J48	PART	RBF	SVM
abdominal-pain	69.9	74.3	69.4	69.1	75.9	72.1
acl	75.8	83.5	85.1	79.3	85.1	84.0
new-thyroid	95.0	91.2	89.5	91.8	96.7	93.6
vehicle	87.3	87.7	86.8	87.5	82.0	93.9
car	5.4	5.4	71.6	89.5	56.0	87.3
scrotal-pain	58.5	64.9	56.9	60.6	66.4	66.1
ionosphere	79.0	77.1	84.9	86.4	90.1	92.1
credit-g	51.7	45.9	45.7	47.1	49.3	53.6
ecoli	49.6	53.1	57.3	45.0	54.3	60.4
hepatitis	46.6	41.9	42.9	45.2	61.0	53.3
haberman	29.9	30.2	39.9	34.9	23.6	2.0
breast-cancer	43.8	35.6	39.3	38.9	44.7	47.0
cmc	35.1	36.0	40.2	36.1	18.6	8.7
cleveland	18.4	15.4	20.9	22.5	12.1	10.1
glass	26.7	18.1	32.0	32.8	23.7	0.0
hsv	0.0	0.0	0.0	2.3	0.8	0.0
abalone	20.9	22.6	36.6	26.9	20.1	0.5
postoperative	4.3	0.0	4.5	11.0	12.1	7.7
solar-flare	11.3	11.5	27.6	17.7	13.2	18.9
transfusion	33.0	38.6	46.2	46.2	41.5	4.2
yeast	39.5	31.2	35.0	28.7	19.0	0.0

Table 4: Specificity [%] of compared classifiers

Dataset	1NN	3NN	J48	PART	RBF	SVM
abdominal-pain	83.7	87.3	87.9	85.5	91.2	89.5
acl	93.6	97.0	94.0	91.8	95.2	95.2
new-thyroid	98.9	98.9	97.3	98.1	98.7	99.9
vehicle	95.3	96.2	95.9	95.9	91.7	97.6
car	100.0	100.0	98.4	99.6	99.0	99.4
scrotal-pain	83.6	91.8	84.8	81.6	90.6	86.4
ionosphere	97.2	98.0	93.5	94.3	91.6	98.0
credit-g	81.3	85.9	75.7	76.5	85.9	82.1
ecoli	93.7	96.1	95.9	96.2	95.7	96.5
hepatitis	90.7	91.5	87.3	89.7	91.1	90.2
haberman	77.4	84.7	80.5	85.9	90.9	98.7
breast-cancer	82.4	90.1	76.7	72.1	84.3	81.0
cmc	77.8	84.4	83.8	79.5	96.0	96.6
cleveland	90.1	95.1	89.9	89.3	95.8	95.2
glass	94.2	98.4	97.3	96.9	96.6	100.0
hsv	90.4	98.5	92.6	90.0	91.7	99.1
abalone	93.5	97.6	96.9	98.5	99.4	100.0
postoperative	79.1	90.3	73.5	69.3	78.3	75.9
solar-flare	98.4	99.3	99.0	96.8	99.1	98.6
transfusion	81.3	87.0	89.1	88.0	92.7	99.7
yeast	98.0	99.0	98.7	98.6	99.2	100.0

Table 5: G-mean for PART used with the preprocessing methods [%]

Dataset	None	RO	NCR	SMOTE	SPIDER
abdominal-pain	78.6	79.3	81.7	79.0	80.8
acl	84.8	86.5	88.7	87.6	88.4
new-thyroid	95.3	93.5	90.7	95.5	93.7
vehicle	91.9	93.3	93.7	93.9	92.1
car	94.3	85.6	94.6	93.3	94.1
scrotal-pain	70.7	71.1	69.4	71.6	68.5
ionosphere	88.8	88.7	85.0	87.6	87.2
credit-g	60.2	60.5	64.7	62.4	63.7
ecoli	55.4	67.0	78.3	80.0	79.6
hepatitis	54.9	66.3	67.7	64.8	63.6
haberman	46.8	59.0	62.1	62.0	56.0
breast-cancer	52.9	53.6	57.3	53.7	56.0
cmc	54.3	58.8	61.9	58.5	60.4
cleveland	38.2	25.9	50.9	38.9	37.5
glass	40.7	38.5	62.0	52.1	47.5
hsv	2.8	10.3	8.4	15.1	10.2
abalone	41.9	59.4	53.8	67.7	65.6
postoperative	13.3	23.3	31.5	15.8	34.0
solar-flare	31.9	52.5	61.0	49.5	60.5
transfusion	60.2	63.1	62.5	57.9	57.6
yeast	42.0	53.6	47.2	64.1	54.4

Table 6: F-measure for PART used with the preprocessing methods [%]

Dataset	None	RO	NCR	SMOTE	SPIDER
abdominal-pain	69.1	69.3	70.9	69.5	69.9
acl	79.3	81.0	82.2	81.9	82.8
new-thyroid	91.8	90.1	85.5	91.8	88.9
vehicle	87.5	89.3	88.5	89.3	85.5
car	89.5	72.8	74.5	84.5	75.2
scrotal-pain	60.6	60.7	58.2	60.8	57.5
ionosphere	86.4	86.3	80.6	83.9	83.7
credit-g	47.1	47.5	53.2	49.6	51.2
ecoli	45.0	52.2	57.3	59.1	61.8
hepatitis	45.2	54.8	52.4	51.1	51.4
haberman	34.9	45.1	48.6	48.6	46.7
breast-cancer	38.9	39.9	47.3	39.9	43.6
cmc	36.1	39.6	42.5	39.0	40.9
cleveland	22.5	14.0	29.0	21.3	21.2
glass	32.8	31.9	45.7	38.2	35.7
hsv	2.3	7.5	5.7	9.0	6.3
abalone	26.9	36.1	35.2	36.9	34.8
postoperative	11.0	17.2	25.2	12.1	25.2
solar-flare	17.7	19.7	29.8	23.9	24.1
transfusion	46.2	46.4	46.7	44.7	44.3
yeast	28.7	30.2	29.0	32.2	28.4

Table 7: Sensitivity for RBF used with the preprocessing methods [%]

Dataset	None	RO	NCR	SMOTE	SPIDER
abdominal-pain	75.0	86.5	83.8	84.6	88.9
acl	84.0	90.5	84.5	88.5	88.0
new-thyroid	99.5	100.0	98.5	98.0	98.7
vehicle	88.0	92.4	93.9	93.9	92.4
car	49.6	68.9	89.5	57.8	70.4
scrotal-pain	62.5	78.7	72.4	68.7	78.1
ionosphere	94.2	97.6	94.0	97.2	97.0
credit-g	43.6	74.1	63.7	65.9	76.7
ecoli	54.7	82.0	81.8	77.3	79.0
hepatitis	60.7	69.8	65.5	69.7	67.5
haberman	18.3	57.8	52.1	73.4	70.8
breast-cancer	40.8	65.5	52.9	54.4	67.5
cmc	12.1	50.6	67.2	75.1	72.0
cleveland	9.5	32.2	34.5	28.0	35.3
glass	25.0	37.0	51.0	59.0	45.0
hsv	1.0	6.0	7.0	20.0	9.0
abalone	12.3	18.0	76.0	26.8	19.5
postoperative	13.7	40.0	23.0	16.7	24.7
solar-flare	10.2	31.2	64.6	41.5	54.3
transfusion	32.9	56.2	66.1	79.0	78.9
yeast	15.1	35.3	70.7	55.3	44.5

Table 8: Types of seed examples for maximally specific rules in BRACID. SB: safe or border examples; RO: rare or outlier examples. For datasets with no maximally specific rules, the corresponding cells are empty.

Dataset	SB [%]	RO [%]
ionosphere	0.00	100.00
cleveland		
ecoli	0.00	100.00
haberman		
solar-flare	0.00	100.00
transfusion	0.00	100.00
vehicle	0.00	100.00
yeast	0.00	100.00
abalone	0.00	100.00
abdominal-pain	0.00	100.00
car	93.75	6.25
cmc	11.54	88.46
credit-g	14.29	85.71
balance-scale	0.00	100.00
breast-w	0.00	100.00
pima	25.00	75.00
breast-cancer	0.00	100.00
postoperative	0.00	100.00
flags	100.00	0.00
hepatitis		
new-thyroid		
scrotal-pain	0.00	100.00

Appendix B

A list of author's publications related to the research carried out in this dissertation is presented in this Appendix. The papers are grouped according to the Chapters which they relate to.

Chapter 3:

- K. Napierala, J. Stefanowski, S. Wilk, 2010
  Learning from imbalanced data in presence of noisy and borderline examples.
  Proceedings of the Conf. on Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science, Springer Verlag 6086, 158-167

- K. Napierala, J,Stefanowski, 2012
  Identification of Different Types of Minority Class Examples in Imbalanced Data.
  Proc. of the 7th International Conf. HAIS 2012, Lecture Notes in Computer Science, Springer 7209, 139-150

- K. Napierala, J,Stefanowski, 2013
  Abstaining in Rule Set Bagging for Imbalanced Data.
  Submitted to Logic Journal of the IGPL

- K. Napierala, J,Stefanowski, 2012
  Modifications of Classification Strategies in Rule Set Based Bagging for Imbalanced Data.
  Proc. of the 7th International Conf. HAIS 2012, Lecture Notes in Computer Science, Springer 7209, 514-525

Chapters 5 and 6:

- Napierala, J. Stefanowski, 2011
  BRACID: a comprehensive approach to learning rules from imbalanced data.
  Journal of Intelligent Information Systems, Springer 2012, Volume 39, Number 2, Pages 335-373. DOI 10.1007/s10844-011-0193-0

Chapters 7 and 8:

- K. Napierala, J. Stefanowski, 2010
  Argument Based Generalization of MODLEM Rule Induction.

Proceedings of the Conf. on Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science, Springer Verlag, vol. 6086, 138-147

- K. Napierala, J,Stefanowski, 2013
  Addressing imbalanced data with argument based rule learning.
  Submitted to Expert Systems with Applications