J Intell Inf Syst (2012) 39:335–373
DOI 10.1007/s10844-011-0193-0

BRACID: a comprehensive approach to learning rules from imbalanced data

Krystyna Napierala · Jerzy Stefanowski

Received: 11 August 2011 / Revised: 19 November 2011 / Accepted: 7 December 2011 /
Published online: 30 December 2011

Abstract In this paper we consider induction of rule-based classifiers from imbalanced data, where one class (a minority class) is under-represented in comparison to the remaining majority classes. The minority class is usually of primary interest. However, most rule-based classifiers are biased towards the majority classes and they have difficulties with correct recognition of the minority class. In this paper we discuss sources of these difficulties related to data characteristics or to an algorithm itself. Among the problems related to the data distribution we focus on the role of small disjuncts, overlapping of classes and presence of noisy examples. Then, we show that standard techniques for induction of rule-based classifiers, such as sequential covering, top-down induction of rules or classification strategies, were created with the assumption of balanced data distribution, and we explain why they are biased towards the majority classes. Some modifications of rule-based classifiers have been already introduced, but they usually concentrate on individual problems. Therefore, we propose a novel algorithm, BRACID, which more comprehensively addresses the issues associated with imbalanced data. Its main characteristics includes a hybrid representation of rules and single examples, bottom-up learning of rules and a local classification strategy using nearest rules. The usefulness of BRACID has been evaluated in experiments on several imbalanced datasets. The results show that BRACID significantly outperforms the well known rule-based classifiers C4.5rules, RIPPER, PART, CN2, MODLEM as well as other related classifiers as RISE or K-NN. Moreover, it is comparable or better than the studied approaches specialized for imbalanced data such as generalizations of rule algorithms or combinations of

K. Napierala (✉) · J. Stefanowski
Institute of Computing Science, Poznan University of Technology, Poznan, Poland
e-mail: krystyna.napierala@cs.put.poznan.pl

J. Stefanowski
e-mail: jerzy.stefanowski@cs.put.poznan.pl

SMOTE + ENN preprocessing with PART. Finally, it improves the support of minority class rules, leading to better recognition of the minority class examples.

Keywords Rule induction · Imbalanced data · Classifiers · Nearest neighbour paradigm · Nearest rules

1 Introduction

In this paper we are concerned with improving rule-based classifiers learned from imbalanced data. Let us remind that learning *classification rules* from examples is one of the oldest and most popular tasks in machine learning and data mining. Generally speaking, such rules are represented as symbolic expressions of the following form:

$$\text{IF}(\textit{conditions})\text{THEN}(\textit{target class}),$$

where *conditions* are formed as a conjunction of elementary tests on values of attributes describing learning examples, and the rule consequence indicates the assignment of an example satisfying the condition part of the rule to a given class. Rules are one of the most popular symbolic representations of knowledge discovered from data. Several researchers have pointed out that they are more *comprehensible* and *human-readable* than other representations, in particular "black boxes" like neural networks or statistical models (as e.g. SVM). Although similar characteristics are associated with a tree representation, a set of rules is typically more compact (Michalski et al. 1998; Quinlan 1993; Zytkow 2002) than a comparable decision tree. Moreover, rule representation can be more powerful because it is not constrained by the arborescent structure of the tree. It is also claimed that 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, where these features often result in increased willingness of decision makers to accept provided suggestions and solutions (e.g., in medicine Michalski et al. 1998). Finally, rules have been successfully used in many applications, see e.g. Langley and Simon (1998), Nabney and Jenkins (1993) or some chapters in Michalski et al. (1998), Stefanowski (2001) and Klosgen and Zytkow (2002).

A number of various algorithms have been developed to induce classification rules (for a review see e.g. Furnkranz 1999; Grzymala-Busse 1992; Stefanowski 2001). Although they have been proven to be successful in solving many learning and classification problems, some data characteristics may cause difficulties and decrease the performance of induced classifiers. One of them is related to *class imbalance* in the set of learning examples. In imbalanced data one of the classes (further called a *minority class*) includes much smaller number of examples than the other classes (further referred to as *majority classes*). At the same time, examples from the minority class are usually of primary interest and their correct recognition is more important than the recognition of examples from the other classes. Such a situation often occurs in medical diagnosis, where the number of patients requiring special attention (e.g. therapy or treatment) is much smaller than the number of patients who do not need it. 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. Similar situations are observed in other domains, such as detecting fraudulent banking operations, detecting network intrusions, managing risk, predicting failures of technical equipment, and information filtering (Chawla 2005; He and Garcia 2009; Weiss 2004).

Standard learning methods often do not work properly with imbalanced data as they are in some way biased to focus on the majority classes while "disregarding" examples from the minority class. As a result, constructed classifiers better recognize new examples from the majority classes and they usually have difficulties (or even are unable) to classify correctly examples from the minority class. This problem affects various types of classifiers, including the rule-based ones, which are the main subject of the described research.

Too small number of examples in the minority class in comparison to the number of examples in the majority classes (expressed by an imbalance ratio) is not the only problem while creating classifiers from imbalanced data. Other data-related factors, which make the learning task even more difficult, include data fragmentation (in particular if these sub-concepts play a role of *small disjuncts* (Jo and Japkowicz 2004)), overlapping of the minority and majority classes, the presence of noisy or rare examples (Garcia et al. 2007; Kubat and Matwin 1997; Napierala et al. 2010; Stefanowski 2012). Performance of classifiers can be also degraded due to algorithmic factors, such as inappropriate use of greedy search strategies or evaluation measures (Weiss 2004).

A number of solutions have been proposed to improve learning classifiers from imbalanced data (He and Garcia 2009; Weiss 2004). They are usually divided into two general categories—methods operating on the *data level* and on the *algorithmic level*. Most works come from the first category and they propose different preprocessing techniques that change the distribution of examples among classes by appropriate resampling or filtering. The works concerned with the algorithmic level usually modify either the induction phase or a classification strategy, assign weights to examples, and use boosting or other multiple classifiers. Moreover, some researchers transform the problem of learning from imbalanced data to the problem of cost learning.

In this paper we are concerned with the algorithmic level. In particular, we are interested in rule-based classifiers due to the reasons mentioned in the first paragraph and a personal experience of the authors with rule induction (cf. e.g. Stefanowski 2001). Rules are, similarly to decision trees, particularly sensitive to class imbalance (Japkowicz and Stephen 2002; Van Hulse et al. 2003). Following the discussions of factors that can degrade their performance on imbalanced data (e.g. in Weiss 2004), below we briefly mention major shortcomings of standard rule induction methods. First, most algorithms induce rules using the top-down technique with *maximum generality bias*, which hinders finding rules for smaller sets of learning examples, especially from the minority class. Second, most algorithms use a *greedy sequential covering* approach, which may increase the data fragmentation and results in weaker rules, i.e., supported by a small number of learning examples. The "weakness" of the minority class rules could be also associated with the third factor—classification strategies, where minority rules have a smaller chance to contribute to the final classification decision.

Some researchers have already proposed extensions of rule-based approaches aimed at class imbalance—we review these proposals in Section 3. However, most

of them address only a single or at most a few of algorithmic or data-related factors. We hypothesize that there is still a place for a new algorithm that could resolve these issues in a more comprehensive way.

Therefore, in this paper we propose a new rule induction algorithm called BRACID (Bottom-up induction of Rules And Cases for Imbalanced Data), which aims at improving the classification performance of classifiers learned from imbalanced data. While developing this algorithm, we have attempted to deal with the above-mentioned problems demonstrated by learning methods and rule-based classifiers in the presence of class imbalance. The most important feature of BRACID is giving up the greedy sequential covering and top-down induction technique in order to overcome the problems of data fragmentation and maximum generality bias. We have decided to induce rules by following the bottom-up generalization of the most specific rules representing single examples. Keeping in mind that local algorithms are more capable to learn difficult class boundaries, our algorithm integrates rule-based and instance-based knowledge representations (in this aspect we have been inspired by the RISE algorithm (Domingos 1996)). Constructed classifiers use a nearest rule strategy to classify new coming examples. Moreover, based on some good experience with the preprocessing techniques (Stefanowski and Wilk 2008) BRACID makes use of information about the nature of examples in the neighborhood. It allows creating more rules for the minority class in the difficult regions and it handles noisy data. Finally, while constructing a classifier we optimize an evaluation measure to pay more attention to the minority class.

The usefulness of the proposed BRACID algorithm has been evaluated in a series of experiments conducted on 22 imbalanced datasets. Moreover, we have compared it against popular rule induction algorithms as well as some specific approaches dedicated for handling the imbalanced data.

The rest of the paper is organized as follows. Section 2 describes the problems related to class imbalance. We also discuss the data-related factors, which can have a negative impact on learning classifiers. In Section 3 we first present the algorithmic factors, then we give a comprehensive review of generalizations of rule classifiers, which aim to deal with these factors. The BRACID algorithm and its new classification strategy are presented in details in Section 4. Then, in Section 5 we describe the experimental setup, present the results and discuss them. Finally, the last section concludes this work.

2 Learning from imbalanced data

In the following we present only these aspects of learning classifiers from imbalanced datasets, which are the most relevant to our proposal. We describe factors in the data which cause the degradation of classifiers (Section 2.1) and evaluation measures specific for class imbalance (Section 2.2). In Section 2.1, we also review data level methods which will be used as comparative methods in the experimental evaluation, or which are at least partly related to the construction of BRACID algorithm. For a more extensive discussion of the problems with class imbalance and comprehensive reviews of other methods, see e.g. Chawla (2005), He and Garcia (2009), Weiss (2004).

2.1 Sources of difficulties in imbalanced datasets

A dataset is considered to be imbalanced if it is characterized by an unequal distribution between classes. There is no unique opinion about the degree of such imbalance between class cardinalities. Some researchers have studied the data 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 suggesting the precise values of this ratio, we repeat after (Weiss 2004) 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* (Jo and Japkowicz 2004). The imbalance 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 (Weiss 2004). Here we remark that the difficulty with lack of sufficient presence of learning examples in the rare class does not apply to binary (two-class) problems only, but it may also concern the multiclass data in which imbalance exists between various classes.

The research with imbalanced data has shown that simple class imbalance ratio is not the only and the main source of difficulty. The degradation of performance is also related to other factors characterizing data distribution, such as data complexity and its decomposition leading to small disjuncts, overlapping between classes, presence of rare instances and other (He and Garcia 2009; Weiss 2004).

Data decomposition leading to small disjuncts The classes may be scattered into smaller sub-parts representing separate sub-concepts. It is particularly critical for the minority class containing sub-concepts with limited numbers of examples. Japkowicz in her research named it *within-class imbalance* (Jo and Japkowicz 2004). This is closely related to the *problem of small disjuncts* (see Fig. 1a— black circles represent minority examples). Briefly speaking, a classifier learns a concept by generating disjunct forms (rules Holte et al. 1989) to describe it. Small disjuncts are these parts of the learned classifier which cover a too small number of examples (Weiss 2004). In case of fragmented concepts (especially in the minority class) the presence of small disjunct arises (He and Garcia 2009). It has been observed in the empirical studies that small disjuncts contribute to the classification error more than larger disjuncts. Japkowicz and her co-operators in their experiments with artificial data showed that

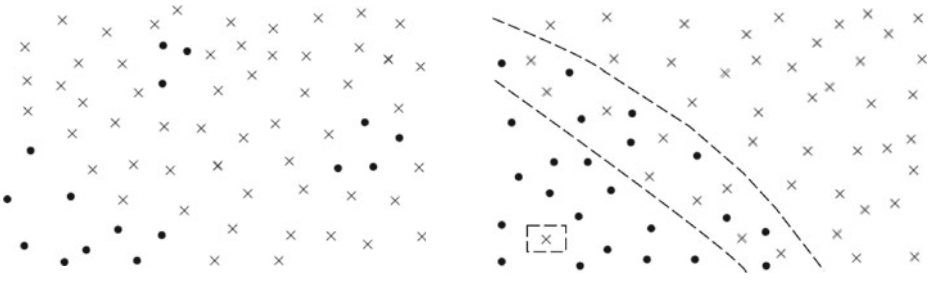

Fig. 1 Difficult data distributions in imbalanced datasets

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 (Japkowicz 2003; Jo and Japkowicz 2004). At the same time, they showed that for much larger datasets with low level of decomposition or with a sufficient number of examples in the sub-concepts, the imbalance ratio alone did not decrease so much the classification performance.

Overlapping between classes and presence of noisy instances Other researchers have explored the effect of overlapping between imbalanced classes—more recent experiments on mainly artificial data with different degrees of overlapping also showed that overlapping was more influential than the overall imbalance ratio (Garcia et al. 2007; Prati et al. 2004). Moreover, in our earlier research (Napierala et al. 2010) we showed that noisy or rare examples located inside another class (far from the decision boundary) also decreased the classification performance for the minority class.

Similar observations as to the nature of "difficult" data distributions have been made with non-artificial data. For instance, Kubat and Matwin in their paper (Kubat and Matwin 1997) claim that the characteristics of mutual positions of learning examples is a source of difficulty for learning classifiers from imbalanced data.

Related techniques on data level Solutions on the data level try to deal with the above problems by sampling or filtering the learning set. Kubat and Matwin in *one-side sampling* (Kubat and Matwin 1997) focus their attention on *noisy* majority class examples located inside the minority class and *borderline* examples (see Fig. 1b—a noisy example is marked with a dashed square). They postulate that such examples should be removed from the majority classes, while the minority class is kept unchanged. As a result, ambiguous regions around the minority class are "cleaned". This approach represents so-called *data cleaning methods* (also called *informed undersampling*). Another representative is the NCR method—see their review in He and Garcia (2009).

Other approaches perform the *oversampling* of the minority class to balance the distribution between classes. The simplest solution is to replicate the randomly selected minority examples. More advanced approaches, called *informed re-sampling*, take into account the data characteristics to perform focused *oversampling* of the minority class. A very well-known representative is SMOTE (Synthetic Minority Over-sampling Technique) introduced by Chawla et al. (2002). It analyses each example from the minority class and generates new synthetic examples along the lines between this example and some of its randomly selected k nearest neighbours from the minority class. Experiments with different classifiers showed that SMOTE can improve the recognition of the minority class.

However, SMOTE may over-generalize the minority class as it introduces artificial minority examples without taking into account the distribution of neighbouring examples from the majority classes. Therefore, SMOTE is also combined with undersampling. In the experiments we will consider one of these combined approaches, called SMOTE-ENN, which post-processes the results of SMOTE by removing the examples that may contribute to misclassification (Batista et al. 2004). More precisely, the *Edited Nearest Neighbour* Rule (ENN) is used after SMOTE generation phase to delete the examples (from all classes) misclassified by its k nearest neighbours.

Several other extensions of SMOTE have recently been proposed, such as Borderline SMOTE, Safe-Level SMOTE, or Local Neighbourhood SMOTE (see their review in He and Garcia 2009; Maciejewski and Stefanowski 2011), which additionally analyse the distribution of the majority class when generating synthetic examples for the minority class.

2.2 Evaluation of classifiers learned from imbalanced datasets

As the overall classification accuracy is biased toward the majority classes (He and Garcia 2009), special measures are considered for imbalanced data. They are usually designed for two-class problems in which class labels for the minority and majority classes are called positive and negative, respectively. When the data contains several majority classes, the classifier performance on these classes can be aggregated into one negative class. Thus, the performance of the classifier can be presented in a confusion matrix, as in Table 1.

From the confusion matrix, apart from other more elaborated measures (see e.g. review He and Garcia 2009), one can construct simpler measures concerning the recognition of the positive and negative classes:

$$TruePositiveRate = TP/(TP + FN)$$

$$TrueNegativeRate = TN/(TN + FP)$$

$$FalsePositiveRate = FP/(TN + FP)$$

$$Precision = TP/(TP + FP)$$

True Positive Rate is also called *Sensitivity* (also known as *Recall*), while True Negative Rate is called *Specificity*. As the improvement of recognizing the minority class is usually associated with the decrease of recognizing the majority classes, aggregated measures are considered to characterize the performance of the classifiers. First of all, several authors use the *ROC (Receiver Operating Characteristics) curve* analysis. ROC curve is a graphical plot of a True Positive Rate (Sensitivity) as a function of a False Positive Rate (1 − Specificity) along different threshold values characterizing the overall performance of a studied classifier. The quality of a classifier performance is reflected by the area under ROC curve, called AUC measure (Chawla 2005; Weiss 2004). AUC varies between 0 and 1, where larger values indicate better classifier performance. Other curve approaches, more appropriate for highly skewed data, are *Precision-Recall Curves*. For a review of such approaches, see He and Garcia (2009).

One can also use simpler measures aggregating two basic measures to characterize classifiers, in particular if they give purely deterministic predictions, like rules in our study (see discussion on applicability of ROC analysis in Wang and Japkowicz 2010).

Table 1 Confusion matrix for performance evaluation

	Predicted positive	Predicted negative
True positive	TP	FN
True negative	FP	TN

Kubat and Matwin (1997) proposed to use the geometric mean of Sensitivity and Specificity defined as:

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

This measure relates to a single point on the ROC curve and its key idea is to 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 (He and Garcia 2009). 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. He and Garcia (2009).

3 Related works on rule approaches to class imbalance

Many algorithms have been proposed to induce rules from examples in a standard classification perspective without class imbalance. First algorithms come from Michalski's proposal of AQ-family (Michalski et al. 1986). Its popular successors are CN2 (Clark and Niblett 1989), PRISM, LEM2 (Grzymala-Busse 1992), MODLEM (Stefanowski 1998) or ELEM2 (An and Cercone 1998). There are also algorithms which perform an intensive pruning of rules, such as IREP (Furnkranz and Widmer 1994), Grow (Cohen 1993), RIPPER (Cohen 1995) or PART (Frank and Witten 1998). Most of these algorithms (beginning from the original Michalski's proposal) are based on a *sequential covering* search technique (also called *conquer and divide*) which we will further discussed.

While classifying new-coming examples, rule sets are either ordered (as in C4.5rules (Quinlan 1993) or RIPPER) and the first matched rule indicates a decision, or the rules are undordered and special strategies for solving conflicts between matched rules have to be applied (see e.g. LERS strategy (Grzymala-Busse 1994), *nearest rules* (Stefanowski 1993, 2007) or special measures for single rules such as *m-estimate* (Dzeroski et al. 1993)). For a more comprehensive review of the current state of the art in rule induction see, e.g., Furnkranz (1999), Flach and Lavrac (2003), Tan et al. (2005).

3.1 Influence of class imbalance on rule classifiers

Most of the existing rule induction algorithms share a number of problems when it comes to learning from imbalanced datasets. The most comprehensive and systematic study was presented in Weiss (2004). Although it concerns data mining in general, most of the observations are also true for rule approaches.

Top-down induction technique Most rules are induced in a top-down manner (also called general-to-specific). A construction of a rule begins with the most general (empty) rule, and it is repeatedly specialised with new conditions as long as it still covers negative examples (or until other stopping criterion is met). Top-down

technique is used to favor general rules and to 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, the maximum-generality bias works well for large disjuncts but it has difficulties with identifying the small disjuncts (Holte et al. 1989). 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 (Weiss 2004). 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 or Laplace measure, support and confidence measures). Due to rarity of the minority examples, their impact on the accuracy and generality is much smaller than for common (majority) examples.

Greedy, sequential covering technique Nearly all popular rule induction algorithms employ a sequential covering search technique to find a minimal set of rules which covers the dataset. The covering algorithm repeatedly generates new rules 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 (Flach and Lavrac 2003). Removing the examples during training partitions the space of examples into smaller and smaller pieces 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 for inducing 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 minority class rules are usually more specific and supported by fewer examples, they can be characterized by worse values of evaluation measures than rules for the majority class. It can cause a classification bias toward the majority class. In particular it concerns the unordered sets of rules where classification strategies are often based on voting of rules with weights depending on their evaluation measures (see e.g. empirical studies conducted in Weiss and Provost 2003; Grzymala-Busse et al. 2004).

3.2 Related works on rules and class imbalance

Several approaches have been proposed to deal with the above problems. The first group of solutions uses less greedy techniques of rule induction aiming at finding more rules for the minority class, and/or improving their evaluation measures. This helps to find rules for small "nuggets" of information, and to increase the chance of classifying new examples as minority ones.

RLSD (Zhang et al. 2004) 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. RLSD was used to find patterns for fraudulent cases in law domain. BRUTE algorithm (Riddle et al. 1994) also performs a more exhaustive search, looking for accurate 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. Another example is the EXPLORE algorithm (Stefanowski and Wilk 2009; Grzymala-Busse et al. 2004), which performs a less greedy search for the minority rules, looking for all rules that satisfy a certain threshold for rule supports. At the same time, 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.

Other solutions also try to improve the generality of the minority rules, but they concentrate on the post-pruning phase of rule induction. In IDL (Nguyen and Ho 2005), 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 these rules with the local neighourhood belonging to the same class.

Grzymala et al. (2000) optimizes the *strength* of minority rules (referring to the support measure) in yet different way, by modifying the classification strategy. It introduces a constant *strength multiplier* which is used to multiply the strength of minority rules when conflicts between the classes occur during classification. A value of a multiplier is optimized for a given dataset to maximize an aggregated measure of Sensitivity and Specificity.

Some solutions try to deal with a learning bias of classifiers caused by the used evaluation metrics, which favor the majority class and can fail to find the rules for small disjuncts (characteristic for minority classes). Holte et al. in (1989) change the bias of a well-known 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 based 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 five 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 An et al. (2001), the authors consider the ELEM2 sequential covering algorithm and modify its evaluation measure and the post-pruning phase. They analyse 11 different evaluation measures and show that the recognition of a minority class strongly depends on the particular measure. Furthermore they propose to post-prune only the minority rules to obtain stronger rules, while leaving the majority rules unprunned.

Using more appropriate evaluation metrics is used also in the PN-rule algorithm (Joshi et al. 2001). This approach is motivated by an observation, that missing the rare cases is a result of optimizing precision and recall simultaneously. PN-rule is

composed 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.

The last 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 (Kubat et al. 1997) 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 pre-processing phase, relabel all majority examples into minority ones in this region and to induce minimal sets of rules for each class (Stefanowski and Wilk 2006). Another algorithm proposed in Liu et al. (2008) 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.

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 imbalanced data. For example, in Blaszczynski et al. (2010) the authors use an ensemble of rule classifiers, which is based on an Ivotes learning scheme. They use the abstaining classification strategy and selective preprocessing of examples to make the ensemble more sensitive to the minority class.

In Garcia et al. (2009), an evolutionary algorithm 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 Milar et al. (2011), 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 classical 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. Evolutionary algorithms are used also in Orriols-Puig et al. (2007). Here the authors propose a rule induction evolutionary algorithm 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.

3.3 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.

Instance-based learning (IBL) is a complementary induction paradigm to rules (RBL) and it is based on the classification according to the similarity of a new example to its local neighbours. 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 Ting (1994). He first uses a decision-tree learner (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 learner is used.

We think that a hybrid use of both, complementary paradigms is a good direction for learning with class imbalance. 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 another 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 (Flach and Lavrac 2003), 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 (Domingos 1996). 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 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.

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 when 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.

The bottom-up search, as well as the nearest rule classification strategy, rely on the distance metrics between the examples (or between rule and example). It can be calculated using a Euclidean or city-block distance metrics (McCane and Albert 2008) defined for numerical attributes, or using a value-difference metrics (Stanfill and Waltz 1986) for mixed domains with nominal and numerical attributes (as it is done in RISE).

In Domingos (1996) RISE was compared experimentally with IBL and three rule learning algorithms (PEBLS, CN2 and C4.5rules) on 30 datasets, using the overall accuracy. According to the Wilcoxon test, RISE was significantly more accurate 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 (Salzberg 1991) 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 (Luaces 2003) 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 (Zhang 1997) 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.

4 BRACID algorithm

4.1 Motivations

In Section 3.1 we discussed several problems on the algorithmic level causing the degradation of rule-based classifiers. Moreover, in Section 2 we showed other problems concerning characteristics of the data. The review of extensions of rule algorithms presented in Section 3 shows that these solutions 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 (Zhang et al. 2004)) or change the maximum-generality bias (e.g. a modification of CN2 algorithm (Holte et al. 1989)). In our opinion, such "selective" approaches are not satisfactory and cannot handle sufficiently too complex difficulties of class imbalance. Therefore, we want to introduce yet another rule induction algorithm, which tries to deal with more of these problems—more precisely with *all* the main drawbacks mentioned in the previous sections.

Briefly speaking, we decided to choose an integrated representation of rules and single instances (see motivations discussed in Section 3.3). Other crucial assumptions include: using a less greedy bottom-up induction of rules from single examples with the specific generalization by looking for 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, and a special treatment of the borderline or noisy examples. These assumptions will be described in detail in the next subsections.

4.2 Notation and basic concepts

The BRACID name is the acronym of Bottom-up induction of Rules And Cases for Imbalanced Data. Before describing it let us introduce some basic concepts.

We assume that each learning example x is represented as a set of attribute–value pairs $(x_i = v_{ij})$ and class label K_l to which it belongs; where x_i represents i-th attribute characterizing an example, v_{ij} is a single value from its domain and $K = \{K_1, \ldots K_k\}$ is a set of classes. Attributes can be defined either on numeric or nominal scales. In a current form, BRACID works with two-class problems of which one is a minority class and the other one is the majority class. For problems with more classes, all the examples from classes other than a selected minority class are merged into a single majority class. As a result, $K = \{K_{\min}, K_{\mathrm{maj}}\}$.

A rule R is represented in the form **if** P **then** K_l, where P is the condition part and the decision part indicates that examples satisfying P should be assigned to a class K_l. The condition part P is a conjunction of elementary conditions being tests on a value (or values) of a corresponding attribute. There is at most one elementary condition per single attribute. For nominal attributes it is a single equality test of the form $(x_i = v_{ij})$. Conditions for numeric attributes are represented as closed intervals $(v_{i,\mathrm{lower}} \leq x_i \leq v_{i,\mathrm{upper}})$, where $v_{i,\mathrm{lower}} \leq v_{i,\mathrm{upper}}$ 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 intervals are degenerated to a single point $v_{i,\mathrm{lower}} = v_{i,\mathrm{upper}}$.

A rule covers an example if it satisfies all conditions in the rule. As a rule can be generalized from a single example, we introduce the term *seed of rule* R, to denote the 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 its k-nearest

neighbours while other (unsafe) examples are misclassified—following the typology introduced in Stefanowski and Wilk (2007). Among the misclassified examples we distinguish *noisy* examples if all neighbours belong to the opposite classes, otherwise they are treated as *borderline*.

To determine the neighbours, either when performing a bottom-up generalization or when classifying new examples, we need to calculate the distance between examples and/or rules. Following some literature inspirations (McCane and Albert 2008; Wilson and Martinez 1997) and our earlier experience with pre-processing methods (Stefanowski and Wilk 2007, 2008) we chose the *Heterogenous Value Difference Metrics* (HVDM). It aggregates normalized Euclidean distances for numeric attributes with Stanfil and Valtz value difference metric for nominal attributes (Stanfill and Waltz 1986; Wilson and Martinez 1997).

More precisely, let x be an example or a rule and y be another example (possible neighbour). In case of a rule x we calculate distances only for these attributes C which occur in its condition part. The aggregated HVDM is defined as

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

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:

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

Value difference metric (a simplified form without tuning attributes' weights) is defined as (Stanfill and Waltz 1986):

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

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 partial distance for numeric attributes is defined as

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

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum values for the i-th attribute.

4.3 Algorithm description

A pseudo-code of the main procedure of BRACID is presented in Algorithm 1. Let us remark that its general loop, final hybrid representation and the idea of starting the search from a rule set corresponding to a set of training examples are inspired by

Algorithm 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
```

RISE algorithm (Domingos 1996). However, many subparts are solved differently and several new elements are introduced to deal with the class imbalance specificity. While discussing the code we will refer to the critical factors mentioned in Section 3.

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 non-linear (difficult) 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 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 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.

Generalization of a rule Generalization of a rule is done using the `MostSpecific Generalization` procedure (Algorithm 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 are minimally broadened to cover the example (lines 6–9).

Algorithm 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 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 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. Thus, we choose the F-measure, which aggregates *recall* and *precision* measures. Both these measures are defined with respect to the positive (minority) class, 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, proposed in RISE. Each learning example is classified by its nearest rule and based on the accuracy of classification decisions, 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 1) using the nearest neighbours as described in Section 4.2.

The BRACID algorithm treats rules differently, depending on the tag and 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 3.

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 between classes and could be overwhelmed by the majority class examples. Thus, we assume that an `UNSAFE` minority

Algorithm 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
```

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 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 being a kind of outlier. Otherwise, the learning set would possess at least one similar majority example, as we assume that this class is well represented in the dataset. In this case, BRACID removes this rule from a set of rules. It also removes the corresponding example from a set of learning examples, 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 1).

Analogous cases from the minority class 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.

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 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 1).

The `Extend` procedure (Algorithm 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).

Let us notice that 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 1). We assume that such examples may be outliers and we do not want to amplify such regions.

Algorithm 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})
```

4.4 Evaluation of computational costs

One can ask a question 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 (Domingos 1996). 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 Domingos 1996 for details). 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.

4.5 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 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. Typically, strategies based either on voting of rules with appropriate evaluation measures (as rule supports or more sophisticated ones; cf. Grzymala-Busse 1994; An 2003; Stefanowski 1995; Yao and Zhong 1999) or on the identification of one best rule according to the additional quality measure (Furnkranz 1999; Janssen and Furnkranz 2008), are biased toward the majority class. For an extended discussion on these strategies see Grzymala-Busse et al. (2000, 2004), Weiss (2004) and Napierala et al. (2010). Therefore, for the imbalanced problems, a less biased classification strategy is needed.

Yet another issue is that BRACID produces both rules and single examples. Having such a combined, double knowledge representation we have decided to make the classification decision on the basis of the *local neighbourhood* of the new example. It means that 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 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 the local strategy may be less biased towards the majority class. Let us also remind that the nearest rule strategy was also successfully used in the earlier works of Stefanowski (1993, 1995), as well as in the related RISE algorithm (Domingos 1994). 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 metrics as in BRACID—see Section 3.2 for details. 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 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 near rules. For instance, Domingos in RISE (Domingos 1994) proposed to choose the rule with the highest value of accuracy calculated with Laplace correction (Niblett 1987), estimated on the very specific choice of so-called winning examples. However, we noticed in the preliminary experiments 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). So, confidence-based measures might not sufficiently discriminate the rules. Additionally, using the Laplace correction favors the majority rules (we will explain it using a toy example). In BRACID the situation is analogous. 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:

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

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

Let us remark that 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 4.3).

Figure 2 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 (marked as rectangles) are equally distant from it. 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.

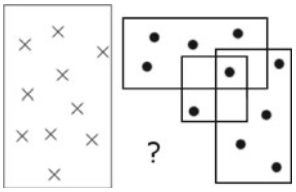

Fig. 2 An example of a conflict situation while using the nearest rule classification strategy

5 Experiments

5.1 Experimental setup

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 a few methods which are dedicated to deal with class imbalance.

We carry out the experiments on 22 datasets. 20 of them come from the UCI repository,[1] 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 (Wilk et al. 2005; Michalowski et al. 2005). The datasets represent a wide range of domains, imbalance ratios (from 3 to 35%), sizes (from 100 to over 4,000 examples) and attributes (purely nominal, purely numeric and mixed). 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 minority class ratio is rather high (e.g. *pima* or *ionosphere*). However, they are also characterized by other influential factors as overlapping decision classes or presence of noise or rare examples which is consistent with the assumptions behind our approach. Moreover, we choose them as they were often used in other experimental studies with related methods for class imbalance. Table 2 summarizes the main characteristics of the datasets.

The performance of all classifiers is evaluated by three measures: Sensitivity of the minority class and two aggregating measures—G-mean and F-measure (see their definitions in Section 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 us also remind 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 Wang and Japkowicz (2010) and other arguments in Section 2.

All 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.

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

Table 2 Basic characteristics of datasets

Dataset	No of examples	Minority class size	Imbalance ratio (%)	No of attributes (numeric)	Minority class name
Abalone	4,177	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	1,728	69	3.99	6 (0)	Good
Cleveland	303	35	11.55	13 (6)	Positive
Cmc	1,473	333	22.61	9 (2)	Long-term
Credit-g	1,000	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	1,066	43	4.03	12 (0)	F
Transfusion	748	178	23.80	4 (4)	Yes
Vehicle	846	199	23.52	18 (18)	Van
Yeast	1,484	51	3.44	8 (8)	ME2

5.2 Studying the role of BRACID's components

First, we would like to evaluate the influence of BRACID's components on its final classification abilities. More precisely, we will study the impact of: the new classification strategy described in Section 4.5 (called in this experiment component C), removal of noisy majority examples (component N) and 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.

Figure 3 shows how these three components influence the Sensitivity measure. To improve the readability of the figures, we present only a 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 and extending the minority rules brings further improvements.

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. Figure 4 (presenting values of G-mean for all the datasets) shows however, that this aggregated measure also improves from left to right bars. It may indicate that these

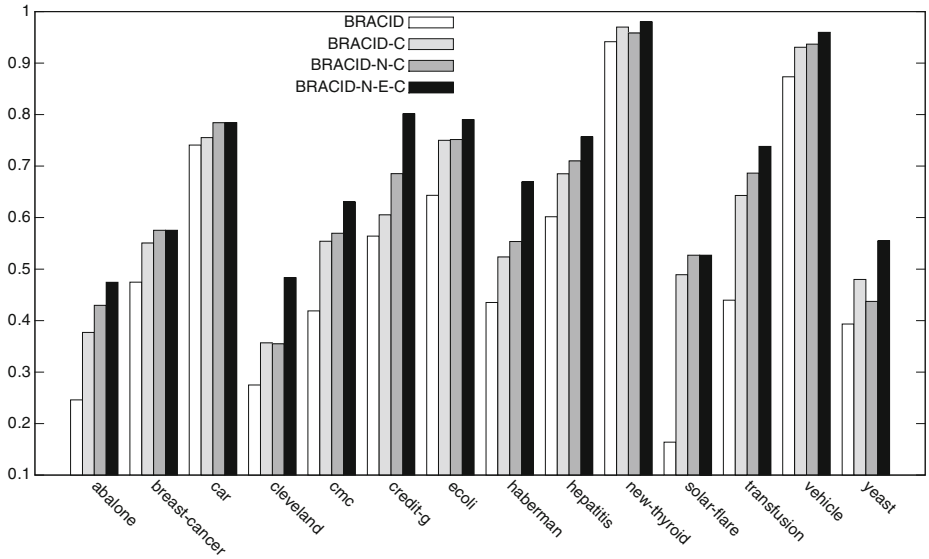

Fig. 3 Influence of BRACID's components on sensitivity measure

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.

Finally, we want to analyse how these components affect the average rule support in the minority class. On Fig. 5 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

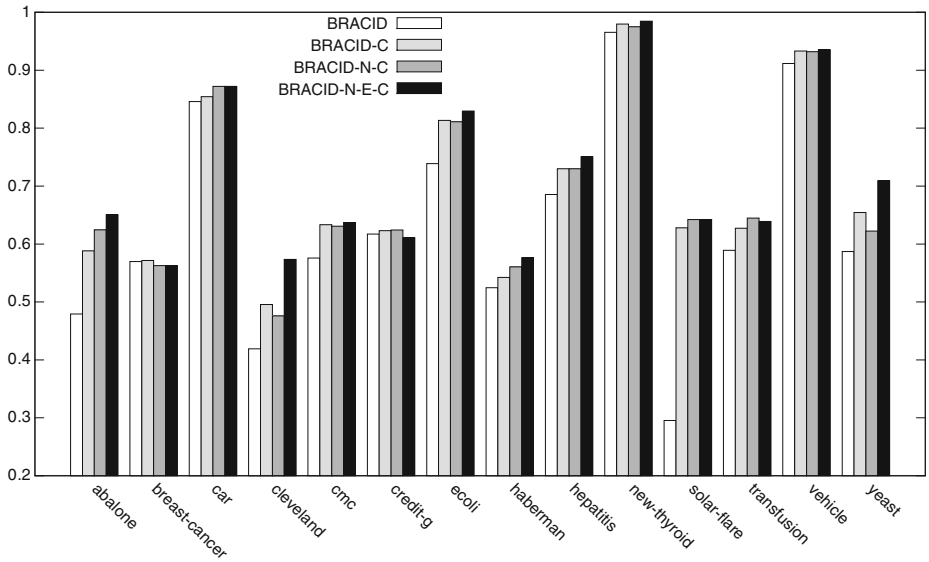

Fig. 4 Influence of BRACID's components on G-mean measure

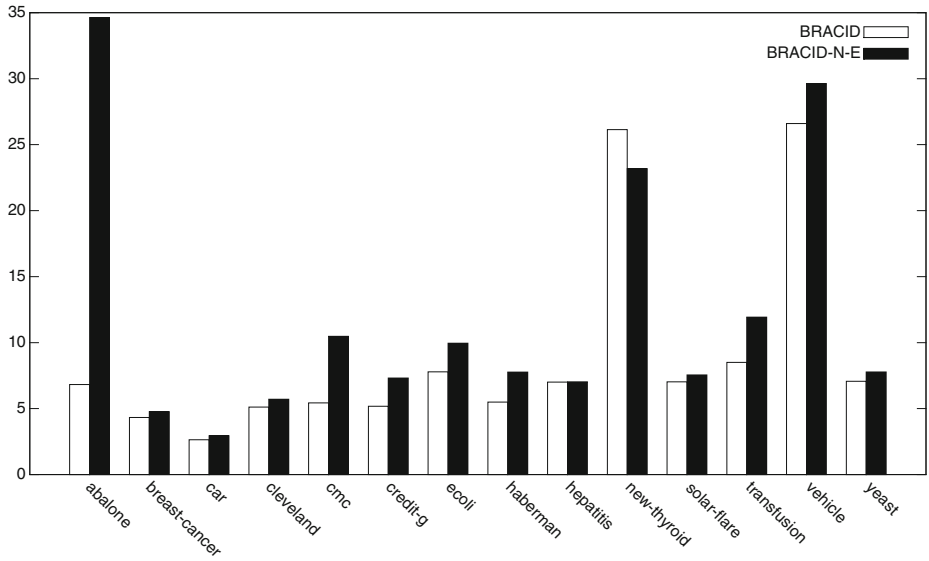

Fig. 5 Influence of BRACID's components on the average rule support for the minority class

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 rule, 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.

5.3 Comparison with standard rule classifiers

In the first phase of experiments, we have decided to compare classification performance of BRACID against four very popular rule algorithms: CN2 (Clark and Niblett 1989), PART (Frank and Witten 1998), RIPPER (Cohen 1995) and C4.5rules—Quinlan's rule list classifier obtained as a post-processing of C4.5 decision tree (Quinlan 1993). We also compare it to MODLEM (Stefanowski 1998) algorithm, as it was already used for imbalanced data (for example in Stefenowski and Wilk 2007, 2008) and its extention MODLEM-C will be used in the experiments presented in the next subsection.

CN2 is run with Laplace Accuracy as a measure evaluating conditions, beam size=5 and it produces unordered rules (classification strategy—voting with rule supports in multiple matching and default rule in non-matching). RIPPER is run with standard parameters (including rule pruning) and its typical classification strategy using an ordered list. PART, C4.5 and MODLEM are also started with standard parameters, however without prunning. Modlem is used with standard Grzymala classification strategy (Grzymala-Busse 1994). BRACID is parameterized only with the neighbourhood size. We tested values 3, 5 and 7, which are often applied to neighbourhood-based methods (such as kNN) and pre-processing 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.

Table 3 Sensitivity for BRACID compared against standard algorithms

Dataset	BRACID	RISE	kNN	C45.rules	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
Breast-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

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 C4.5rules algorithms, the original authors' implementations were used.[2] All other implementations come from the WEKA library. BRACID was implemented by us using WEKA components.

Tables 3, 4 and 5 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 (Kononenko and Kukar 2007; Japkowicz and Shah 2011). The null-hypothesis in this test is that all compared classifiers perform equally well. It uses ranks of all classifiers on each of the data sets. The lower rank, the better classifier.

We started from analyzing the results for the Sensitivity measure. Friedman statistics for these results gives 92.35 which exceeds the critical value for confidence level 0.05 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 1.09; RISE 4.48; KNN

[2]CN2, RISE and MODLEM are available on the corresponding authors' websites, a code of C4.5rules was attached to a book (Quinlan 1993).

Table 4 G-mean for BRACID and standard algorithms

Dataset	BRACID	RISE	kNN	C45.rules	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

5.5; C45rules 3.75; CN2 6.75; PART 2.85; RIPPER 6.21; MODLEM 5.43. Then, we carried out a complete post-hoc analysis of differences between classifiers with a Nemenyi test. 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 75.236 and we can again reject the null hypothesis. The average ranks of the classifiers are the following: BRACID 1.31; RISE 4.34; KNN 5.25; C45rules 3.89; CN2 6.79; PART 3.27; RIPPER 5.86; MODLEM 5.36. 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 get a better insight in the comparison of these 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 (Kononenko and Kukar 2007; Demsar 2006). 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. One can easily notice that BRACID can better recognize the minority class than

Table 5 F-measure for BRACID and standard algorithms

Dataset	BRACID	RISE	kNN	C45.rules	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
Breast-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

all other classifiers (Table 3). In particular, improvements of Sensitivity, sometimes relatively high, are 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 5.5. We can also say that this improved recognition of the minority class does not degrade too much the recognition of the majority class—see values of G-mean in the Table 4 which 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%), the degradation on the majority class is more serious and it influences the G-mean measure. The same observation refers to the F-measure (see the Table 5).

5.4 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 suitable 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 3 was impossible (it appears that most of these algorithms are not available publicly or their authors do not maintain the software anymore). We received a Modlem-C implementation,[3] which is a generalization of the MODLEM algorithm with a modified Grzymala classification strategy (Grzymala-Busse et al. 2000)—see also

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

Section 3.2. For each dataset separately, we tested 10 possible values of a strength multiplier (from 1–10) and chose the best one (according to F-measure and G-mean). 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. We have chosen two methods for handling class imbalance which transform the original class distribution into a more balanced one in a pre-processing step. First, we direct our interest to a well known SMOTE algorithm (Chawla et al. 2002) as in many experimental studies it has been evaluated to be one of the most efficient methods of this category 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, to get an even more competitive classifier, we include an extension of SMOTE—SMOTE-ENN—as it should improve even more the abilities of the PART rule classifier; see Section 2.1 for more details of SMOTE-ENN.

Here we should stress that our aim in this part of the experiment is not to generally study the pre-processing methods as they are based on different principles than rule

Table 6 Sensitivity for algorithms specialized 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
Breast-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

algorithms. We want to check whether BRACID is not worse or competitive to the well known representative of these methods.

The results, presented in Tables 6, 7 and 8, 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 3). Similarly, PART+SMOTE and PART+SMOTE+ENN work better than PART alone. We conducted again the Friedman test for all classifiers. For all measures we can reject the null hypothesis. Critical values are: Sensitivity 54.94; G-means 34.78 and 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 1.28; RISE 4.59; MODLEM-C 3.89; SMOTE+PART 2.93 and SMOTE+ENN+PART 2.25. G-mean—BRACID 1.75; RISE 3.93; MODLEM-C 3.66; SMOTE+PART 3.16 and SMOTE +ENN+PART 2.51. F-measure—BRACID 1.68; RISE 3.96; MODLEM-C 3.71; SMOTE+PART 3.16 and SMOTE+ENN+PART 2.53. Using the critical difference $CD = 1.3$ we cannot say that BRACID is significantly better than SMOTE+ENN+PART; however the difference between them is around 1 in favor of BRACID. All the other algorithms are ouperformed by BRACID. In all cases RISE is the worst algorithm while MODLEM-C is worse than SMOTE combined with PART.

Again we applied the Wilcoxon signed rank test to verify more deeply the differences between BRACID and SMOTE+ENN. With respect to Sensitivity, BRACID is significantly better than SMOTE+ENN+PART ($p < 0.0308$). For G-mean and F-measure BRACID is better with $p < 0.043$ and $p < 0.028$, respec-

Table 7 G-mean for algorithms specialized 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 8 F-measure for algorithms specialized 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
Breast-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

tively. Determining win-loss between them also shows that BRACID dominates SMOTE+ENN+PART for 14–15 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 classification performance is comparable or even slightly better than the best method for informed re-sampling used together with the most competitive rule algorithm PART.

5.5 Analysis of rule sets

To analyse the differences between the induced rule sets for the selected algorithms, we decided to calculate some descriptive statistics such as the 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 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 C4.5rules 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 9, include additionally the number of maximally specific rules representing single minority examples in the final BRACID classifier. This parameter shows how hybrid is the knowledge representation corresponding to the minority class for a given dataset. It

Table 9 Rule statistics

Dataset	Classifier	No of rules (MIN)	No of rules (MAJ)	No of cases	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

could also refer to the difficulty of data—a lot of maximally specific minority rules suggests a limited number of large disjuncts and more difficult border region between the classes.

We present the results for 9 selected datasets only. The first five datasets (over the double horizontal line) represent the standard behaviour of the algorithms, which we also observed for the remaining datasets, not included in this table due to limits of its size. The last four datasets represent different untypical situations which we will further discuss. Typically (first five 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 notice 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 six times more minority rules than CN2. However, if we come back to Table 3 and analyse its results, 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 9 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 3 shows, however, that these rules were completely useless—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 notice that this dataset is more balanced than other datasets (28%), which may be the reason why CN2 can learn it reasonably well. The same behaviour was observed for another dataset not included in Table 9, *ionosphere*, which is even more balanced (36%).

Finally, we report the rule statistics for *car* dataset to check why BRACID performed on it worse than classical PART algorithm and comparably to MODLEM. First of all, it can be seen that the distribution of examples in this dataset seems to be very scattered—almost 35% of the final BRACID classifier corresponds to single cases. As a result, the average rule strength of BRACID is rather low compared to other algorithms. 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.

6 Conclusions and future works

In this paper we have considered improving rule-based classifiers learned from imbalanced data. We have started our study from identifying these data-related factors that make learning difficult and pointed out such problems as data fragmentation, overlapping of classes or noisy examples. Then, we have considered algorithmic factors and showed that in particular greedy search strategies, top-down induction of rules, choice of evaluation measures to control generation of rules, and classification strategies can have adverse effects on the recognition of the minority classes.

We have presented the up-to-date review of the existing modifications of rule learning algorithms suited for class imbalance, and concluded that most of them concentrate on a single or at most a few selected algorithmic or data-related factors. Therefore, we have introduced a new rule algorithm, called BRACID, which deals with the described factors in a much more comprehensive way.

While developing BRACID we have addressed these factors at many levels. The most important features of BRACID, which in our opinion should improve classifiers constructed from imbalanced data, are:

- 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 or noisy regions 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.
- It handles 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 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 actual learning examples. This significantly distinguishes BRACID from preprocessing methods based on oversampling (e.g., 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 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 state-of-the-art rule-based 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 than the most competitive rule algorithm—PART.
- BRACID is able to better recognize the minority class than other compared algorithms (except for the car dataset, for which PART is superior; in Section 5.5 we have showed that BRACID could not produce good rules for this dataset);
- The improvement of the sensitivity measure is associated with a very 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 approaches specialized for class imbalance, including the integration of the PART algorithm with the basic version SMOTE. Only after extending SMOTE by Edited Nearest Neighbor Rule (ENN), the difference of average ranks between this approach

(SMOTE+ENN+PART) 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 approach to informed re-sampling data before inducing rules and it is a well known, effective solution at the data level. Moreover, BRACID could be seen as better with respect to an 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.

We are aware of using a limited number of rule-based classifiers specialized for class imbalance in the second part of our comparative experiment. Although in the current study we were unable to access the most interesting of generalized rule classifiers described in Section 3 as they are not available publicly, in future research we plan to extend the experiments including at least one of these classifiers which handle several factors.

Yet another topic for our future research concerns relating data properties with a successful use of BRACID. Following our other studies on the nature of class imbalance problem, we claim that the imbalance ratio is not the only and main source of difficulties for learning classifiers (Napierala et al. 2010). We have paid attention to such data factors as decomposition of the minority class into many sub-concepts with very few examples, overlapping of the minority and majority classes, and the presence of noisy, outlier and rare examples. So, in our future work we plan to conduct a series of experiments on specially prepared artificial datasets to evaluate separately the effectiveness of BRACID on data sets suffering from specific factors from the above list. By controlling and changing the intensity of each factor we should be able to check more precisely its influence on the performance of BRACID and possibly other competitive classifiers. Such experiments with artificial data could extend our current knowledge resulting from running algorithms on different real data. Preliminary research on constructing such artificial datasets and evaluating different classifiers have been described in Napierala et al. (2010) and Stefanowski (2012).

Acknowledgements The research has been supported by the Ministry of Science and Higher Education, grant no. N N519 441939. The authors would like to thank the anonymous reviewers for their useful comments and suggestions, which were very helpful in improving this study.

References

An, A. (2003). Learning classification rules from data. *Computers and Mathematics with Applications, 45*, 737–748.

An, A., & Cercone, N. (1998). Elem2: A learning system for more accurate classifications. In *Proceedings of the 12th Conference on Advances in Artificial Intelligence* (pp. 426–441).

An, A., Cercone, N., & Huang, X. (2001). A case study for learning from imbalanced data sets. In *Proceedings of the 14th Canadian conference on Artificial Intelligence (AI2001)*, (pp. 1–15).
Batista, G., Prati, R. C. & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter, 6*(1), 20–29.
Blaszczynski, J., Deckert, M., Stefanowski, J., & Wilk, S. (2010). Integrating selective pre-processing of imbalanced data with ivotes ensemble. In *LNAI* (Vol. 6086, pp. 148–157). Verlag: Springer.
Chawla, N. (2005). Data mining for imbalanced datasets: An overview. In O. Maimon, & L. Rokach (Eds.), The data mining and knowledge discovery handbook (pp. 853–867). Springer Verlag.
Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR), 16*, 321–357.
Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*, 261–283.
Cohen, W. (1993). Efficient pruning methods for separate-and-conquer rule learning systems. In *Proceedings of the 13th international joint conference on artificial intelligence* (pp. 988–994).
Cohen, W. (1995). Fast effective rule induction. In *Proceedings of the 12th international conference on machine learning* (pp. 115–123).
Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.
Domingos, P. (1994). The RISE system: Conquering without separating. In *Proceedings of 6th IEEE international conference on tools with artificial intelligence* (pp. 704–707). IEEE Computer Society Press.
Domingos, P. (1996). Unifying instance-based and rule-based induction. *Machine Learning, 24*, 141–168.
Dzeroski, S., Cestnik, B., & Petrovski, I. (1993). Using the m-estimate in rule induction. *Journal of computing and information technology* (pp. 37–46).
Flach, P., & Lavrac, N. (2003). Rule induction. In M. Berthold, & D. Hand (Eds.), *Intelligent data analysis: An introduction* (pp. 229–267). Springer.
Frank, E., & Witten, I. (1998). Generating accurate rule sets without global optimization. In *Proceedings of the 15th int. conf. on machine learning* (pp. 144–151).
Furnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review, 13*(1), 3–54.
Furnkranz, J., & Widmer, G. (1994). Incremental reduced error pruning. In *Proceedings of the int. conf. on machine learning* (pp. 70–77).
Garcia, S., Fernandez, A., & Herrera, F. (2009). 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.
Garcia, V., Sanchez, J., & Mollineda, R. (2007). An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In *Proceedings of the 12th iberoamerican conf. on progress in pattern recognition, image analysis and applications* (pp. 397–406).
Grzymala-Busse, J. (1992). LERS—a system for learning from examples based on rough sets. In R. Slowinski (Ed.), *Intelligent decision support* (pp. 3–18). Kluwer Academic Publishers.
Grzymala-Busse, J. (1994). Managing uncertainty in machine learning from examples. In *Proceedings of the 3rd international symposium in intelligent systems* (pp. 70–84). IPI PAN Press.
Grzymala-Busse, J., Goodwin, L., Grzymala-Busse, W., & Zheng, X. (2000). 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* (pp. 69–74).
Grzymala-Busse, J., Stefanowski, J., & Wilk, S. (2004). 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*. *LNCS* (Vol. 3213 pp. 757–763). Springer.
He, H., & Garcia, E. (2009). Learning from imbalanced data. *IEEE Transactions on Data and Knowledge Engineering, 21*(9), 1263–1284.
Holte, R., Acker, L., & Porter, B. (1989). Concept learning and the problem of small disjuncts. In *Proceedings of the 11th international joint conference on artificial intelligence* (pp. 813–818).
Janssen, F., & Furnkranz, J. (2008). An empirical investigation of the trade-off between consistency and coverage in rule learning heuristics. In *Proceedings of the 11th international conference on discovery science*.
Japkowicz, N. (2003). Class imbalance: Are we focusing on the right issue? In *Proceedings of 2nd workshop on learning from imbalanced data sets (ICML)* (pp. 17–23).
Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithm: A classification perspective*. Cambridge University Press.

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

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter, 6*, 40–49.

Joshi, M., Agarwal, R., & Kumar, V. (2001). Mining needles in a haystack: Classifying rare classes via two-phase rule induction. In *Proceedings of the SIGMOD KDD conference on management of data*. (pp. 91–102). ACM, New York, USA.

Klosgen, W., & Zytkow, J., eds. (2002). *Handbook of data mining and knowledge discovery*. Oxford University Press.

Kononenko, I., & Kukar, M. (2007). *Machine learning and data mining*. Horwood Pub.

Kubat, M., Holte, R., & Matwin, S. (1997). Learning when negative examples abound. In *Proceedings of the 9th European conference on machine learning* (pp. 146–153).

Kubat, M., & Matwin, S. (1997). Addresing the curse of imbalanced training sets: one-side selection. In *Proceedings of the 14th int. conf. on machine learning* (pp. 179–186).

Langley, P., & Simon, H. (1998). Fielded applications of machine learning. In R. Michalski, I. Bratko, & M. Kubat (Eds.), *Machine learning and data mining* (pp. 113–129). John Wiley & Sons.

Liu, Y., Feng, B., & Bai, G. (2008). 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*. *LNAI* (Vol. 5306, pp. 262–271). Springer-Verlag.

Luaces, O. (2003). Inflating examples to obtain rules. *International Journal of Intelligent Systems, 18*, 1113–1143.

Maciejewski, T., & Stefanowski, J. (2011). Local neighbourhood extension of smote for mining imbalanced data. In *Proceedings of the IEEE symposium on computational intelligence and data mining* (pp. 104–111). IEEE Press.

McCane, B., & Albert, M. (2008). Distance functions for categorical and mixed variables. *Pattern Recognition Letters, 29*, 986–993.

Michalowski, W., Wilk, S., Farion, K., Pike, J., Rubin, S., & Sowiski, R. (2005). 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.

Michalski, R., Bratko, I., & Bratko, A., eds. (1998). *Machine learning and data mining; methods and applications*. John Wiley & Sons, Inc.

Michalski, R., Mozetic, I., Hong, J., & Lavrac, N. (1986). The multi-purpose incremental learning system aq15 and its testing application in three medical domains. In *Proceedings of 5th national conference on AI* (pp. 619–625). AAAI-Press.

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

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

Napierala, K., Stefanowski, J., & Wilk, S. (2010). Learning from imbalanced data in presence of noisy and borderline examples. In *Proceedings of the conf. on rough sets and current trends in computing* (Vol. 6086, pp. 148–157). *LNCS*, Springer-Verlag.

Nguyen, C., & Ho, T. (2005). An imbalanced data rule learner. In *Proceedings of 9th European conference on principles and Practice of Knowledge Discovery in Databases* (PKDD05) (pp. 617–624).

Niblett, T. (1987). Constructing decision trees in noisy domains. In *Proceedings of EWSL* (pp. 67–78).

Orriols-Puig, A., Goldberg, D., Sastry, K., & Bernado-Mansilla, E. (2007). Modeling xcs in class imbalances: Population size and parameter settings. In *Proceedings of the 9th annual conference on genetic and evolutionary computation* (pp. 1838–1845). GECCO, ACM.

Prati, R. C., Batista, G., & Monard, M. C. (2004). Class imbalances versus class overlapping: An analysis of a learning system behavior. *Advances in artificial intelligence* (pp. 704–707).

Quinlan, J. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

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

Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning, 6*, 251–276.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM, 29*(12), 1213–1228.

Stefanowski, J. (1993). Classification support based on the rough sets. *Foundations of Computing and Decision Sciences, 18*, 371–380.

Stefanowski, J. (1995). Using valued closeness relation in classification support of new objects. In T. Lin, & A. Wildberg (Eds.), *Soft computing: Rough sets, fuzzy logic, neural networks, uncertainty management, knowledge discovery* (pp. 324–327). Simulation Council Inc.

Stefanowski, J. (1998). Rough set based rule induction techniques for classification problems. In *Proceedings of 6th European congress on intelligent techniques and soft computing* (Vol. 1, pp. 109–113).

Stefanowski, J. (2001). *Algorithms of rule induction for knowledge discovery*. Habilitation Thesis published as Series Rozprawy no. 361, PUT Publishing House (in Polish).

Stefanowski, J. (2007). On combined classifiers, rule induction and rough sets. *Transactions on Rough Sets, 6*, 329–350.

Stefanowski, J. (2012). Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. Chapter in Jain, L., Howlett, R., Ramanna S. (Eds). Emerging Paradigms in Machine Learning and Applications. Springer Verlag (to appear).

Stefanowski, J., & Wilk, S. (2006). Rough sets for handling imbalanced data: Combining filtering and rule-based classifiers. *Fundamenta Informaticae, 72*, 379–391.

Stefanowski, J., & Wilk, S. (2007). Improving rule based classifiers induced by modlem by selective pre-processing of imbalanced data. In *Proceedings of the RSKD Workshop at ECML/PKDD* (pp. 54–65).

Stefanowski, J., & Wilk, S. (2008). Selective pre-processing of imbalanced data for improving classification performance. In *Proceedings of the 10th int. conf. DaWaK. LNCS* (Vol. 5182, pp. 283–292). Springer.

Stefanowski, J., & Wilk, S. (2009). Extending rule-based classifiers to improve recognition of imbalanced classes. In Z. Ras, & A. Dardzinska (Eds.), *Advances in data management. Studies in computational intelligence* (Vol. 223, pp. 131–154). Springer Berlin/Heidelberg.

Tan, P., Steinbach, M., & Kumar, V. (2005) Classification: Alternative techniques. In *Introduction to data mining* (pp. 207–223). Pearson Addison Wesley.

Ting, K. (1994). The problem of small disjuncts: Its remedy in decision trees. In *Proceeding of the 10th Canadian conference on artificial intelligence* (pp. 91–97).

Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2003) Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th Int. Conf. on ML (ICML)* (pp. 17–23).

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

Weiss, G. (2004). Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter, 6*(1), 7–19.

Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research, 19*, 315–354.

Wilk, S., Slowinski, R., Michalowski, W., & Greco, S. (2005). Supporting triage of children with abdominal pain in the emergency room. *European Journal of Operational Research, 160*, 696–709.

Wilson D. R., & Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research (JAIR), 6*, 1–34.

Yao, Y., & Zhong, N. (1999). An analysis of quantitative measures associated with rules. In *Proceedings of the 3rd Pacific-Asia conference on knowledge discovery and data mining. LNAI*, (Vol. 1574, pp. 479–488). Springer.

Zhang, J. (1997). A method that combines inductive learning with exemplar-based learning. In *Proceedings of the 2nd IEEE international conference on tools for artificial intelligence* (pp. 31–37). IEEE Computer Society Press.

Zhang, J., Bloedorn, E., Rosen, L., & Venese, D. (2004). Learning rules from highly unbalanced data sets. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM '04)* (pp. 571–574). IEEE Computer Society.

Zytkow, J. (2002). Types and forms of knowledge (patterns): Rules. In *Handbook of data mining and knowledge discovery* (pp. 51–54). Oxford University Press, Inc.