Chapter 3
Improving Bagging Ensembles for Class Imbalanced Data by Active Learning

Jerzy Błaszczyński and Jerzy Stefanowski

Abstract Extensions of under-sampling bagging ensemble classifiers for class imbalanced data are considered. We propose a two phase approach, called Actively Balanced Bagging, which aims to improve recognition of minority and majority classes with respect to so far proposed extensions of bagging. Its key idea consists in additional improving of an under-sampling bagging classifier (learned in the first phase) by updating in the second phase the bootstrap samples with a limited number of examples selected according to an active learning strategy. The results of an experimental evaluation of Actively Balanced Bagging show that this approach improves predictions of the two different baseline variants of under-sampling bagging. The other experiments demonstrate the differentiated influence of four active selection strategies on the final results and the role of tuning main parameters of the ensemble.

Keywords Class imbalance · Active learning · Bagging ensembles · Under-sampling

3.1 Introduction

Supervised learning of classifiers from class imbalanced data is still a challenging task in machine learning and pattern recognition. Class imbalanced data sets are characterized by uneven cardinalities of classes. One of the classes, usually called a *minority class* and being of key importance in a given problem, contains significantly less learning examples than other majority classes. Class imbalance occurs in many real-world application fields, such as: medical data analysis, fraud detection,

J. Błaszczyński (✉) · J. Stefanowski
Institute of Computing Science, Poznań University of Technology, Piotrowo 2,
60-965 Poznań, Poland
e-mail: jerzy.blaszczynski@cs.put.poznan.pl

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

© Springer International Publishing AG 2018

U. Stańczyk et al. (eds.), *Advances in Feature Selection for Data and Pattern Recognition*, Intelligent Systems Reference Library 138,
https://doi.org/10.1007/978-3-319-67588-6_3

technical diagnostics, image recognition or text categorization. More information about them can be found in [24, 54, 57].

If imbalance in the class distribution is severe, i.e., some classes are strongly under-represented, standard learning algorithms do not work properly. Constructed classifiers may have difficulties, in some cases they may be even completely unable, to classify correctly new instances from the minority class. Such behaviour have been demonstrated in several experimental studies such as [23, 29, 39].

Several approaches to improve classifiers for imbalanced data have been proposed [11, 25, 54]. They are usually categorized as: classifier-independent *pre-processing* methods or *modifications of algorithms* for learning particular classifiers. Methods within the first category try to re-balance the class distribution inside the training data by either adding examples to the minority class (*over-sampling*) or by removing examples from the majority class (*under-sampling*). The other category of algorithm level methods involves specific solutions dedicated to improving a given classifier. Specialized ensembles are among the most effective methods within this category [40].

Besides developing new approaches, some researchers attempt to better understand the nature of the imbalance data and key properties of its underlying distribution, which makes the class imbalanced problem difficult to be handled. They have shown, that so called, *data difficulty factors* hinder the learning performance of classification algorithms [22, 29, 41, 53]. The data difficulty factors are related to characteristics of class distribution, such as decomposition of the class into rare sub-concepts, overlapping between classes or presence of rare minority examples inside the majority class regions. It has been shown that some classifiers and data pre-processing methods are more sensitive to some of these difficulty factors than others [45, 52].

Napierała et al. have shown that several data difficulty factors may be approximated by analyzing the content of the minority example neighbourhood and modeling several types of data difficulties [45]. Moreover, in our previous works [6, 7] it has been observed, that neighbourhood analysis of minority examples may be used to change the distribution of examples in bootstrap samples of ensembles. The resulting extensions of bagging ensembles are cable to significantly improve classification performance on imbalanced data sets. The interest in studying extensions of bagging ensembles is justified by recent promising experimental results of their comparison against other classifiers dedicated to imbalanced data [6, 7, 33, 37].

Nevertheless, a research question could be posed, whether it is still possible to improve performance of these ensembles. In experimental studies, such as [5, 7, 37], it has been shown that the best proposals of extending bagging by under-sampling may improve the minority class recognition at the cost of strong decrease of recognition of majority class examples. We claim that it would be more beneficial to construct an ensemble providing a good trade-off between performance in both classes instead.

To address this research question we plan to consider a quite different perspective of extending bagging ensembles than it is present in the current solutions, which mainly modify the generation of bootstrap samples. Here, we propose instead a two phase approach. First, we start with construction of an ensemble classifier according to one of under-sampling extensions designed for imbalanced data. Then, we modify bootstrap samples, constructed in the first phase, by adding a limited number of learning examples, which are important to improve performance in both classes. To perform this kind of an example selection we follow inspiration coming from the *active learning* paradigm [2]. This type of learning is commonly used in the semi-supervised framework to update the classifier learned on labeled part of data by selecting the most informative examples from the pool of unlabeled ones. Active learning can also be considered to filter examples from the fully labeled data sets [2]. In this way, active strategies have been already applied to imbalanced learning, although these attempts are still quite limited, see Sect. 3.3.

In this chapter we will discuss a new perspective of using active learning to select examples while extending under-sampling bagging ensembles. We call the proposed extension *Actively Balanced Bagging* (ABBag) [8].

In the first phase of the approach, ABBag is constructed with previously proposed algorithms for generating under-sampling bagging extensions for imbalanced data. In the experiments we will consider two different efficient algorithms, namely Exactly Balanced Bagging (EBBag) [13], and Neighbourhood Balanced Bagging (NBBag) [7]. Then, in the second phase the ensemble classifier will be integrated with the active selection of examples. In ABBag this strategy exploits the decision margin of component classifiers in ensemble votes, which is more typical for the active learning. Since, contrary to typical active learning setting, we are dealing with fully labeled data, errors of component classifiers in ensemble will be taken into account as well. Moreover, following experiences from the previous research on data difficulty factors, the neighbourhood analysis of the examples will be also explored. All these elements could be integrated in different way, which leads us to consider four versions of the active selection strategies.

The preliminary idea of ABBag was presented in our earlier conference paper [8]. In this chapter, we discuss it in more details and put in the context of other related approaches. The next contributions include carrying out a comprehensive experimental study of ABBag usefulness and its comparison against the baseline versions of under-sampling extensions of bagging for imbalanced data. Furthermore, we experimentally study properties of ABBag with respect to different active selection strategies and tuning its parameters.

The chapter is organized as follows. The next section summarizes the most related research on improving classifiers learned from class imbalanced data. The following Sect. 3.3, discusses use of active learning in class imbalanced problems. Ensembles specialized for imbalanced data are described in Sect. 3.4. The *Actively Balanced Bagging* (ABBag) is presented in Sect. 3.5. The results of experimental evaluation of ABBag are given in Sect. 3.6. The final section draws conclusions.

3.2 Improving Classifiers Learned from Imbalanced Data

In this section we discuss concepts, which are the most related to our proposal. For more comprehensive reviews of specialized methods for class imbalanced data the reader could refer to, e.g., [11, 25, 54]. In this chapter, we consider only a typical binary definition of the class imbalance problem, where the selected minority class is distinguished from a single majority class. This formulation is justified by focusing our interest on the most important class and its real-life semantics [24, 54]. Recently some researchers study more complex scenarios with multiple minority classes, see e.g., reviews in [49, 56].

3.2.1 Nature of Imbalanced Data

In some problems characterized by high class imbalance, standard classifiers have been found to be accurate, see e.g., [3]. In particular, it has been found that, when there is a good separation (e.g., linear) between classes, the minority class may be sufficiently recognized regardless of the high *global imbalance ratio* between classes [46]. The global imbalance ratio is usually expressed as either $N_{min}{:}N_{maj}$ or $\frac{N_{min}}{N}$, where N_{maj}, N_{min}, N are the number of majority, minority, and total number of examples in the data set, respectively.

Some researches have shown that the global class imbalance ratio is not necessarily the only, or even the main, problem causing the decrease of classification performance [22, 32, 41, 42, 47, 51]. These researchers have drawn attention to other characteristics of example distributions in the attribute space called *data complexity* or *data difficulty factors*. Although these factors should affect learning also in balanced domains, when they occur *together* with class imbalance, then the deterioration of classification performance is amplified and affects mostly the minority class. The main data difficulty factors are: decomposition the minority class into rare sub-concepts, overlapping between classes, and presence of outliers, rare instances, or noise.

The influence of *class decomposition* has been noticed by Japkowicz et al. [29, 32]. They experimental showed that the degradation of classification performance has resulted from decomposing the minority class into many sub-parts containing very few examples, rather than from changing the global imbalance ratio. They have also argued that the minority class often does not form a compact homogeneous distribution of the single concept, but is scattered into many smaller sub-clusters surrounded by majority examples. Such sub-clusters are referred to *small disjuncts*, which are harder to learn and cause more classification errors than larger sub-concepts.

Other factors related to the class distribution are linked to high *overlapping* between regions of minority and majority class examples in the attribute space. This difficulty factor has already been recognized as particularly important for standard, balanced, classification problems, however, its role is more influential for the minority class. For instance, a series of experimental studies of popular classifiers on synthetic data have pointed out that increasing overlapping has been more influential than changing the class imbalance ratio [22, 47]. The authors of [22] have also shown that the local imbalance ratio inside the overlapping region is more influential than the global ratio.

Yet another data difficulty factor which causes degradation of classifier performance on imbalanced data is the presence of minority examples inside distributions of the majority class. Experiments presented in a study by Napierała et al. [42] have shown that single minority examples located inside the majority class regions cannot be always treated as noise since their proper treatment by informed pre-processing may lead to improvement of classifiers. In more recent papers [45, 46], they have distinguished between safe and unsafe examples. *Safe examples* are the ones located in homogeneous regions populated by examples from one class only. Other examples are *unsafe* and they are more difficult to learn from. Unsafe examples were further categorized into *borderline* (placed close to the decision boundary between classes), *rare cases* (isolated groups of few examples located deeper inside the opposite class), and *outliers*.

The same authors have introduced an approach [45] to automatically identify the aforementioned types of examples in real world data sets by analyzing class labels of examples in the local neighbourhood of a considered example. Depending on the number of examples from the majority class in the local neighbourhood of the given minority example, we can evaluate whether this example could be safe or unsafe (difficult) to be learned.

3.2.2 Evaluation of Classifiers on Imbalanced Data

Class imbalance constitutes difficulty not only during construction of a classifier but also when one evaluates classifier performance. The overall classification accuracy is not a good criterion characterizing classifier performance, in this type of problem, as it is dominated by the better recognition of the majority class which compensates the lower recognition of the minority class [30, 34]. Therefore, other measures defined for binary classification are considered, where typically the class label of the minority class is called positive and the class label of the majority class is negative. The performance of the classifiers is presented in a binary confusion matrix as in Table 3.1.

Table 3.1 Confusion matrix for the classifier evaluation

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

One may construct basic metrics concerning recognition of the positive (minority) and negative (majority) classes from the confusion matrix:

$$Sensitivity \;=\; Recall = \frac{TP}{TP+FN}, \tag{3.1}$$

$$Specificity = \frac{TN}{FP+TN}, \tag{3.2}$$

$$Precision = \frac{TP}{TP+FP}. \tag{3.3}$$

Some more elaborated measures may also be considered (please see e.g., overviews of the measures presented in [25, 30]).

As the class imbalance task invariably involves a trade off between false positives FP and false negatives FN, to control both, some single-class measures are commonly considered in pairs, e.g., $Sensitivity$ and $Specificity$ or $Sensitivity$ and $Precision$. These single-class measures are often aggregated to form further measures [28, 30]. The two admittedly most popular aggregations are the following:

$$\text{G-mean} = \sqrt{Sensitivity \cdot Specificity}, \tag{3.4}$$

$$\text{F-measure} = \frac{(1+\beta)\cdot Precision \cdot Recall}{\beta \cdot Precision + Recall}. \tag{3.5}$$

The F-measure combines $Recall$ ($Sensitivity$) and $Precision$ as a weighted harmonic mean, with the β parameter ($\beta > 0$) as the relative weight. It is most commonly used with $\beta = 1$. This measure is exclusively concerned with the positive (minority) class. Following inspiration from its original use in the information retrieval context, $Recall$ is a recognition rate of examples originally from the positive class while precision assesses to what extent the classifier was correct in classifying examples as positive that were actually positive. Unfortunately it is dependent to the class imbalance ratio.

The most popular alternative, G-mean, was introduced in [34] as a geometric mean of $Sensitivity$ and $Specificity$. It has a straightforward interpretation since it takes into account the relative balance of the classifier performance in both positive class and negative class. An important, useful property of the G-mean is that it is independent of the distribution of examples between classes. As both classes have equal importance in this formula, various further modifications to prioritize the positive class, like the adjusted geometric mean, have been postulated (for their overview see [30]).

The aforementioned measures are based on single point evaluation of classifiers with purely deterministic predictions. In case of scoring classifiers, several authors use the *ROC (Receiver Operating Characteristics) curve* analysis. The quality of the classifier performance is reflected by the area under a ROC curve (so called AUC measure). Alternative proposals include Precision Recall Curves or other special cost curves (see their review in [25, 30]).

3.2.3 Main Approaches to Improve Classifiers for Imbalanced Data

The class imbalance problem has received growing research interest in the last decade and several specialized methods have been proposed. Please see [11, 24, 25, 54] for reviews of these methods, which are usually categorized in two groups:

- Classifier-independent methods that rely on transforming the original data to change the distribution of classes, e.g., by re-sampling.
- Modifications of either a learning phase of the algorithm, classification strategies, construction of specialized ensembles or adaptation of cost sensitive learning.

The first group include data *pre-processing methods*. The simplest data pre-processing (re-sampling) techniques are: *random over-sampling*, which replicates examples from the minority class, and *random under-sampling*, which randomly eliminates examples from the majority classes until a required degree of balance between classes is reached. *Focused* (also called *informed*) *methods* attempt to take into account the internal characteristics of regions around minority class examples. Popular examples of such methods are: OSS [34], NCR [38], SMOTE [14] and some extensions these methods: see e.g., [11]. Moreover, some hybrid methods integrating over-sampling of selected minority class examples with removing the most harmful majority class examples have been also proposed, see e.g., SPIDER [42, 51].

The other group includes many quite specialized methods based on different principles. For instance, some authors changed search strategies, evaluation criteria or parameters in the internal optimization of the learning algorithm - see e.g., extensions of induction of decision tress with the Hellinger distance or the asymmetric entropy [16], or reformulation of the optimization task in generalized versions of SVM [24]. The final prediction technique can be also revised, for instance authors of [23] have modified conflict strategies with rules to give more chance for minority rules. Finally, other researchers adapt the imbalance problem to cost sensitive learning. For a more comprehensive discussion of various methods for modifying algorithm refer to [24, 25].

The neighbourhood analysis has been also used to modify pre-processing methods, see extensions of SMOTE or over-sampling [9], rule induction algorithm BARCID [43] or ensembles [7].

3.3 Active Learning

Active learning is a research paradigm in which the learning algorithm is able to select examples used for its training. Traditionally, this methodology has been applied interactively with respect to unlabeled data. Please refer to the following survey for a review of different active strategies in semi-supervised learning perspective [50]. The goal of active learning, in this traditional view, is to minimize costs, i.e., time, effort, and other resources related to inquiring for class labels needed to update / train classifier.

Nevertheless, active learning may also be applied when class labels are known. The goal is then to select the best examples for training. Such definition of a goal is particularly appealing to learning from imbalanced data, where one has a limited number of examples from the minority class and too high number of examples from the majority class. Thus, a specialized selection of the best examples from majority class may be solved by an active approach. The recent survey [11] clearly demonstrates an increasing interest in applying active learning strategies to imbalanced data.

In pool-based active learning, which is of our interest here, one starts with a given pool (i.e., a set) of examples. The classifier is first built on examples from the pool. Then one queries these examples outside the pool that are considered to be potentially the most useful to update the classifier. The main problem for active learning strategy is computing the *utility* of examples outside the pool. Various definitions of *utility* have already been considered in the literature [48]. Uncertainty sampling and query-by-committee are the two most frequently applied solutions.

Uncertainty sampling queries examples one by one, at each step, selecting the one for which the current classifier is the most uncertain while predicting the class. For imbalanced data, it has been applied together with support vector machines (SVM) classifiers. In such a case, uncertainty is defined simply as a distance to the margin of SVM classifier. Ertkin et al. have started this direction and proposed an active learning with early stopping with online SVM [17]. These authors have also considered an adaptive over-sampling algorithm VIRTUAL, which is able to generate synthetic minority class examples [18]. Another method, also based on uncertainty sampling, has been proposed by Zięba and Tomczak. This proposal consists in an ensemble of boosted SVMs. Base SVM classifiers are trained iteratively on examples identified by an extended margin created in previous iteration [60].

Query by committee (QBC) [1], on the other hand, queries examples, again, one by one, at each step selecting the one for which a committee of classifiers disagrees the most. The committee may be formed in different ways, e.g., by sampling hypotheses from the version space, or through bagging ensembles [48]. Yang and Ma have proposed a random subspace ensemble for class imbalance problem that makes use of QBC [59]. More precisely, they calculate the margin between two highest membership probabilities for the two most likely classes predicted by the ensemble.

The idea of QBC have also been considered by Napierala and Stefanowski in argument based rule learning for imbalanced data, where it selects the most difficult examples to be annotated [44]. The annotated examples are further handled in generalized rule induction. The experimental results of [44] show that this approach significantly improved recognition of both classes (minority and majority) in particular for rare cases and outliers.

Other strategies to compute utility of examples were also considered. For example, Certainty-Based Active Learning (CBAL) algorithm has been proposed for imbalanced data [20]. In CBAL, neighbourhoods are explored incrementally to select examples for training. The importance of an example is measured within the neighbourhood. In this way, certain, and uncertain areas are constructed and then used to select the best example. A hybrid algorithm has been also proposed for on-line active learning with imbalanced classes [19]. This algorithm switches between different selection strategies: uncertainty, density, certainty, and sparsity.

All of the algorithms mentioned this far query only one example at time. However, querying more examples, in a batch, may reduce the labeling effort and computation time. One does not need to rebuild the classifier after each query. On the other hand, batch querying introduces additional challenges, like diversity of batch [10]. To best of our knowledge no batch querying active learning algorithm has been proposed for class imbalanced data.

3.4 Ensembles Specialized for Imbalanced Data

Specialized extensions of ensembles of classifiers are among the most efficient currently known approaches to improve recognition of the minority class in imbalanced setting. These extensions may be categorized differently. The taxonomy proposed by Galar et al. in [21] distinguishes between *cost-sensitive* approaches vs. integrations with *data pre-processing*. The first group covers mainly cost-minimizing techniques combined with boosting ensembles, e.g., AdaCost, AdaC or RareBoost. The second group of approaches is divided into three sub-categories: Boosting-based, Bagging-based or Hybrid depending on the type of ensemble technique which is integrated into the schema for learning component classifiers and their aggregation. Liu et al. categorize the ensembles for class imbalance into bagging-like, boosting-based methods or hybrid ensembles depending on their relation to standard approaches [40].

Since the most of related works [4, 6, 21, 33, 36] indicate superior performance of bagging extensions versus the other types ensembles (e.g., boosting), we focus our consideration, in this study, on bagging ensembles.

Bagging [12] classifier, proposed by Breiman, is an ensemble of m_{bag} base (component) classifiers constructed by the same algorithm from m_{bag} *bootstrap samples* drawn from the original training set. The predictions of component classifiers are combined to form the final decision as the result of the equal weight majority voting. The key concept in bagging is *bootstrap* aggregation, where the training set, called a bootstrap, for each component classifier is constructed by random uniform sampling examples from the original training set. Usually the size of each bootstrap is equal to the size of the original training set and examples are drawn with replacement.

Algorithm 3.1: Bagging scheme

```
Input  : LS training set; TS testing set; CLA learning algorithm;
         $m_{bag}$ number of bootstrap samples;
Output: $C^*$ final classifier
Learning phase;
for $i := 1$ to $m_{bag}$ do
    $S_i$ := bootstrap sample {sample examples with replacement} ;
    $C_i$ := CLA ($S_i$) {generate a component classifier} ;
end
Classification phase;
foreach $y$ in TS do
    $C^*(x)$ := combination of predictions $C_i(x)$, where $i = 1, \ldots, m_{bag}$
        {prediction for example $x$ results from majority voting $C_i$} ;
end
```

Since bootstrap sampling, in the standard version, will not change drastically the class distribution in constructed bootstrap samples, they will be still biased toward the majority class. Thus, most of proposals to adapt/extend bagging to class imbalance overcome this drawback by applying pre-processing techniques, which change the balance between classes in each bootstrap sample. Usually they construct bootstrap samples with the same, or similar, cardinalities of both minority and majority classes.

In *under-sampling* bagging approaches the number of the majority class examples in each bootstrap sample is randomly reduced to the cardinality of the minority class (N_{min}). In the simplest proposal, called *Exactly Balanced Bagging* (EBBag), while constructing training bootstrap sample, the entire minority class is copied and combined with randomly chosen subsets of the majority class to exactly balance cardinalities between classes [13].

While such under-sampling bagging strategies seem to be intuitive and work efficiently in some studies, Hido et al. [26] observed that they do not truly reflect the philosophy of bagging and could be still improved. In the original bagging the class distribution of each sampled subset varies according to the binomial distribution while in the above under-sampling bagging strategy each subset has the same class ratio as the desired balanced distribution. In *Roughly Balanced Bagging* (RBBag) the numbers of instances for both classes are determined in a different way by equalizing the sampling probability for each class. The number of minority examples (S_{min}) in each bootstrap is set to the size of the minority class N_{min} in the original data. In contrast, the number of majority examples is decided probabilistically according to the negative binomial distribution, whose parameters are the number of minority examples (N_{min}) and the probability of success equal to 0.5. In this approach only the size of the majority examples (S_{maj}) varies, and the number of examples in the minority class is kept constant since it is small. Finally, component classifiers are induced by the same learning algorithm from each i-th bootstrap sample ($S^i_{min} \cup S^i_{maj}$) and their predictions form the final decision with the equal weight majority voting.

Yet another approach has been considered in *Neighbourhood Balanced Bagging* (NBBag), which is based on different principles than aforementioned under-sampling bagging ensembles. Instead of using uniform sampling, in NBBag, probability of an example being drawn into the bootstrap is modified according to the class distribution in his neighbourhood [7]. NBBag shifts sampling probability toward unsafe examples located in difficult to learn sub-regions of the minority class. To perform this type of sampling weights are assigned to the learning examples. The weight of minority example is defined as: $w = 0.5 \times \left(\frac{(N'_{min})^{\psi}}{k} + 1\right)$ where N'_{min} is the number of majority examples among k nearest neighbours of the example, and ψ is a scaling factor. Setting $\psi = 1$ causes a linear amplification of an example weight together with an increase of unsafeness, and setting ψ to values higher than 1 results in an exponential amplification. Each majority example is assigned a constant weight $w = 0.5 \times \frac{N_{maj}}{N_{min}}$, where N_{maj} is the number of majority class examples in the training set and N_{min} is the number of minority class examples in the training set. Then sampling is performed according to the distribution of weights. In this sampling, probability of an example being drawn to the bootstrap sample is reflected by its weight.

Another way to overcome class imbalance in a bootstrap sample consists in performing *over-sampling* of the minority class before training a component classifier. In this way, the number of minority examples is increased in each sample (e.g., by a random replication), while the majority class is not reduced as in under-sampling bagging. This idea was realized in many ways as authors considered several kinds of integrations with different over-sampling techniques. Some of these ways are also focused on increasing diversity of bootstrap samples. *OverBagging* is the simplest version which applies a simplest random over-sampling to transform each training bootstrap sample. S_{maj} of minority class examples is sampled with replacement to exactly balance the cardinality of the minority and the majority class in each sample. Majority examples are sampled with replacement as in the original bagging. An over-sampling variant of Neighbourhood Balanced Bagging (NBBag) has also been proposed [7]. In this variant, weights of examples are calculated in the same way as for under-sampling NBBag.

Finally, Lango et al. have proposed to integrate a random selection of attributes (following inspirations of [27, 35]) into Roughly Balanced Bagging [36]. Then the same authors have introduced a generalization of RBBag for multiple imbalanced classes, which exploits the multinomial distribution to estimate cardinalities of class examples in bootstrap samples [37].

3.5 Active Selection of Examples in Under-Sampling Bagging

Although a number of interesting under-sampling extensions of bagging ensembles, for class imbalanced data, have been recently proposed, the prediction improvement brought by these extensions may come with a decrease of recognition of majority class examples. Thus, we identify a need for better learning a trade-of between performance in both classes. Then an open research problem is how to achieve this balance of performance in both classes.

In this study we want to take a different perspective than in current proposals. These proposals mainly make use of various modifications of sampling examples to bootstraps (usually oriented toward balancing bootstrap) and then construct an ensemble, in a standard way, by learning component classifiers in one step and aggregating their predictions according to the majority voting (please see details of bagging and its variants in [12, 35]).

More precisely, we want to consider another hypothesis: given an already good technique of constructing under-sampling bagging, could one perform an additional step of updating its bootstraps by selecting a limited number of remaining learning examples, which could be useful for improving the trade-off between recognizing minority and majority classes.

Our proposed approach, called *Actively Balanced Bagging* (ABBag) [8], is composed of two phases. The first phase consists in learning an ensemble classifier by one of approaches for constructing under-sampling extensions of bagging. Although one can choose any good performing extension, we will further consider quite simple, yet effective one: Exactly Balanced Bagging (EBBag) [21], and more complex one based on other principles: Neighbourhood Balanced Bagging NBBag [7]. The current literature, such as [7, 33, 36], contains several experimental studies, which have clearly demonstrated that both these ensembles, and Roughly Balanced Bagging [26], are the best ensembles and they also out-performed single classifiers for difficult imbalanced data. Furthermore their modifications of sampling examples are based on completely different principles which is an additional argument to better verify the usefulness of the proposed active selection strategies in ABBag. For more information on constructing the EBBag and NBBag ensembles the reader may refer to Sect. 3.4.

The second phase includes an *active selection of examples*. It includes:

1. An iterative modification of bootstrap samples, constructed in the first phase, by adding selected examples from the training set;
2. Re-learning of component classifiers on modified bootstraps. The examples selected in (1) are added to bootstraps in *batches*, i.e., small portions of learning examples.

The proposed active selection of examples can be seen as a variant of Query-by-committee (QBC) approach [1]. As discussed in the previous sections QBC uses a decision margin, or simply a measure of disagreement between members of the committee, to select the examples. Although QBC has been already successfully applied in active learning of ensemble classifiers in [10]. It has been observed that QBC does not take into account global (i.e., concerning the whole training set) properties of examples distribution, and in result, it can focus too much on selecting outliers and sparse regions [10]. Therefore, we need to adapt this strategy for imbalanced data, which are commonly affected by data difficulty factors.

Furthermore, one should remember that selecting one single example at a time is a standard strategy in active learning [50]. Contrary, in ABBag we promote, in each iteration, to select small batches of examples instead of single example. We motivate the batch selection by a potential reduction of computation time, as well as, an increase of diversity of examples in the batch. As it was observed in [10], a greedy selection of single example with respect to a single criterion, typical for active strategy, where highest utility/uncertainty measure is taken into account [50] does not provide desired diversity. In our view, giving chance for random drawing also some slightly sub-optimal examples besides the best ones may result in a higher diversity of new bootstraps and increased diversity of re-learned component classifiers.

We address the above mentioned issues twofold. First, and foremost, the proposed active selection of examples considers multiple factors to determine the usefulness of an example to be selected. More precisely, they are following:

1. Decision margin of component classifiers, and a prediction error of the single component classifier (which is a modification of QBC).
2. Factors specific to imbalanced data, which reflect more global (i.e., concerning the whole training set) and/or local (i.e., concerning example neighbourhood) class distribution of examples.
3. Additionally we use a specific variant of rejection sampling to enforce diversity within the batch through extra randomization.

The algorithm for learning ABBag ensemble is presented as a pseudocode Algorithm 3.2. It starts with training set LS, and m_{bag} bootstrap samples $\boldsymbol{S}$ and results in constructing an under-sampling extension of bagging in the first phase (lines 2–4). Moreover, it makes use of initial balancing weights $\boldsymbol{w}$, which are calculated in accordance with the under-sampling bagging extension, used in this phase. These initial *balancing weights* $\boldsymbol{w}$ allow us to direct sampling toward more difficult to learn examples. In case of EBBag, balancing weights $\boldsymbol{w}$ reflect only the global imbalance of an example in the training set. In case of NBBag, balancing weights $\boldsymbol{w}$ expresses both global and local imbalance of an example in the training set. In the end of the first phase, component classifiers are generated from each of bootstraps $\boldsymbol{S}$ (line 3).

In the second phase, the active selection of examples is performed between lines 5–13. All bootstraps from $\boldsymbol{S}$ are iteratively (m_{al} times) enlarged by adding batches, and new component classifiers are re-learned.

In each iteration, new weights $\boldsymbol{w'}$ of examples are calculated according to weights update method um (which is described in the next paragraph), and then they are sorted (lines 7–8). Each bootstrap is enhanced by n_{al} examples selected randomly with the rejection sampling according to $\alpha = w'(x_{n_{al}}) + \varepsilon$, i.e., n_{al} random examples with weights w' higher than α are selected (lines 9–10). The parameter ε introduces additional (after α) level of randomness into the sampling. Finally, new component classifier C_i is learned resulting in new ensemble classifier $\boldsymbol{C}$ (line 11).

We consider here four different weights update methods. The simplest method, called *margin* (m),[1] is substituting the initial weights of examples with a decision margin between component classifiers in $\boldsymbol{C}$. For a given testing example it is defined

[1] For simplicity margin will be denoted as m - in particular in experiments see Tables 3.3, 3.4, 3.5 and 3.6; further introduced weight update methods will be denoted analogously.

Algorithm 3.2: Actively Balanced Bagging Algorithm

```
Input  : LS training set; TS testing set; CLA component classifier learning algorithm; $m_{bag}$
         number of bootstrap samples; $\boldsymbol{S}$ bootstrap samples; $\boldsymbol{w}$ weights of examples from
         LS; um weights update method; $m_{al}$ number of active learning iterations; $n_{al}$
         maximum size of active learning batch
Output: $\boldsymbol{C}$ ensemble classifier

Learning phase;
for $i := 1$ to $m_{bag}$ do
    $C_i$ := CLA ($S_i$) {generate a component classifier} ;
end
for $l := 1$ to $m_{al}$ do
    for $i := 1$ to $m_{bag}$ do
        $\boldsymbol{w'}$ := updateWeights($\boldsymbol{w}$,$\boldsymbol{C}$,um) {update weights used in sampling} ;
        sort all $x$ with respect to $w'(x)$, so that $w'(x_1) \geq w'(x_2) \geq \ldots \geq w'(x_n)$ ;
        $S_i'$ := random sample from $x_1, x_2, \ldots, x_{n_{al}}$ according to $\boldsymbol{w'}$ {rejection sampling from
            top $n_{al}$ $x$ sorted according to $\boldsymbol{w'}$; $\alpha = w'(x_{n_{al}})$ } ;
        $S_i := S_i \cup S_i'$ ;
        $C_i$ := CLA ($S_i$) {re-train a new component classifier} ;
    end
end

Classification phase;
foreach $x$ in TS do
    $\boldsymbol{C}(x)$ := majority vote of $C_i(x)$, where $i = 1, \ldots, m_{bag}$ {the class
    assignment for object $x$ is a combination of predictions of component classifiers $C_i$} ;
end
```

as: *margin* or $m = 1 - \left|\frac{V_{maj}-V_{min}}{m_{bag}}\right|$, where V_{maj} is a number of votes for majority class and V_{min} is number of votes for minority class. As the margin may not be directly reflecting the characteristic of imbalanced data (indeed under-sampling somehow should reduce bias of the classifiers) we consider combining it with additional factors. This leads to three variants of weights update methods. In the first extension, called, *margin and weight* (mw), new weight w' is a product of margin m and initial balancing weight w. We reduce the influence of w in subsequent iterations of active example selection, as l is increasing. The reason for this reduction of influence is that we expect margin m to improve (i.e., better reflect the usefulness of examples) with subsequent iterations, and thus initial weights w becoming less important. More precisely, $mw = m \times w^{\left(\frac{m_{al}-l}{m_{al}}\right)}$.

Both considered so far weights update methods produce bootstrap samples which, in the same iteration l, differ only according to randomization introduced by the rejection sampling, i.e., weights $\boldsymbol{w'}$ are the same for each i. That is why, we consider yet another modification of methods m and mw, which makes $\boldsymbol{w'}$, and, consequently, each bootstrap dependent on performance of the corresponding component classifier. These two new update methods: *margin and component error* (mce), and *margin, weight and component error* ($mwce$) are defined, respectfully, as follows: $mce = m + 1_e \times w$, and $mwce = mw + 1_e \times w$. In this notation, 1_e is an indicator function

Table 3.2 Characteristics of benchmark real-world data sets

Data set	# examples	# attributes	Minority class	IR [%]
abalone	4177	8	0-4 16-29	11.47
breast-cancer	286	9	recurrence-events	2.36
car	1728	6	good	24.04
cleveland	303	13	3	7.66
cmc	1473	9	2	3.42
ecoli	336	7	imU	8.60
haberman	306	4	2	2.78
hepatitis	155	19	1	3.84
scrotal-pain	201	13	positive	2.41
solar-flare	1066	12	f	23.79
transfusion	748	4	1	3.20
vehicle	846	18	van	3.25
yeast	1484	8	ME2	28.10

defined so that $1_e = 1$ when a component classifier is making a prediction error on example, and $1_e = 0$ otherwise.

3.6 Experimental Evaluation

In this section we will carry out experiments designed to provide better understanding of the classification performance of Actively Balanced Bagging. The following two aims of these experiments are considered. First, we want to check to what extent the predictive performance of Actively Balanced Bagging can be improved in comparison to under-sampling extensions of bagging. For this part of experiments we choose two quite efficient, in classification performance, extensions of bagging: Exactly Balanced Bagging (EBBag) [13], and Neighbourhood Balanced Bagging (NBBag) [7]. Then, the second aim of experiments is to compare different variants of proposed active selection methods, which result in different versions of ABBag. Moreover, the sensitivity analysis of tuning basic parameters of ABBag is carried out.

3.6.1 Experimental Setup

The considered Actively Balanced Bagging ensembles are evaluated with respect to averaged performance in both minority and majority classes. That is why we consider G-mean measure, introduced in Sect. 3.2.2, since we want to find a good trade-off between recognition in both classes.

Similarly to our previous studies [6–8, 45] we will focus in our experiments on 13 benchmark real-world class imbalanced data sets. In this way, we include in this study data sets which have been often analyzed in many experimental studies with imbalanced data. This should make it easier to compare the achieved performances to the best results reported in the literature. The characteristics is of these data sets are presented in Table 3.2. The data sets represent different sizes, imbalance ratios (denoted by IR), domains and have both continuous and nominal attributes. Taking into account results presented in [45] some of data sets should be easier to learn for standard classifiers while most of them constitute different degrees of difficulties. More precisely, such data as `vehicle` and `car`, are easier ones as many minority class examples may categorized as safe ones. On the other hand, data sets `breast cancer`, `clevaland`, `ecoli` contain many borderline examples, while the remaining data sets could be estimated as the most difficult one as they additionally contain many rare cases or outliers.

Nearly all of benchmark real-world data sets were taken from the UCI repository.[2] One data set includes a medical problem and it was also used in our earlier works of on class imbalance.[3] In data sets with more than one majority class, they are aggregated into one class to have only binary problems, which is also typically done in other studies presented in the literature.

Furthermore, we include in this study a few synthetic data sets with a priori known (i.e., designed) data distribution. To this end, we applied a specialized generator for imbalanced data [58] and we produced two different types of data sets. In these data sets, examples from the minority class are generated randomly inside predefined spheres and majority class examples are randomly distributed in an area surrounding them. We consider two configurations of minority class spheres, called according to the shape they form: `paw` and `flower`, respectively. In both data sets the global imbalance ratio IR is equal to 7, and the total cardinality of examples are 1200 for `paw` and 1500 for `flower` always with three attributes. The minority class is decomposed into 3 or 5 sub-parts. Moreover, each of this data set has been generated with a different number of potentially unsafe examples. This fact is denoted by four numbers included in the name of data set. For instance, `flower5-3d-30-40-15-15` represents `flower` with minority class that contains approximately 30% of safe examples, 40% inside the class overlapping (i.e., boundary), 15% rare and 15% outliers.

3.6.2 Results of Experiments

We conducted our experiments in two variants of constructing ensembles. In the first variant, standard EBBag or under-sampling NBBag was used. In the second variant,

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

[3] We are grateful to prof. W. Michalowski and the MET Research Group from the University of Ottawa for providing us an access to scrotal-pain data set.

the size of each of the classes in bootstrap samples was further reduced to 50% of the size of the minority class in the training set. Active selection parameters, used in the second phase, m_{al}, and n_{al} were chosen in a way, which enables the bootstrap samples constructed in ABBag to excess the size of standard under-sampling bootstrap by a factor not higher than two. The size of ensembles m_{bag}, in accordance with previous experiments [7], was always fixed to 50. We used WEKA[4] implementation of `J48` decision tree as the component classifier in all of considered ensembles. We set under-sampling NBBag parameters to the same values as we have already used in [7]. All measures are estimated by a stratified 10-fold cross-validation repeated five times to improve repeatability of observed results.

In Tables 3.3, 3.4, 3.5 and 3.6, we present values of G-mean for all considered variants of ABBag on all considered real-world and synthetic data sets. Note that in Tables 3.3 and 3.4, we present results of active balancing of 50% under-sampling EBBag, and 50% under-sampling NBBag, respectively. Moreover, in Tables 3.5 and 3.6, we present results of active balancing of standard under-sampling EBBag, and standard under-sampling NBBag, respectively. The last row of each of Tables 3.3, 3.4, 3.5 and 3.6, contains average ranks calculated as in the Friedman test [30]. The interpretation of average rank is that the lower the value, the better the classifier.

The first, general conclusion resulting from our experiments is that ABBag performs better than under-sampling extensions of bagging, both: EBBag, and NBBag. Let us treat EBBag, and NBBag as baselines in Tables 3.3, 3.4, 3.5 and 3.6, respectively. The observed improvements of G-mean are statistically significant regardless of the considered version of ABBag. More precisely, each actively balanced EBBag has the lower average rank than the baseline EBBag, and, similarly, each actively balanced NBBag has lower average rank than the baseline NBBag. Moreover, Friedman tests result in p-values $\ll 0.00001$ in all of comparisons of both EBBag (Tables 3.3 and 3.5) NBBag (Tables 3.4 and 3.6). According to Nemenyi post-hoc test, critical difference CD between average ranks in our comparison is around 1.272. The observed difference between average ranks of each actively balanced EBBag and the baseline EBBag is thus higher than calculated CD. We can state that ABBag improves significantly classification performance over base line EBBag. An analogous observation holds for each actively balanced NBBag and the baseline NBBag. We can conclude this part of experiments by stating that ABBag is able to improve baseline classifier regardless of the setting.

However, th observed improvement of G-mean introduced by ABBag depends on the data set. Usually improvements for easier to learn data sets, like `car`, are smaller than these observed for the other data sets. The most apparent (and consistent) improvements are noted for the hardest to learn versions of synthetic data sets. In both cases of `flower5-3d-10-20-35-35`, and `paw3-3d-10-20-35-35` application of baseline versions of EBBag, and NBBag gives values of G-mean equal to 0. These results are remarkably improved by all considered versions of ABBag.

[4] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Table 3.3 G-mean of actively balanced 50% under-sampling EBBag

Data set	EBBag	*m*-EBBag	*mce*-EBBag	*mw*-EBBag	*mwce*-EBBag
abalone	79.486	79.486	80.056	79.486	79.603
breast-cancer	57.144	59.628	60.471	60.640	60.900
car	96.513	97.785	98.359	97.806	98.359
cleveland	70.818	73.154	70.818	73.672	70.818
cmc	64.203	65.146	64.572	64.771	64.687
ecoli	87.836	88.870	89.259	88.926	88.638
flower5-3d-10-20-35-35	0.000	55.055	54.251	52.653	54.046
flower5-3d-100-0-0-0	92.315	93.415	94.570	93.501	94.812
flower5-3d-30-40-15-15	77.248	77.995	78.502	78.022	78.423
flower5-3d-30-70-0-0	91.105	91.764	93.729	92.019	93.993
flower5-3d-50-50-0-0	91.966	92.470	93.972	92.317	93.834
haberman	62.908	65.355	65.916	67.299	65.520
hepatitis	78.561	79.132	80.125	79.208	80.079
paw3-3d-10-20-35-35	0.000	51.152	51.148	52.318	50.836
paw3-3d-100-0-0-0	90.857	93.020	94.001	93.011	94.391
paw3-3d-30-40-15-15	74.872	76.277	78.241	76.546	77.544
paw3-3d-30-70-0-0	88.545	90.510	91.410	90.927	91.106
paw3-3d-50-50-0-0	91.424	92.087	92.825	92.038	93.537
scrotal-pain	72.838	73.572	73.574	73.692	72.838
solar-flare	82.048	83.126	83.064	83.013	83.064
transfusion	66.812	67.929	66.812	67.448	66.812
vehicle	95.506	95.840	97.120	96.010	97.120
yeast	82.658	84.026	84.818	85.337	84.984
average rank	4.848	3.087	2.065	2.652	2.348

Now we move to examination of the influence of the proposed active modifications, i.e., weights update methods in the active selection of examples, on the classification performance. We make the following observations. First, if we consider actively balanced 50% under-sampling EBBag, *margin and component error* weights update method, thus (*mce*-EBBag), has the best average rank and the best value of median calculated for all G-mean results in Table 3.3. The second best performing weight update method in this comparison is *margin, weight and component error*, and thus (*mwce*-EBBag). These observations are, however, not statistically significant according to the critical difference and results of Wilcoxon test for a selected pair of classifiers. Results of actively balanced standard under-sampling EBBag, presented in Table 3.5, provide more considerable distinction. The best weights update method in this case is *margin, weight and component error*, and thus (*mwce*-EBBag). Moreover the observed difference in both average rank and p-value in Wilcoxon test allows us to state that *mwce*-EBBag is significantly better performing than *m*-EBBag.

Table 3.4 G-mean of actively balanced 50% under-sampling NBBag

Data set	NBBag	*m*-NBBag	*mce*-NBBag	*mw*-NBBag	*mwce*-NBBag
abalone	78.297	79.384	79.264	79.034	79.339
breast-cancer	56.521	62.559	61.957	60.106	62.316
car	93.918	95.698	97.816	97.405	98.182
cleveland	74.275	76.131	75.000	78.000	74.275
cmc	63.944	64.969	64.390	64.969	64.807
ecoli	88.056	89.326	88.846	89.412	89.139
flower5-3d-10-20-35-35	0.000	51.940	51.302	52.069	52.105
flower5-3d-100-0-0-0	89.851	92.844	94.465	93.427	94.869
flower5-3d-30-40-15-15	73.869	77.081	77.266	76.614	77.484
flower5-3d-30-70-0-0	88.513	92.055	93.814	91.903	93.975
flower5-3d-50-50-0-0	89.164	92.227	93.969	91.734	94.107
haberman	58.618	65.165	65.087	65.068	65.386
hepatitis	79.632	79.632	80.449	80.778	80.270
paw3-3d-10-20-35-35	0.000	50.644	49.804	43.549	52.541
paw3-3d-100-0-0-0	85.165	91.889	94.000	92.596	93.584
paw3-3d-30-40-15-15	29.157	75.499	75.000	74.818	70.950
paw3-3d-30-70-0-0	82.767	89.944	91.631	90.739	91.331
paw3-3d-50-50-0-0	84.787	91.327	92.079	91.826	92.474
scrotal-pain	74.471	75.180	75.625	76.507	75.363
solar-flare	82.275	85.049	83.620	83.954	83.233
transfusion	65.259	65.816	65.351	65.259	65.509
vehicle	94.304	96.720	97.733	96.513	97.498
yeast	82.450	84.454	84.887	83.549	85.005
average rank	4.935	2.652	2.435	2.957	2.022

Second best performing weight update method in this case is *margin and component error* weights update method, thus (*mce*-EBBag). Differences between this method and the other weights update methods are, however, again not statistically significant.

Similar observations are valid for actively balanced NBBag. In this case, the best weights update method according to average rank is *margin, weight and component error*, and thus (*mwce*-NBBag), regardless of the variant of the fist phase of the active selection, as seen in Tables 3.4 and 3.6. On the other hand, *margin and component error* weights update method, and thus (*mce*-NBBag), has the best value of median when 50% under-sampling *mce*-NBBag is considered. However, this observation is not statistically significant. So are all the observed differences among different weights update methods in active selection for NBBag.

To sum up observations noted so far, we should report that all different factors that we take into account in weights update methods: margin of classifiers in ensemble, weight of example, and component error are important for improving ABBag

Table 3.5 G-mean of actively balanced under-sampling EBBag

Data set	EBBag	*m*-EBBag	*mce*-EBBag	*mw*-EBBag	*mwce*-EBBag
abalone	76.927	77.740	77.940	77.740	77.874
breast-cancer	57.979	58.488	58.316	58.632	59.305
car	97.611	97.760	98.100	97.780	98.185
cleveland	68.207	68.207	68.207	68.207	68.207
cmc	62.242	62.552	62.242	62.280	62.242
ecoli	87.677	87.677	87.677	87.677	87.677
flower5-3d-10-20-35-35	0.000	43.713	54.854	51.077	54.419
flower5-3d-100-0-0-0	92.601	93.506	94.860	93.399	95.007
flower5-3d-30-40-15-15	77.749	78.027	78.654	77.977	78.979
flower5-3d-30-70-0-0	91.614	92.165	93.629	91.994	93.544
flower5-3d-50-50-0-0	91.338	92.651	94.622	92.451	94.469
haberman	60.673	64.379	64.327	65.124	65.251
hepatitis	74.217	77.505	78.631	76.937	76.688
paw3-3d-10-20-35-35	0.000	41.322	53.729	37.655	52.275
paw3-3d-100-0-0-0	92.507	93.811	94.233	93.765	94.329
paw3-3d-30-40-15-15	76.756	76.756	78.846	76.756	78.839
paw3-3d-30-70-0-0	89.362	90.206	91.069	91.443	90.990
paw3-3d-50-50-0-0	92.107	92.107	92.445	92.107	92.757
scrotal-pain	74.258	74.258	74.258	74.549	74.258
solar-flare	84.444	84.444	84.444	84.444	84.654
transfusion	64.078	65.492	67.534	65.911	67.589
vehicle	95.117	96.327	96.771	96.476	96.849
yeast	81.689	82.248	84.234	83.933	84.541
average rank	4.565	3.283	2.174	3.087	1.891

performance. Moreover, two weight update methods tend to give better results than the others. These are: *margin and component error* (*mce*), and *margin, weight and component error* (*mwce*). These results may be interpreted as an indication that proposed active selection of examples is able to make good use of the known labels of minority and majority examples, since weight, and component error factors are important.

In the next step of our experiments, we tested the influence of ε parameter on the performance of ABBag. ε controls the level of randomness in active selection of examples (see Sect. 3.3 for details). The results of this analysis favour small values of ε, which means that, in our setting, it is better to select the best (or almost the best, to be more precise) examples into active learning batches. This result may be partially explained by a relatively small size of batches used in the experimental evaluation. For small batches it should be important to select as good examples as possible

Table 3.6 G-mean of actively balanced under-sampling NBBag

Data set	NBBag	m-NBBag	mce-NBBag	mw-NBBag	$mwce$-NBBag
abalone	78.714	79.308	79.291	79.317	79.460
breast-cancer	58.691	62.752	62.698	62.191	62.501
car	96.200	97.518	97.847	97.775	98.801
cleveland	73.004	73.004	73.931	74.170	74.776
cmc	65.128	65.128	65.128	65.365	65.128
ecoli	88.581	88.581	88.581	88.581	88.581
flower5-3d-10-20-35-35	0.000	51.952	51.527	52.800	51.073
flower5-3d-100-0-0-0	92.373	93.594	94.481	93.437	94.683
flower5-3d-30-40-15-15	76.914	78.080	77.913	77.570	78.196
flower5-3d-30-70-0-0	91.120	92.297	93.490	92.141	94.112
flower5-3d-50-50-0-0	92.003	93.126	93.209	92.889	94.322
haberman	64.128	65.101	65.251	66.059	65.590
hepatitis	78.017	78.078	79.665	79.269	80.739
paw3-3d-10-20-35-35	0.000	50.916	49.912	51.239	52.142
paw3-3d-100-0-0-0	90.122	92.792	93.141	93.190	94.388
paw3-3d-30-40-15-15	63.966	76.440	75.945	76.990	77.057
paw3-3d-30-70-0-0	87.208	90.072	90.966	90.871	91.116
paw3-3d-50-50-0-0	91.317	92.105	92.295	91.582	92.984
scrotal-pain	73.205	75.023	74.812	75.636	74.891
solar-flare	83.435	84.731	83.574	84.015	84.286
transfusion	65.226	65.943	66.239	65.226	65.226
vehicle	95.339	96.776	97.463	96.759	97.401
yeast	84.226	84.780	85.580	85.067	85.247
average rank	4.783	3.000	2.630	2.783	1.804

while with an increase of the size of batch, more diversity in batches resulting from selection of more sub-optimal examples, should be also found to be useful.

We finish the presented experimental evaluation with an analysis of influence of parameter m_{al}, and parameter n_{al} on the classification performance of ABBag. Parameter n_{al} is, in the results presented further on, the size of active learning batch determined as the percentage of the size of minority class. Parameter m_{al} is the number of performed active learning iterations. We restrict ourselves in this analysis to *margin, weight and component error* ($mwce$) weight update method, since it gave the best results for majority of considered variants. Moreover, we will only present results of an analysis of some harder to learn data sets since they provide more information about behaviour of ABBag. Results for other data sets demonstrate usually the same tendencies as for the chosen data sets, though these tendencies might be less visible. We present changes in values of G-mean for four different data sets: `cleveland`, `flower5-3d-10-20-35-35`, `paw3-3d-10-20-35-35`,

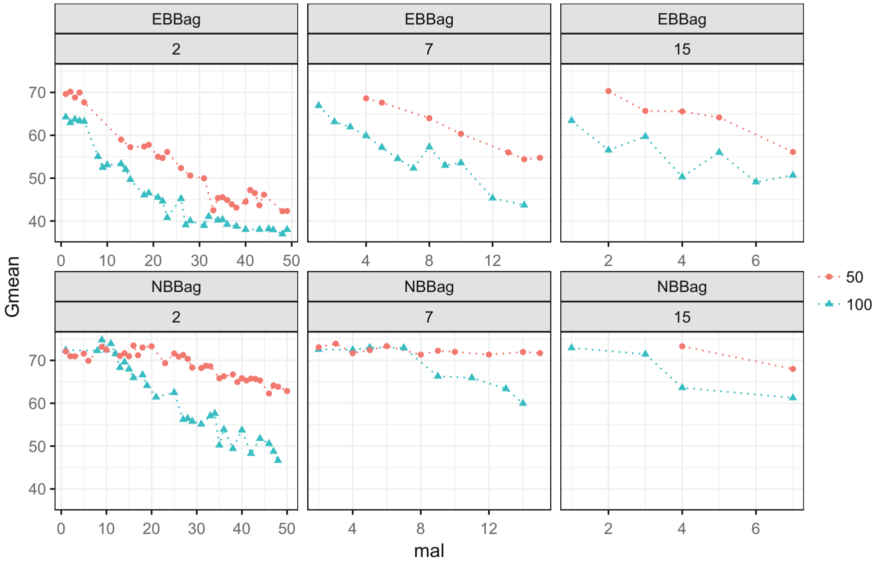

Fig. 3.1 `cleveland` - influence of m_{al} and n_{al} on G-mean of actively balanced (*mwce*) 50% under-sampling, and standard under-sampling of EBBag and NBBags

and `yeast` in the following figures: Figs. 3.1, 3.2, 3.3, and 3.4, respectively. In each of these figures one can compare G-mean performance of actively balanced EBBag ensemble, and actively balanced NBBag ensemble (in a trellis of smaller figures). Three different plots are presented, in each of figures, for each of ensembles, according to tested value of $n_{al} = \{2, 7, 15\}$. On each of plots relation between m_{al} and G-mean is presented for ensembles resulting from 50% under-sampling, and standard (i.e., 100%) under-sampling performed at the first phase of ABBag.

A common tendency for majority of plots representing G-mean performance of *mwce* ABBag on real-world data sets (here represented by data sets: `cleveland` in Fig. 3.1, and `yeast` in Fig. 3.4) is that, regardless of other parameters, one can observe an increase of G-mean performance for initial active balancing iterations (i.e., small values of m_{al}), followed by stabilization or decrease of performance for the further iterations. A decrease of performance is more visible for further iterations of actively balanced EBBag. Thus, the tendency observed for real-word data sets is in line with our motivation for proposing ABBag expressed by an intuition that it should suffice to perform a limited number of small updates of bootstrap by actively selected examples to sufficiently improve performance of under-sampling extensions of bagging.

A different tendency is visible on plots representing G-mean performance of *mwce* ABBag on hard to learn synthetic data sets that we analyzed (represented by data sets: `flower5-3d-10-20-35-35` in Fig. 3.2, and `paw3-3d-10-20-35-35` in Fig. 3.3). One can observe an almost permanent increase of actively balanced

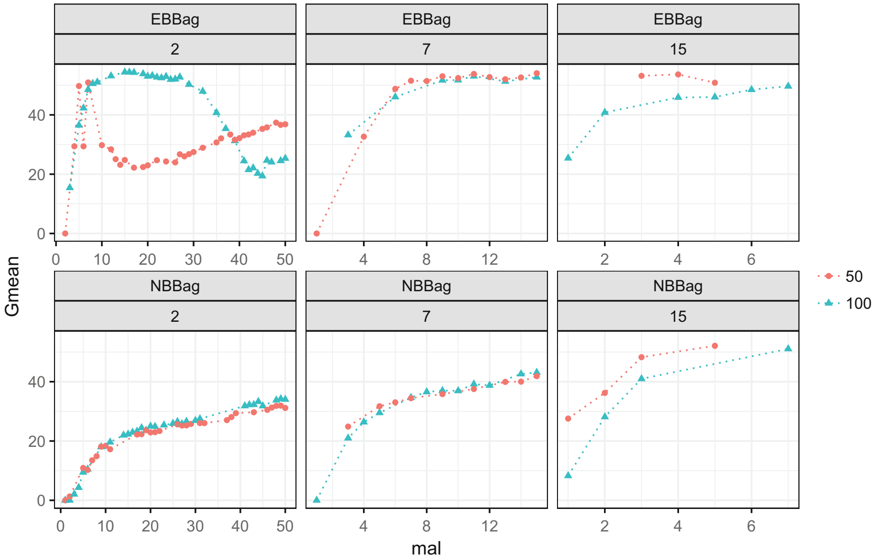

Fig. 3.2 `flower5-3d-10-20-35-35`—influence of m_{al} and n_{al} on G-mean of actively balanced (*mwce*) 50% under-sampling, and standard under-sampling of EBBag and NBBag

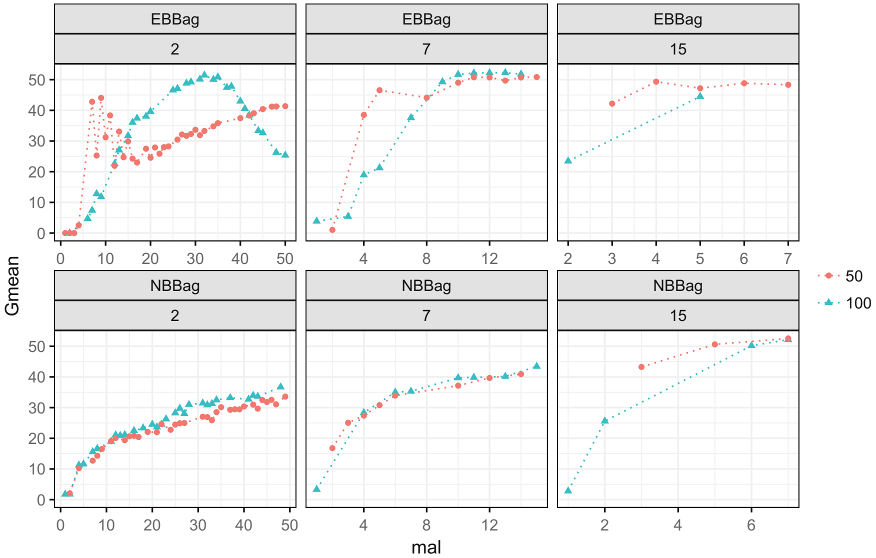

Fig. 3.3 `paw3-3d-10-20-35-35`—influence of m_{al} and n_{al} on G-mean of actively balanced (*mwce*) 50% under-sampling, and standard under-sampling of EBBag and NBBag

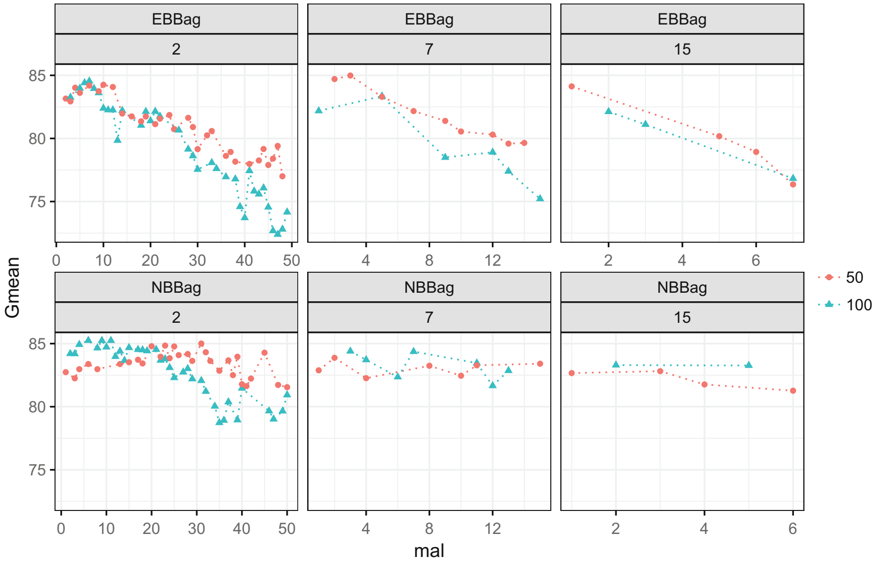

Fig. 3.4 yeast—influence of m_{al} and n_{al} on G-mean of actively balanced ($mwce$) 50% under-sampling, and standard under-sampling of EBBag and NBBag

NBBag ensemble G-mean performance in the following iterations of actively balancing procedure, regardless of other parameters. For active balancing EBBag this tendency is more similar to performance on real-world data sets. After an increase of performance observed for initial iterations, comes a decrease (e.g., see 50% under-sampling EBBag for $m_{al} = 2$ in Fig. 3.3) or stabilization (e.g., see under-sampling EBBag for $m_{al} > 2$ in the same Fig. 3.3). One should take into account, however, that these are really hard to learn data sets, for which base-line classifiers were performing poorly. Thus, the active selection of examples, in the second phase of ABBag, had more place for improvement of the bootstraps, which may explain why more iterations were needed.

3.7 Conclusions

The main aim of this chapter has been focused on the attempts to improve classification performance of the under-sampling extensions of the bagging ensembles to better address class imbalanced data. The current extensions are mainly based on modifying the example distributions inside bootstrap samples (bags). In our proposal we have decided to additionally add a limited number of learning examples coming from outside the given bag. As a result of such small adjusting bootstrap bags, the final ensemble classifier should better balance recognition of examples from minority and majority classes. To achieve this aim, we have introduced a two phase approach, which we have called Actively Balanced Bagging (ABBag).

The key idea, in this approach, is to first learn an under-sampling bagging ensemble and, then, to carry out steps of updating bootstraps by small batches composed of a limited number of actively selected learning examples. These batches are drawn by one of proposed active learning strategies, with one of the example weights. Their definition takes into account a decision margin of ensemble votes for the classified instance, balancing of the example class distribution in its neighbourhood, and prediction errors of component classifiers. To best of our knowledge such a combined approach has not been considered yet.

The results of experiments have demonstrated that:

- The new proposed ABBag ensemble with the active selection of examples has improved G-mean performance of two baseline under-sampling bagging Exactly Balanced Bagging (EBBag) and Nearest Balanced Bagging (NBBag), which are already known to be very good ensemble classifiers for imbalanced data and outperforms several other classifiers (see, e.g., earlier experimental results in [7]).
- Another observation resulting from the presented experiments is that an active selection strategy performs best when it integrates the ensemble disagreement, i.e. the decision margin (which is typically applied in the standard active learning such as QBC) with information on class distribution in imbalanced data and prediction errors of component classifiers. The best performing selection strategy is *mwce* for both EBBag and NBBag classifiers.
- A more detailed analysis of ABBag performance on harder to learn data sets allowed us to observe that it is usually better to add a small number of examples in batches to obtain the best classification performance.

Finally, an open problem for further research, related to the main topic of this book, concerns dealing with a higher number of attributes in the proposed ensemble. Highly dimensional imbalanced data sets are still not sufficiently investigated in the current research (please see, e.g., discussions in [11, 15]). In case of ensembles, it is possible to look for other solutions than typical attribute selection in a preprocessing step or in a special wrapper such as, e.g., proposed in [31]. A quite natural modification could be applying random subspace methods (such as Ho's proposal [27]) while constructing bootstraps. It has already been applied in extensions of Roughly Balanced Bagging [37].

However, our new proposal exploits modeling of the neighbourhood for minority class examples. It is based on distances between examples, e.g. with Heterogeneous Value Difference Metric. As it has been recently shown by Tomasev's research [55] on, so called, *hubness*, a k-nearest neighbourhood constructed on highly dimensional data may suffer from the curse of dimensionality and such metrics are not sufficient. Therefore, the current proposal of ABBag should be be rather applied to datasets with a smaller or medium number of attributes. The prospect extensions of ABBag for a larger number of attributes should be constructed with different techniques to estimate the example neighbourhoods.

Acknowledgements The research was supported from Poznań University of Technology Statutory Funds.

References

1. Abe, N., Mamitsuka, H.: Query learning strategies using boosting and bagging. In: Proceedings of 15th International Conference on Machine Learning, pp. 1–10 (2004)
2. Aggarwal, C., X., K., Gu, Q., Han, J., Yu, P.: Data Classification: Algorithms and Applications. Active learning: A survey, pp. 571–606. CRC Press (2015)
3. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. **6**, 20–29 (2004). https://doi.org/10.1145/1007730.1007735
4. Błaszczyński, J., Deckert, M., Stefanowski, J., Wilk, S.: Integrating selective preprocessing of imbalanced data with Ivotes ensemble. In: Proceedings of 7th International Conference RSCTC 2010, LNAI, vol. 6086, pp. 148–157. Springer (2010)
5. Błaszczyński, J., Lango, M.: Diversity analysis on imbalanced data using neighbourhood and roughly balanced bagging ensembles. In: Proceedings ICAISC 2016, LNCS, vol. 9692, pp. 552–562. Springer (2016)
6. Błaszczyński, J., Stefanowski, J., Idkowiak, L.: Extending bagging for imbalanced data. In: Proc. of the 8th CORES 2013, Springer Series on Advances in Intelligent Systems and Computing, vol. 226, pp. 226–269 (2013)
7. Błaszczyński, J., Stefanowski, J.: Neighbourhood sampling in bagging for imbalanced data. Neurocomputing **150A**, 184–203 (2015)
8. Błaszczyński, J., Stefanowski, J.: Actively Balanced Bagging for Imbalanced Data. In: Proceedings ISMIS 2017, Springer LNAI, vol. 10352, pp. 271–281 (2017)
9. Błaszczyński, J., Stefanowski, J.: Local data characteristics in learning classifiers from imbalanced data. In: J. Kacprzyk, L. Rutkowski, A. Gaweda, G. Yen (eds.) Advances in Data Analysis with Computational Intelligence Methods, Studies in Computational Intelligence. p. 738. Springer (2017). doi:https://doi.org/10.1007/978-3-319-67946-4_2 (to appear)
10. Borisov, A., Tuv, E., Runger, G.: Active Batch Learning with Stochastic Query-by-Forest (SQBF). Work. Act. Learn. Exp. Des. JMLR **16**, 59–69 (2011)
11. Branco, P., Torgo, L., Ribeiro, R.: A survey of predictive modeling under imbalanced distributions. ACM Comput. Surv. **49**(2), 31 (2016). https://doi.org/10.1145/2907070
12. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996). https://doi.org/10.1007/BF00058655
13. Chang, E.: Statistical learning for effective visual information retrieval. In: Proceedings of ICIP 2003, pp. 609–612 (2003). doi:https://doi.org/10.1109/ICIP.2003.1247318
14. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-Sampling Technique. J. Artif. Intell. Res. **16**, 341–378 (2002)
15. Chen, X., Wasikowski, M.: FAST: A ROC–based feature selection metric for small samples and imbalanced data classification problems. In: Proceedings of the 14th ACM SIGKDD, pp. 124–133 (2008). doi:https://doi.org/10.1145/1401890.1401910
16. Cieslak, D., Chawla, N.: Learning decision trees for unbalanced data. In: D. et al. (ed.) Proceedings of the ECML PKDD 2008, Part I, LNAI, vol. 5211, pp. 241–256. Springer (2008). doi:https://doi.org/10.1007/978-3-540-87479-9_34
17. Ertekin, S., Huang, J., Bottou, L., Giles, C.: Learning on the border: Active learning in imbalanced data classification. In: Proceedings ACM Conference on Information and Knowledge Management, pp. 127–136 (2007). doi:https://doi.org/10.1145/1321440.1321461
18. Ertekin, S.: Adaptive oversampling for imbalanced data classification. Inf. Sci. Syst. **264**, 261–269 (2013)

19. Ferdowsi, Z., Ghani, R., Settimi, R.: Online Active Learning with Imbalanced Classes. In: Proceedings IEEE 13th International Conference on Data Mining, pp. 1043–1048 (2013)
20. Fu, J., Lee, S.: Certainty-based Active Learning for Sampling Imbalanced Datasets. Neurocomputing **119**, 350–358 (2013). https://doi.org/10.1016/j.neucom.2013.03.023
21. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H.: Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. C **99**, 1–22 (2011)
22. Garcia, V., Sanchez, J., Mollineda, R.: An empirical study of the behaviour of classifiers on imbalanced and overlapped data sets. In: Proceedings of Progress in Pattern Recognition, Image Analysis and Applications, LNCS, vol. 4756, pp. 397–406. Springer (2007)
23. Grzymala-Busse, J., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data. J. Intell. Manuf. **16**, 565–574 (2005). https://doi.org/10.1007/s10845-005-4362-2
24. He H. Yungian, M.: Imbalanced Learning. Foundations, Algorithms and Applications. IEEE - Wiley (2013)
25. He, H., Garcia, E.: Learning from imbalanced data. IEEE Trans. Data Knowl. Eng. **21**, 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239
26. Hido, S., Kashima, H.: Roughly balanced bagging for imbalance data. Stat. Anal. Data Min. **2**(5–6), 412–426 (2009)
27. Ho, T.: The random subspace method for constructing decision forests. Pattern Anal. Mach. Intell. **20**(8), 832–844 (1998)
28. Hu, B., Dong, W.: A study on cost behaviors of binary classification measures in class-imbalanced problems. CoRR abs/1403.7100 (2014)
29. Japkowicz, N., Stephen, S.: Class imbalance problem: a systematic study. Intell. Data Anal. J. **6**(5), 429–450 (2002)
30. Japkowicz, N.: Shah, Mohak: Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press (2011). doi:https://doi.org/10.1017/CBO9780511921803
31. Jelonek, J., Stefanowski, J.: Feature subset selection for classification of histological images. Artif. Intell. Med. **9**, 227–239 (1997). https://doi.org/10.1016/S0933-3657(96)00375-2
32. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. ACM SIGKDD Explor. Newsl. **6**(1), 40–49 (2004). https://doi.org/10.1145/1007730.1007737
33. Khoshgoftaar, T., Van Hulse, J., Napolitano, A.: Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Trans. Syst. Man Cybern. Part A **41**(3), 552–568 (2011). https://doi.org/10.1109/TSMCA.2010.2084081
34. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-side selection. In: Proceedings of the 14th International Conference on Machine Learning ICML-1997, pp. 179–186 (1997)
35. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms, 2nd edn. Wiley (2014)
36. Lango, M., Stefanowski, J.: The usefulness of roughly balanced bagging for complex and high-dimensional imbalanced data. In: Proceedings of International ECML PKDD Workshop on New Frontiers in Mining Complex Patterns NFmC, LNAI, vol. 9607, pp. 94–107, Springer (2015)
37. Lango, M., Stefanowski, J.: Multi-class and feature selection extensions of Roughly Balanced Bagging for imbalanced data. J. Intell. Inf. Syst. (to appear). doi:https://doi.org/10.1007/s10844-017-0446-7
38. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Tech. Rep. A-2001-2, University of Tampere (2001). doi:https://doi.org/10.1007/3-540-48229-6_9
39. Lewis, D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Proceedings of 11th International Conference on Machine Learning, pp. 148–156 (1994)
40. Liu, A., Zhu, Z.: Ensemble methods for class imbalance learning. In: Y.M. He H. (ed.) Imbalanced Learning. Foundations, Algorithms and Applications, pp. 61–82. Wiley (2013). doi:https://doi.org/10.1002/9781118646106.ch4

41. Lopez, V., Fernandez, A., Garcia, S., Palade, V., Herrera, F.: An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. Inf. Sci. **257**, 113–141 (2014)
42. Napierała, K., Stefanowski, J., Wilk, S.: Learning from imbalanced data in presence of noisy and borderline examples. In: Proceedings of 7th International Conference RSCTC 2010, LNAI, vol. 6086, pp. 158–167. Springer (2010). doi:https://doi.org/10.1007/978-3-642-13529-3_18
43. Napierała, K., Stefanowski, J.: BRACID: A comprehensive approach to learning rules from imbalanced data. J. Intell. Inf. Syst. **39**, 335–373 (2012). https://doi.org/10.1007/s10844-011-0193-0
44. Napierała, K., Stefanowski, J.: Addressing imbalanced data with argument based rule learning. Expert Syst. Appl. **42**, 9468–9481 (2015). https://doi.org/10.1016/j.eswa.2015.07.076
45. Napierała, K., Stefanowski, J.: Types of minority class examples and their influence on learning classifiers from imbalanced data. J. Intell. Inf. Syst. **46**, 563–597 (2016). https://doi.org/10.1007/s10844-015-0368-1
46. Napierała, K.: Improving rule classifiers for imbalanced data. Ph.D. thesis, Poznań University of Technology (2013)
47. Prati, R., Batista, G., Monard, M.: Class imbalance versus class overlapping: an analysis of a learning system behavior. In: Proceedings 3rd Mexican International Conference on Artificial Intelligence, pp. 312–321 (2004)
48. Ramirez-Loaiza, M., Sharma, M., Kumar, G., Bilgic, M.: Active learning: An empirical study of common baselines. Data Min. Knowl. Discov. **31**, 287–313 (2017). https://doi.org/10.1007/s10618-016-0469-7
49. Seaz, J., Krawczyk, B., Woźniak, M.: Analyzing the oversampling of different classes and types in multi-class imbalanced data. Pattern Recognit **57**, 164–178 (2016). https://doi.org/10.1016/j.atcog.2016.03.012
50. Settles, B.: Active learning literature survey. Tech. Rep. 1648, University of Wisconsin-Madison (2009)
51. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Proceedings of the 10th International Conference DaWaK. LNCS, vol. 5182, pp. 283–292. Springer (2008). doi:https://doi.org/10.1007/978-3-540-85836-2_27
52. Stefanowski, J.: Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In: S. Ramanna, L.C. Jain, R.J. Howlett (eds.) Emerging Paradigms in Machine Learning, vol. 13, pp. 277–306. Springer (2013). doi:https://doi.org/10.1007/978-3-642-28699-5_11
53. Stefanowski, J.: Dealing with data difficulty factors while learning from imbalanced data. In: J. Mielniczuk, S. Matwin (eds.) Challenges in Computational Statistics and Data Mining, pp. 333–363. Springer (2016). doi:https://doi.org/10.1007/978-3-319-18781-5_17
54. Sun, Y., Wong, A., Kamel, M.: Classification of imbalanced data: a review. Int. J.Pattern Recognit Artif. Intell. **23**(4), 687–719 (2009). https://doi.org/10.1142/S0218001409007326
55. Tomasev, N., Mladenic, D.: Class imbalance and the curse of minority hubs. Knowl. Based Syst. **53**, 157–172 (2013)
56. Wang, S., Yao, X.: Mutliclass imbalance problems: analysis and potential solutions. IEEE Trans. Syst. Man Cybern. Part B **42**(4), 1119–1130 (2012). https://doi.org/10.1109/TSMCB.2012.2187280
57. Weiss, G.: Mining with rarity: A unifying framework. ACM SIGKDD Explor. Newsl. **6**(1), 7–19 (2004). https://doi.org/10.1145/1007730.1007734
58. Wojciechowski, S., Wilk, S.: Difficulty factors and preprocessing in imbalanced data sets: an experimental study on artificial data. Found. Comput. Decis. Sci. **42**(2), 149–176 (2017)
59. Yang, Y., Ma, G.: Ensemble-based active learning for class imbalance problem. J. Biomed. Sci. Eng. **3**(10), 1022–1029 (2010). https://doi.org/10.4236/jbise.2010.310133
60. Zięba, M., Tomczak, J.: Boosted SVM with active learning strategy for imbalanced data. Soft Comput. **19**(12), 3357–3368 (2015). https://doi.org/10.1007/s00500-014-1407-5