Local Data Characteristics in Learning Classifiers from Imbalanced Data

Jerzy Błaszczyński and Jerzy Stefanowski*

Poznań University of Technology, Institute of Computing Science,
Piotrowo 2, 60-965 Poznań, Poland
{jerzy.blaszczynski,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. Learning classifiers from imbalanced data is still one of challenging tasks in machine learning and data mining. Data difficulty factors referring to internal and local characteristics of class distributions deteriorate performance of standard classifiers. Many of these factors may be approximated by analyzing the neighbourhood of the learning examples and identifying different types of examples from the minority class. In this paper, we follow recent research on developing such methods for assessing the types of examples which exploit either k-nearest neighbours or kernels. We discuss the approaches to tune the size of both kinds of neighborhoods depending on the data set characteristics and evaluate their usefulness in series of experiments with real-world and synthetic data sets. Furthermore, we claim that the proper analysis of these neighborhoods could be the basis for developing new specialized algorithms for imbalanced data. To illustrate it, we study generalizations of oversampling in pre-processing methods and neighbourhood based ensembles.

1 Introduction

Supervised classification is one of the well studied tasks of machine learning, data mining and statistical data analysis. Its aim is to learn the relationship between values of attributes describing examples and a target class of interest. Since many problems can be represented in the attribute value form it has a wide spectrum of possible applications [1]. The classification relationships learned from labeled examples can be used as a classifier to predict class labels for new, unclassified examples. Numerous approaches, based on different principles, have been already introduced to learn classifiers. Nevertheless they may be insufficient when dealing with complexities affecting the data representation.

One of these complexities is *class imbalanced data*, where at least one of the target classes contains a much smaller number of examples than the other classes. This class is usually called the *minority class*, while the remaining classes are denoted as *majority class*(*es*). Imbalanced data often occur in practical problems, such as, medical data analysis, fraud detection, technical diagnostics or image recognition, see, e.g., [8, 20, 60]. In all these problems correct recognition

* A corresponding author

of the minority class is of key importance. Nevertheless, the standard learning algorithms usually do not work properly for these problems since they are biased toward better recognition of the majority classes and they met difficulties, or even are unable, to classify correctly new objects from the minority class [61].

Although the difficulty while learning classifiers from imbalanced data has been known in practical applications for decades, this problem received a particular, growing research interest in the beginning of the current century and several specialized methods have been proposed (for their review see, e.g., [7, 20, 21, 56]). They are usually categorized as classifier-independent pre-processing techniques or modifications of algorithms for learning particular classifiers.

Researchers still treat learning from class imbalanced data as a research challenge and look for new more effective directions. One of these directions includes studying the nature of the imbalanced data, key properties of its underlying distribution and consequences they bring for learning better classifiers or for constructing specialized pre-processing methods.

While examining these properties, it has been noticed that the high, global imbalance ratio between cardinalities of minority and majority classes is not the only and not even the main reason of difficulties in learning classifiers. Other, as we call them, *data difficulty factors*, referring to internal characteristics of class distributions, are also influential. They include: decomposition of the minority class into many rare sub-concepts playing a role of small disjuncts [25, 26], the effect overlapping between the classes [46, 15] or presence of many minority class examples inside the majority class region [39]. When these data difficulty factors occur *together* with class imbalance, they may seriously hinder the recognition of the minority class, see e.g., experimental studies [36, 40, 42, 48].

Please note that aforementioned data factors correspond to *local data characteristics*, occurring in some sub-regions of the minority class distribution rather than at the global level of the entire data set. Furthermore, the development of several informed pre-processing methods, such as [31, 9], is strongly based on exploiting information about example distribution in the neighborhood of considered minority examples.

In the previous research Napierala and Stefanowski have linked data difficulty factors to *different types of examples* forming the minority class distribution [39, 40, 52, 55]. It has led the authors to a differentiation between safe and unsafe examples for recognizing the minority class. These types of examples were identified by analyzing class labels distribution among examples' *neighbours* [40]. Two ways of modeling the neighbourhood have been proposed, either by considering, k – *nearest neighbours* or *kernel functions* [40, 38]. These approaches can be applied to several crucial issues for learning classifiers from imbalanced data:

- to analyze internal characteristics of real-world data sets and establish their difficulty for recognizing minority classes [40, 38];
- to support comparisons of algorithms for learning classifiers as well as pre-processing methods [42];
- to construct new, specialized algorithms for improving classifiers [5].

Nevertheless, in these studies the size of neighborhood was chosen in the simplest way and usually with the same value of the crucial hyper–parameter for all considered data sets. Although it has proven to be sufficiently effective in previous works, a more systematic tuning of this parameter with respect to data set characteristics is still an open research problem and requires more studies.

Therefore, the main aims of the this paper are the following:

1. To introduce a new approach to tune the size of the neighborhood depending on the data characteristics. Unlike the previous works [40, 42], we pay more attention to using kernels in this analysis.
2. To experimentally study usefulness of kernels for an analysis of imbalanced data - also for identifying more types of examples than proposed in [40].
3. To discuss the applicability of this special tuned neighborhood for constructing dynamic pre-processing methods as well as to learning neighbourhood based ensembles dedicated to, imbalanced data.

The paper is organized as follows. The next section summarizes related works on data difficulty factors and using local information in pre-processing methods. The previous approach to an identification of types of minority examples is discussed in Section 3. The new proposal of tuning its parameters is introduced in Section 4 and validated in the experiments in Section 5. The following section discusses its use to construct new pre-processing techniques. Similarly, its applicability for the Nearest Neighbourhood Ensemble is presented in Section 7. Other possible extensions of the presented neighborhood analysis are discussed in Section 8. The final section draws conclusions.

2 Related Research on Imbalanced Data Characteristics

In this section we will briefly discuss the issues most related to studying local characteristics of class imbalanced data. We do not intend to provide here a comprehensive review of methods for dealing with these data. For such a review, the reader is referred to the monograph [20] covering the most representative issues, as well as to systematic surveys, such as [7, 8, 21, 56].

2.1 Nature of the Class Imbalance Problem

Recall that a data set is considered class imbalanced when it is characterized by an unequal distribution of objects in classes. Japkowicz names it a *between-class imbalance* [24]. It may be quantified by a *class imbalance ratio* – which represents a global point of view at data characteristics.

Generally speaking, any data set with unequal distribution of examples between class could be considered as imbalanced. However, there is no common agreement with regard to a precise threshold defined for the global imbalance ratio that would allow to distinguish imbalanced data set [21]. Here we also do not define a precise threshold value but share an opinion saying that the class imbalance problem is associated with lack of data (called also absolute rarity [60]), which hinder the accurate recognition of minority classes [53].

In this study we consider a two class (minority class vs. majority class) formulation of class imbalance problem. It is justified by semantic importance of the rare class versus other classes, which can be considered as the two class problem. Moreover, this formulation of the imbalance problem is mostly studied in the current literature. Even if the original definition of classification problem includes more classes, they are aggregated into one majority class. Note, however, that in some applications it may be reasonable to consider multi-class data sets, where imbalances may exist between various classes and it is required to improve classifier performance with respect to more than one minority class. We will come back to these issues in Section 8.

The class imbalance observed in a data set can be either *intrinsic* (in the sense that it is a direct result of the nature of the data space) or *extrinsic* (caused by reasons external to the data space). Extrinsic imbalance can be caused by high costs of acquiring the examples from the minority class, e.g., due to economic or privacy reasons or it comes from technical, time or storage, limitations [60] .

2.2 Data Complexity and Difficulty Factors

Although many authors have experimentally shown that standard classifiers have difficulties while recognizing the minority class, it has also been observed that in some problems characterized by high imbalance between classes (expressed by the value of the global imbalanced data) standard classifiers are still sufficiently accurate [2]. For instance, Napierala reports several experimental studies which conclude that when there is a clear separation between classes, the minority class can be sufficiently recognized regardless of the high imbalance ratio [38].

These and other studies prove that the *global class imbalance ratio* is not necessarily the only, or even the main, problem causing the decrease of classification performance and focusing only on the global ratio may be insufficient for improving classification performance. *Data complexity*, understood here as the distribution of examples from both classes in the attribute space, has a crucial impact on learning. It is not particularly surprising, since data complexity affects learning also in standard, balanced domains. However, when data complexity occurs *together* with the class imbalance, the deterioration of classification performance is amplified and it affects mostly (or even only) the minority class.

In the context of learning from imbalanced data the term "data complexity" may comprise different data distribution patterns, such as: overlapping, small disjuncts, outliers or noise. Several authors call them as *data difficulty factors*. We describe them briefly below.

Within class decomposition and small disjuncts

The experimental studies with several data sets have shown that minority class usually does not form a homogeneous, compact distribution of the target concept but it is often scattered into smaller sub-parts representing separate subconcepts. Japkowicz named it *within-class imbalance* [26]. This is closely related to the problem of *small disjuncts* which are harder to learn and cause more classification errors than larger sub-concepts.

Although the problem of within-class imbalance may occur in both minority and majority classes, small disjuncts are more characteristic and more critical for a minority class. In the majority class, the sub-concepts will be most often represented by a sufficient number of examples forming larger disjuncts, while in the minority class, in which the examples are already rare, their further decomposition into several sub-concepts will produce small disjuncts, represented by a too small number of examples to be correctly learned. Such fragmentation of the minority class into five smaller sub-parts is illustrated in Figure 1. Additionally each sub-part of the minority has a small over-lapping with the neighbours from the majority class (which constitute an additional difficulty).

According to [25, 26] the higher deterioration of classification performance results from an increased decomposition of the minority class into many sub-parts containing too few examples rather than by changing the global imbalance ratio.

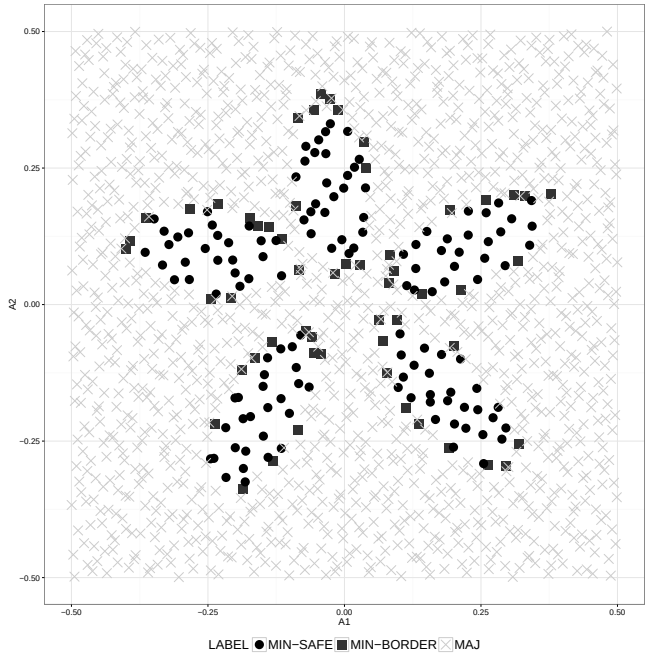

Fig. 1. Visualization of sub-concepts of the minority class additionally affecting by class over-lapping (here represented by borderline examples) in flower data.

Overlapping between the classes

In the boundary regions between classes, the examples from different classes may overlap – which hinders learning classifiers even in a standard, balanced case. As the minority class is underrepresented in the data set, it may be underrepresented

also in the overlapping region. Most learning algorithms tend to shift the decision boundary too close to the minority class, treating the whole overlapping area as belonging to the majority class. Indeed, the experiments on mainly artificial data with different degrees of overlapping have shown that overlapping deteriorated the classifier performance, especially when the minority class was concerned [46]. Furthermore, according to research of [15] the imbalance ratio calculated locally inside the overlapping regions is more influential for the minority class than the global ratio concerning the complete data. In other experiments a combination of increased overlapping between the classes with decomposition of the minority class influenced results more than than changing the class imbalance ratio [39].

Dealing with noisy or outlier examples

Single examples from one class, located far from the decision boundary inside the other class, are usually called noisy examples. Handling noise is often considered in standard machine learning problems, however it becomes even more important issue in learning from imbalanced data. Noisy majority examples are particularly harmful for recognition of the minority class. They may cause a fragmentation of the minority class and increase the difficulties in learning its definition – see a discussion in [38]. Thus, examples of this type are usually either removed/relabeled in the pre-processing phase [48, 55].

On the other hand, distant minority examples surrounded by the majority class examples are not necessarily noisy. As the minority class examples are underrepresented in the data set, such lonely examples may represent a rare but valid sub-concept of which no other representatives could be collected for training [40, 38]. We will call such examples *outliers*.

The role of noise and outliers in learning from imbalanced data has not been deeply studied yet. Few authors have shown that randomly introduced class or attribute value noise results in degradation of classifiers performance on imbalanced data, see e.g., [38]. Some other authors have studied the role of iterative filtering (or removing) noisy (difficult to be correctly classified) minority case examples [48]. More interesting experiments presented in [39] have also shown that single minority examples located inside the majority class regions cannot be simply deleted from the data since their proper treatment by informed pre-processing may improve classification performance for the minority class.

To summarize the discussion of the aforementioned data complexity factors we would like to stress that their identification in real world data sets is not a trivial task. The discussion of this issue and references to known methods are presented in [38, 53].

2.3 Local Data Characteristics in Informed Pre-processing

Recall that the pre-processing methods are classifier independent and they are designed to modify imbalanced data set in a way that transforms the class distribution to a more appropriate one for learning classifiers. Many of these methods generate a more balanced distribution of examples into classes. In general, changing the class distribution towards a more balanced one improves the performance for most data sets and classifiers [21].

The simplest pre-processing methods 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 class cardinalities is reached. Therefore these methods exploit global information about the data set: the current and expected imbalance ratios.

Since simple random pre-processing methods are often not effective, *focused* (also called *informed*) methods have been introduced; see their comprehensive reviews in [21, 7]. Many of these methods attempt to take into account internal characteristics of data regions around minority class examples. Historically, the first such method resulted from Kubat and Matwin's proposal of the *one-side-sampling* method (OSS) [29]. These authors observed that characteristics of mutual positions of examples from different classes is a source of difficulty. Thus, OSS is based on distinguishing different types of learning examples: safe examples (located inside the regions occupied by examples from the given class), borderline (located near the decision boundary) and, so called, noisy examples (these authors understood them as examples from the given class localed inside safe regions of the other classes). According to the OSS filtering approach, borderline and noisy examples are removed from the majority classes, while the minority class is kept unchanged (even for noisy minority examples).

Many other filtering (mainly under-sampling) methods exploits the paradigm of edited nearest classifiers. For instance, the *Nearest Cleaning Rule* (NCR) [31] applies it to removal of "difficult" examples from the majority classes. Briefly speaking, NCR first looks for a specific number k of *nearest neighbours* ($k = 3$ is recommended in [31]) of the "seed" example. Then, it re-classifies seed example according to most frequent class label among neighbours. Finally, it removes from majority class these examples, which cause the wrong re-classification.

The analysis of class labels among k nearest neighbors is also exploited in a hybrid method SPIDER that selectively filters out the majority examples which may lead to incorrect re-classification of the minority ones. [55]. In the first stage it applies the edited nearest rule to distinguish between safe and unsafe examples (which is depending how strongly k neighbours may correctly – or incorrectly – re-classify the given "seed" example). For the majority class, the neighbours which misclassify the seed minority example are either removed or relabeled. Then, in the next stage, the reclassification analysis is repeated and the remaining unsafe minority examples are additionally replicated depending on the number of majority neighbours.

The best known method of informative over-sampling is called Synthetic Minority Over-sampling Technique (SMOTE) [9]. It is also based on the k nearest neighbourhood and exploits it to selectively over-sample the minority class by creating *new synthetic examples* with respect to the global parameter, called *over-sampling* ratio. SMOTE has been further extended in different ways – see reviews in [21, 7]. Quite often these extensions exploit different local information about the learning examples. For instance, the authors of BORDERLINE SMOTE do not treat all minority examples in the same way and focus oversampling around examples from borderline region between classes [19].

3 Analyzing Neighbourhoods of Minority Class Examples

3.1 Motivations

Following the critical analysis of earlier works on using local data characteristics in informed pre-processing and studies on the complexity of imbalanced data Napierala and Stefanowski have decided to link data difficulty factors to *different types of examples* forming the minority class distribution. They proposed to differierate between safe and unsafe examples in learning from imbalanced data [40], however in a different way than earlier proposed, e.g. by [29]. Below we present this categorization following their definitions from [40, 38, 42].

Safe examples are ones located in the homogeneous regions populated by examples from one class only. Other examples are *unsafe* and more difficult for learning. Unsafe examples are categorized into *borderline* (placed close to the decision boundary between classes), *rare cases* (isolated groups of few examples located deeper inside the opposite class), or *outliers*. As the minority class can be highly under-represented in the data, it is claimed that the rare examples or outliers, could represent a very small but valid sub-concepts of which no other representatives could be collected for training [38]. Therefore, they cannot be considered as noise examples which typically are then removed or re-labeled. In Figure 2 all these four types of examples from the minority class are illustrated in the 2-dimensional distribution of the two class data set called `paw`.

Recall experimental studies from [40, 38], where the graphical visualizations techniques based on multi-dimensional scaling and non-linear t-SNE projection have confirmed the occurrence of this categorization of example types in several real-world imbalanced data sets. However, such an analysis cannot be directly applied to larger data. Napierala and Stefanowski have looked for new simple techniques which should more directly identify these types of examples.

Their method origins from the hypotheses [40] on role of the mutual positions of the learning examples in the attribute space and the idea of assessing the type of example by analyzing class labels of the other examples in its *local neighbourhood*.

Following the proposal of [40, 38] – a term local refers to studying characteristics of the nearest examples due to the possible sparse decomposition of the minority class into rather rare sub-concepts with non-linear decision boundaries. Considering a larger size of the neighbourhood may not reflect the underlying distribution of the minority class.

Such a neighbourhood of an example could be modeled in different ways. In the previous research Napierala and Stefanowski proposed to construct it with:

- k–nearest neighbours,
- or kernel functions.

The analysis of class labels of examples in the k–nearest approach concerns a fixed number of nearest examples (without taking into account their distances to the seed examples) while in the kernel approach all examples within a given radius (the kernel bandwidth) are taken into account together with their distances.

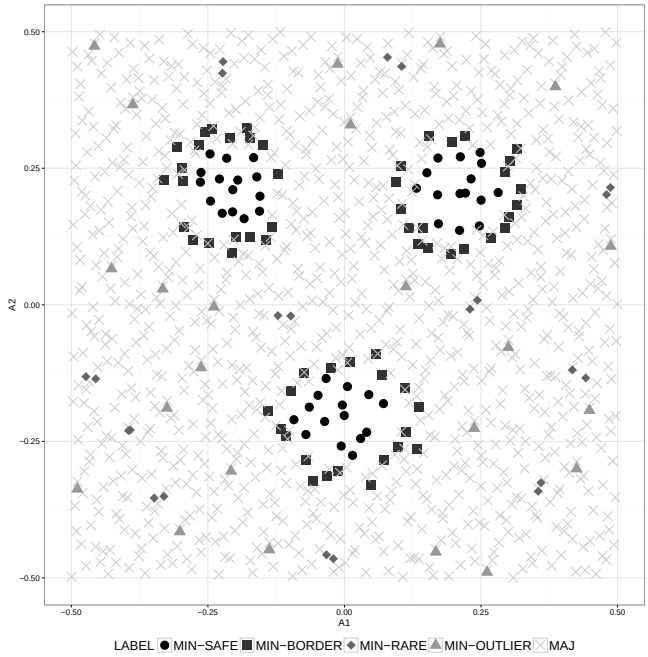

Fig. 2. Visualization of four types of minority class examples in paw data.

We will come back to the problem of tuning their proper values in Section 4. An analysis of the class label distribution of examples inside the neighborhood of the given example allow us to assess its level of difficulty and as a result its type (safe vs. unsafe to be learned).

Note, however, that constructing both types of the neighbourhood involves decisions on choosing the *distance function*. In previous considerations Napierala and Stefanowski have followed results of analyzing different distance metrics [32] and chose the HVDM metric (*Heterogeneous Value Difference Metric*) [63]. Its main advantage for mixed attributes is that it aggregates normalized distances for qualitative and quantitative attributes. In particular, comparing to other metrics, HVDM provides more appropriate handling of qualitative attributes as instead of simple value matching, as it makes use of the class information to compute attribute value conditional probabilities by using a Stanfil and Valtz value difference metric for nominal attributes [63].

More precisely, let x be a seed example and y be another example (potential neighbour). The HVDM is defined over m attributes as

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

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 partial distance for numeric attributes is defined as a normalized metric $(y_i - x_i)$. Then, the partial 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 *svdm* is defined as [10]:

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

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.

In the next two sub-sections we will discuss more precisely previous proposals of modeling these two kinds of the neighbourhood (with k–nearest neighbours or kernel functions) and establishing types of minority class examples [40, 38].

In both cases, deciding about the type of minority examples is based on analyzing class labels of examples in its neighbourhood.

3.2 Modeling *k*-neighbourhood

The k–nearest neighbourhood has been mainly exploited in the previous studies [40, 38, 42] and some applications of this approach to pre-processing [43, 62] or specialized ensembles [5]. These authors have aimed at distinguishing whether an example is safe, borderline, rare or outlier depending on the numbers of examples from minority vs. majority classes in the considered neighbourhood. As we will also discuss in the next section, the size neighbourhood k should not be smaller than 5 as it may poorly distinguish between four types of examples.

In [40] the following rule has been introduced to identify the type of the given example. If all, or nearly all, its neighbours belong the same (usually minority) class, this example is treated as the safe example, otherwise it is one of unsafe types. If the number of both classes inside the k-neighbourhood are quite similar, the example is treating as borderline one. For an extreme situation - all neighbours belong to the opposite class it is clearly an outlier. Finally, the examples with one or sometimes two (for larger sized of the k) neighbours from its class was identified as a rare case.

For the most used the size of neighbourhood $k = 5$, the proportion of neighbours from the same class against neighbours from the opposite class can range from 5:0 (all neighbours are from the same class as the analyzed example) to 0:5 (all neighbours belong to the opposite class). Depending on this proportion, Napierala and Stefanowski have proposed to assign the labels to the examples in the following way:

- 5:0 or 4:1 – an example is labelled as a safe example.
- 3:2 or 2:3 – a borderline example; Note that although the examples with the proportion 3:2 are still correctly re-classified by its neighbours, the number of neighbours from both classes is approximately the same, so it was assumed that this example could be located too close to the decision boundary between the classes.
- 1:4 – labelled as a rare example.
- 0:5 – an example is labelled as an outlier.

Similar interpretations has been extended for larger values of k. For instance, in case of $k = 7$ and the neighbourhood distribution 7:0 or 6:1 or 5:2 – a safe example; 4:3 or 3:4 – a borderline example; again the number of neighbours from both classes are approximately the same; 2:5 or 1:6 – a rare example; and 0:7 – an outlier [38].

Besides using such thresholding, these authors also considered defining the one coefficient expressing a safe level of the given example x – being an estimator of conditional probability of its assignment to the minority class as $p(C_{min}|x) = \frac{k_{min}}{k}$, where C_{min} is a minority class, k is the number of neighbours and k_{min} is the number of minority class neighbours [42].

3.3 Modeling Kernel Neighborhood

An alternative approach to fixing the number of neighbours is to fix the local area around the example as it done in kernel approaches – which was discussed in [38] and studied in [42]. Note that due to the form of the kernel function, different weights (probabilities) could be assigned to the neighbours, based on their distance from the analyzed minority example x. Moreover, unlike having always the same number of examples in the k-neighbourhood modeling, each kernel may cover different number of examples within a fixed radius which rises wider interpretation of local density (see our further experimental analysis in Section 5.2).

Several kernel functions could be considered – besides the most popular Gaussian kernel, other triangular or Epanechnikov functions are among common choices. In this study we have decided to apply Epanechnikov function which is defined as:

$$K(u) = \frac{3}{4}(1 - u^2)\mathbf{1}_{|u|\leq 1},$$

where $u = \frac{d_i}{h}$, d_i is the distance of i-th example (x_i) to the considered example x, and h is bandwidth of the kernel. Epanechnikov kernel is suitable for our purposes since it takes values 0 when $d_i > h$. In this sense, it resembles limits of k-neighbourhood. Moreover, this property will be very useful inside the procedure for tuning the neighborhood size discussed in Section 5.2. The distance between examples d_i is calculated according to HVDM metric (see motivations presented in the earlier section 3.1). Given the definition of the kernel function we estimate a weighted sum of all minority neighbours, where weights depend on the distance from the analyzed example. Comparing it to the weighted sum calculated for

the majority class neighbours we can estimate the probability that the analyzed example x may belong to the minority class $p(C_{min}|x)$.

To assess the type of a minority example, we need to discretize the range of this value into subintervals. Inspired by earlier research [38], in this paper we proposed the following rule: if $1 \geq p(C_{min}|x) > 0.7$ then label x as safe; if $0.7 \geq p(C_{min}|x) > 0.4$ then label x as borderline; if $0.4 \geq p(C_{min}|x) > 0.2$ then label x as rare; if $0.2 \geq p(C_{min}|x) > 0$ then label x as outlier (we keep this type similarly to earlier name); if $p(C_{min}|x) = 0$ then label x as a new type called *zero*. Finally, if there is no other example inside the neighbourhood of x (even from the opposite majority class), then label x as a singleton in an empty sub-region (further called simply *empty*).

Note that this rule is different than the one proposed in [38, 42] as it introduces two new labels, which allow to better understand types of the kernel neighbourhood discovered in data.

3.4 Experiences with Analyzing Types of Minority Examples

The previous experiments with modeling k-nearest neighbourhood applied to UCI imbalanced data sets are described in [38, 42]. They have clearly demonstrated that most of these real-world data do not include many safe minority examples. They rather contain all types of examples, but in different proportions. Depending on the dominating type of identified minority examples, the considered data sets could be labeled as: safe, border, rare or outlier - which show the level of their potential difficulty. Moreover, the thesis [38] has shown that the classifier performance could be related to the category of data. First, for the safe data nearly compared single classifiers (SVM, RBF, k-NN, decision trees or rules) have achieved good, comparable prediction results. The larger differentiation among these classifiers has been noticed for more unsafe data sets (e.g. SVM is worse than k-NN and trees for data with higher number of rare cases and outliers). The similar analysis has been carried out for the most representative pre-processing approaches, showing that the competence area of each method depends on the data difficulty level, based on the types of minority class examples. For more details see [38, 42].

4 Tuning the Neighbourhood Size

In this paper we focus our interest on tuning the size of the neighborhood with respect to characteristics of each data set.

4.1 Tuning k Value

In the previous studies Napierala and Stefanowski [40, 38, 42] exploited mainly k nearest neighbourhood and they showed that values smaller than 5, e.g., $k = 1$ and $k = 3$, may poorly distinguish the type of examples, especially if one wants to assign them to four types. Too high values, on the other hand, would be

inconsistent with the assumption of the locality of the method (see [42] for more details of the discussion why the locality is important for analyzing complex minority class distributions in imbalanced data).

They proposed to set $k = 5$ as the default value. To check whether this parameter k could strongly influence the results of labelling minority examples, a special sensitivity analysis over 26 different data sets was carried out in [42]. Its results have shown that proportions of identified types of examples are quite stable while changing k values (between 5 and 13 – globally defined for all of these data sets). The recommendation of the smallest value of k has come from the paradigm of the most local analysis of the complex decision boundaries of the minority class and its sparsity. Furthermore, the authors pointed out that the parameter $k = 5$ was recommended for many related, informed pre-processing methods (see e.g. [9, 31, 55]).

Nevertheless, the idea of tuning of k parameter, for each imbalanced data set individually, has not been considered so far. Studying the literature one may find some positions that consider changing size of neighbourhoods in a standard k-NN classifier for class balanced data. In these works choosing value k is made with respect to the data set or class cardinality. Refer, e.g., to [17] which recommends approximating $k \approx \sqrt{n}$, where n is the total number of learning examples. However, we hypothesize that in case of imbalanced data n should be rather the size of the minority class. Other researches have proposed some slightly different approximations. Enas and Chai [12] postulated to take

$$k = n^{2/8} \text{ or } k = n^{3/8}.$$

See also [16] for more detailed presentation of similar proposals. Since these formulas have been designed with typical problems and k-NN classifier in mind, Napierala and Stefanowski have expressed their doubts whether they can be directly transferred into a different context of modeling neighborhoods for class imbalanced data [42].

Here, we share this point of view and we propose a method of tuning k value in a cross-validation procedure. The important question concerns the choice of optimization criterion for the tuning method. If one refers to the idea of recognizing the minority class examples as good as possible (which is a key issue in learning from imbalanced data) - such a criterion may reflect abilities of k neighborhood to correctly re-classify examples. This idea is consistent with some earlier proposals of using cross-validation to choose k value which minimize the classification error of a standard k-NN classifier, as it was argued by Dasarathy [11]. We will describe it in more detail in sub-section 4.3.

4.2 Tuning Kernel Bandwidth

Modeling neighbourhood with kernels was preliminary discussed in [38, 42] as an alternative to using k neighbours analysis of imbalanced data. The authors postulated that the Epanechnikov function should be equal to the average distance to the 5^{th} neighbour of each minority example in the data set, as they wanted to

keep the link to their basic k neighbourhood method. Furthermore, in [42] they presented an comparative experiment of labelling the minority class examples in 26 popular imbalanced data sets and demonstrated that using the kernel method does not change the results of k neighbourhood more than by 5-10%.

In this paper we want to consider new approaches for tuning the size of kernel neighbourhood with respect to each data set. Firstly, note that the kernel analysis is often related to *kernel density estimation*, i.e., non-parametric approach to estimation of probability density function, which is one of the most fundamental issues in statistics [33, 50, 51]. Although there are important differences between the density estimation and our problem, one can still notice some similarities while calculating probabilities in considered points of the example space. Recall that exploiting class probabilities inside the kernel neighbourhood of the seed example x may be equivalent to operating on contribution of neighbours with respect to their kernel distance to x. It may be also interpreted in the context of the kernel density estimator

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i),$$

where n is a number of neighbours x_i (or more generally considered data points), K_h a kernel function with a bandwidth size h.

It is also known that the kernel bandwidth is this parameter which strongly influences the resulting probability estimate. Its tuning has been already intensively studied in statistics. The most of approaches attempt to optimize a criterion referring to the expected L_2 risk, which is a kind of the mean integrated squared error between $\hat{f}_h(x) - f(x)$. Although basic formulations involve unknown density function f many automatic, data-based methods have been developed for selecting the bandwidth h; for some reviews refer, e.g., to [27].

If Gaussian basis kernel functions are used to approximate univariate data, and the underlying density being estimated is assumed to be Gaussian, the choice for h (that is, the bandwidth that minimizes the mean integrated squared error) is often estimated as

$$h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.066\hat{\sigma}n^{-1/5}.$$

where $\hat{\sigma}$ is the standard deviation of the examples in the data. This approximation is known as Silverman's rule of thumb [51] and quite often implemented in statistical software. Other bandwidth selection methods were also proposed, for instance Terrell and Scott proposed oversmoothed density estimates which in case of the standard Gaussian kernel leads to the oversmoothed bandwidth $h = 1.144\hat{\sigma}n^{-1/5}$. These considerations could be generalized for the multi-dimensional kernel with H – a symmetric positive bandwidth matrix [33]. For instance the aforementioned rules of thumbs are generalized to

$$h_i = \hat{\sigma}_i \left(\frac{4}{(d+2)/n}\right)^{\frac{1}{d+4}}.$$

Nevertheless, the above tuning methods concern a typical estimation of density function in the unsupervised setting. Although they are sometimes applied as a kind of pre-processing inside the supervised classifiers – in particular Bayesian classifiers, see e.g., [34], in our opinion these methods cannot be transferred directly to our problem of supervised neighbourhood analysis for imbalanced data. However, due to some similarities, we acknowledge inspiration in specialized density estimation methods, which are based on cross-validation optimization of Least Squares forms representing the integrated squared error (ISE) of density functions or, so called, biased versions [50].

4.3 A New Tuning Method based on Cross-validation

Following the critical analysis of tuning k parameter (see Section 4.1), and kernel bandwidth in density estimation (in Section 4.2), we propose a simple cross-validation method to tune both of thes parameters. Our motivation is to make use of abilities of the neighbourhoods of the example x to correctly recognize its class labels. Recall that in learning classifiers from imbalanced data one attempts to improve recognition of the minority class, so studying the neighborhood from the re-classification perspective may be connected with this aim.

The tuning method is based on the optimization procedure which scans a value of neighbourhood parameter (k for k nearest neighbourhood and bandwidth h for kernel neighbourhood) from a pre-defined set of possible values. In our further experiments, for the kernel version we will refer these values to the average distances between minority class examples calculated for a given data set (see Section 5.2). However, in general, they could be other appropriate values. In case of k nearest neighbourhood we will enumerate k values starting from the smallest possible value.

As the optimization criterion we should choose measure reflecting ability of the neighborhoods built on the training examples to recognize the type of the testing example. In further experiment we have decided to apply popular G-mean measure as it aggregates re-classifications of examples from both classes.

For a given value of an analyzed parameter (bandwidth h or k) the data set is split into training and testing parts following the stratified version of cross validation technique. For each split the following schema is carried out:

- For each example from the training part its neighborhood is constructed and tuned with respect to the given parameter value – its size.
- Each example from the testing part is classified by the tuned neighborhood (of the same size as the optimized parameter).
- The classification by the neighbourhood is performed according to highest probability $p(C_i|x)$ that example x, from the test set may belong to class C_i (for problems considered in this paper $i = \{1, 2\}$, since we have only minority class C_{min}, and majority class C_{maj}), estimated according to distribution of classes of examples in the neighbourhood constructed in the training set.
- The value of the optimization criterion is calculated on the basis of how many examples from a test set are correctly classified by the tuned neighbourhood.

The final value of the optimization criterion comes from averaging over several folds inside the cross-validation. The cross-validation may be repeated several times to reduce variance of optimization criterion. The value of the finally chosen neighbourhood parameter that corresponds to the best average optimization criterion is the result of this tuning method.

5 Experimental Analysis of Data Characteristics

5.1 Experimental Setup

In this section we will carry out two kinds of experiments. Firstly, we will show how to tune the kernel neighbourhood and k- neighbourhood sizes, i.e., bandwidth h and k, over different benchmark real-world data sets and synthetic data sets. It should illustrate the usefulness of the method presented in section 4. Secondly, given the tuned sizes of neighbourhood, we will analyze the internal characteristics of imbalanced data sets and establish the level of their difficulty (with respect to different types of the minority examples). This part of experiment should show the applicability of the neighbourhood analysis to recognize the different categories of imbalanced data sets.

Similarly to the related study [42] we will focus our experiments on 13 benchmark real-world imbalanced data sets. Their characteristics is presented in Table 1. We have chosen the data sets which have been often studied in many experimental studies with imbalanced data. They represent different sizes, imbalance ratios (denoted by IR), domains and have both continuous and nominal attributes. Following the most related results [42] some of these data sets should be easier to learn standard classifiers while most of them constitute different degrees of difficulties.

Table 1. Characteristics of real-world data

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

Nearly all of benchmark real-world data sets come from the UCI repository[1]. One data set is medical data set which was used in the earlier works of Stefanowski et al. on class imbalance[2]. 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 the literature.

Furthermore, we have decided to study few synthetic data sets with known data distribution. We apply a specialized generator for imbalanced data [64] and produced two different types of data sets. The examples of both minority classes are generated randomly inside predefined spheres and the majority class examples are uniformly distributed in an area surrounding them. We consider two configurations of these minority class spheres: called `paw` and `flower` – see their 2-D illustrations at Figures 1 and 2. In both data sets the global imbalanced 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 sub-parts or 5 sub-parts. Moreover, each of this data sets has been generated with different numbers of unsafe examples – which is denoted by four numbers inside the name of data. For instance `flower5-3d-30-40-15-15` means that the generated minority class should contain approximately 30% of safe examples, 30% inside the class overlapping, 15% rare and 15% outliers.

5.2 Tuning Kernel Bandwidth and k-neighbourhood

In this experiment we used the method presented in Section 4 to tune the best size of kernels' bandwidth h and the best value of parameter k representing the number of nearest neighbours. The results of the tuning on benchmark real-world data are presented in Table 2, while the results of tuning on synthetic data are presented in Table 3. The results presented in these tables come from stratified 10-fold cross-validation averaged 5 times to improve reproducibility and reduce possible variance of the optimization criterion (here G-mean).

Note that the considered bandwidth h sizes refer to the average distance to k-th nearest neighbour in the minority class of the given data set. This setting allows us to obtain more comparable results and make the bandwidth size dependent on the characteristics of each data set that was analyzed. Please note that value of k-neighbour according to the average distance in the minority class relates to some extend to the value of k in the other approach based on nearest neighbours. Technically, we considered values of the kernel bandwidth corresponding to average distance to k-th neighbour, with k from interval $[0.5, 9]$ with a basic step 0.5.

We have chosen these values as we wanted to check smaller neighbourhoods, which was already well motivated in the previous research presented in [42]. In case of the other approach based on nearest neighbours, we considered only $k = \{5, 6, 7, 8, 9\}$ for the same reasons. The choice of $k \geq 5$ is motivated here by

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

[2] 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 fact that neighbourhoods smaller than 5 do not allow to perform sensible labelling of example types that we presented in Section 5.3. This argument is not viable for average k values related to the bandwidth size. In Tables 4 and 5, we present an average number of examples inside the kernel for bandwidths tuned in experiments on real-world and synthetic data sets, respectively.

Note that average numbers of nearest neighbours in kernels of real-world data sets, presented in Tables 4, are always higher than 5. For synthetic data sets, presented in Table 5, one can observe that the average number of examples inside kernels is smaller than 3 in case of the most difficult to learn distributions of examples (data sets: `flower5-3d-10-20-35-35`, `paw3-3d-10-20-35-35`). In case of these two data sets, rare and outlier examples are the most numerous in the minority class. This result can be explained when we take a look at results from the Table 3. For these data sets the value of average k is the smallest possible, which means that it was better to keep the neighbourhood (and the bandwidth) as small as possible to obtain the best optimization result of G-mean.

Table 2. Bandwidth h and k tuned on real-world data

	kernel			k-NN	
data set	avg. k	h	G-mean	k	G-mean
abalone	6.5	0.074	36.679	5	45.547
breast-cancer	8	0.087	52.480	7	57.324
car	8	$\simeq 0$	77.265	5	87.627
cleveland	1	0.523	22.190	5	41.997
cmc	1	0.059	47.963	5	58.233
ecoli	7	0.332	76.739	9	80.300
haberman	9	0.328	43.624	5	56.552
hepatitis	6	0.812	65.695	7	71.893
scrotal-pain	8.5	0.408	55.955	9	77.244
solar-flare	1	0.038	27.095	5	50.609
transfusion	3	0.128	53.976	7	60.710
vehicle	8.5	0.516	88.682	5	93.883
yeast	2.5	0.430	34.391	5	60.018

A comparison of results obtained with tuning kernels and nearest neighbours variants, reported in Tables 2, and 3, shows that kernel neighbourhoods works differently than k nearest neighbourhoods. This observation comes mainly from the comparison of G-mean values obtained in the tuning process. Regardless whether we compare on real-world or synthetic data sets, k-neighbourhood achieves higher G-mean values than kernel neighbourhood.

However, one should be careful with drawing conclusions from comparing average k related to the tuned kernel bandwidth with k tuned directly for nearest neighbours as the kernel approach using other ranges. Nevertheless, it is visible that higher values of bandwidths in kernels relate always to higher values of k in nearest neighbours. We can also notice that larger neighbourhoods are selected for easier data sets.

Table 3. Bandwidth h and k tuned on synthetic data

	kernel			k-NN	
data set	avg. k	h	G-mean	k	G-mean
flower5-3d-10-20-35-35	0.5	0.058	43.199	7	52.549
flower5-3d-100-0-0-0	9	0.077	91.906	9	96.407
flower5-3d-30-40-15-15	2.5	0.103	79.623	9	80.998
flower5-3d-30-70-0-0	9	0.076	89.802	9	96.082
flower5-3d-50-50-0-0	9	0.077	92.757	8	96.506
paw3-3d-10-20-35-35	0.5	0.066	44.088	7	49.319
paw3-3d-100-0-0-0	8.5	0.099	95.425	9	97.067
paw3-3d-30-40-15-15	2	0.113	78.178	7	79.186
paw3-3d-30-70-0-0	9	0.100	90.252	7	93.189
paw3-3d-50-50-0-0	8.5	0.098	92.458	9	95.090

Table 4. Average k (for tuned bandwidth) and average number of examples inside a kernel for real-world data

data set	avg. k	avg. n
abalone	6.5	115.04
breast-cancer	8	41.12
car	8	14.39
cleveland	1	18.74
cmc	1	6.96
ecoli	7	25.37
haberman	9	54.25
hepatitis	6	36.69
scrotal-pain	8.5	58.46
solar-flare	1	273.93
transfusion	3	38.55
vehicle	8.5	22.33
yeast	2.5	62.24

Table 5. Average k (for tuned bandwidth) and average number of examples inside a kernel for synthetic data

data set	avg. k	avg. n
flower5-3d-10-20-35-35	0.5	3.10
flower5-3d-100-0-0-0	9	12.56
flower5-3d-30-40-15-15	2.5	18.16
flower5-3d-30-70-0-0	9	12.96
flower5-3d-50-50-0-0	9	12.55
paw3-3d-10-20-35-35	0.5	2.88
paw3-3d-100-0-0-0	8.5	12.28
paw3-3d-30-40-15-15	2	15.82
paw3-3d-30-70-0-0	9	14.81
paw3-3d-50-50-0-0	8.5	12.94

The size of the kernel bandwidth (the distance values) presented in Tables 2, and 3 is not easy to interpret since it is a value of HVDM metric (please see Section 3). Note, however, that values of the bandwidth on real-world data sets have higher variance than these observed for synthetic data sets. It seems natural that real-world data sets should present more variability than synthetic ones.

5.3 Analyzing Types of Minority Examples

In this part experiment, we used the previously tuned bandwidths of kernels and k-neighbourhoods to label different types of minority class examples in real-world and synthetic data sets (it is somehow inspired by the earlier analysis in [40]). The results obtained for benchmark real-world data sets with kernel neighbourhood are presented in Table 6, and the ones obtained with k-neighbourhood are presented in Table 7.

Table 6. Labelling of minority class examples in real-word data for the tuned bandwidth

data set	safe [%]	borderline [%]	rare [%]	outlier [%]	zero [%]	empty [%]
abalone	4.78	10.15	8.66	70.75	3.58	2.09
breast-cancer	17.65	18.82	31.76	29.41	1.18	1.18
car	0.00	47.83	43.48	8.70	0.00	0.00
cleveland	2.86	2.86	25.71	42.86	17.14	8.57
cmc	13.81	21.32	24.02	13.21	20.42	7.21
ecoli	5.71	68.57	14.29	5.71	5.71	0.00
haberman	1.23	25.93	39.51	29.63	2.47	1.23
hepatitis	28.12	21.88	3.12	34.38	6.25	6.25
scrotal-pain	15.25	20.34	28.81	22.03	1.69	11.86
solar-flare	4.65	6.98	16.28	65.12	4.65	2.33
transfusion	5.06	38.76	27.53	16.85	6.74	5.06
vehicle	55.78	35.68	5.53	0.00	0.50	2.51
yeast	7.84	11.76	27.45	39.22	9.80	3.92

Let us first explain differences in the number of example types identified by the two approaches to model neighbourhoods. Recall that differently than in [42], we have not applied the same labelling rule and the tuned values of k are different and vary depending on the given data set (see values of k for k-NN in Table 2 for details). Instead we used analogous rules, which are formulated according to estimated values of probability of minority class, for both kernels and k-neighbourhood (please see Section 4 for details).

The next important difference comes from the new assumption that the kernel approach allows us to identify more types of examples. It is clearly visible for the real-world data sets (see Table 6) which contain minority examples of all six different types. A similar observation is valid for the same data sets analyzed with k-neighbourhood (in Table 7), although here we distinguish four types. Let us also note that the results presented in Table 7 correspond well with the

Table 7. Labelling of minority class examples in real-word data for tuned k

data set	safe [%]	borderline [%]	rare [%]	outlier [%]
abalone	11.04	8.36	23.58	57.01
breast-cancer	29.41	28.24	29.41	12.94
car	60.87	21.74	13.04	4.35
cleveland	0.00	22.86	17.14	60.00
cmc	23.72	18.32	31.23	26.73
ecoli	28.57	48.57	14.29	8.57
haberman	14.81	29.63	38.27	17.28
hepatitis	43.75	28.12	12.50	15.62
scrotal-pain	38.98	42.37	15.25	3.39
solar-flare	0.00	18.60	32.56	48.84
transfusion	26.97	33.71	15.17	24.16
vehicle	78.89	13.57	6.03	1.51
yeast	15.69	19.61	21.57	43.14

previous ones presented in [42]. Nevertheless, some differences in proportions are visible mostly for more difficult data sets (e.g., `abalone`, `solar-flare`, `yeast`).

Even though numbers of examples into different types labelled by kernel neighbourhood and k-neighbourhood are not exactly the same, the characteristics of the particular data sets (i.e. their categorization with respect to dominating types of minority examples) are generally quite similar. In particular, the highest number of outliers is discovered for the same data sets: `yeast`, `solar-flare`, `abalone`, `cleveland`. The highest number of rare type examples is also discovered for the same data sets: `cmc`, `breast-cancer` (although k-neighbourhood discovers the same amount of safe examples), `haberman`. The same applies to borderline and safe examples. The highest number of borderline examples is discovered for data sets: `transfusion`, and `ecoli`. The highest number of safe examples is discovered by both kernel and k neighbourhood for `vehicle`. Limited differences in labeling are observed for few data sets only: `hepatitis`, `scrotal-pain`, and `car`.

Table 8. Labelling of minority class examples in synthetic data for tuned bandwidth

data set	safe [%]	borderline [%]	rare [%]	outlier [%]	zero [%]	empty [%]
flower5-3d-10-20-35-35	20.21	22.87	21.28	0.00	35.11	0.53
flower5-3d-100-0-0-0	84.57	14.89	0.53	0.00	0.00	0.00
flower5-3d-30-40-15-15	35.64	34.04	3.19	14.36	12.77	0.00
flower5-3d-30-70-0-0	76.60	23.40	0.00	0.00	0.00	0.00
flower5-3d-50-50-0-0	77.13	22.34	0.53	0.00	0.00	0.00
paw3-3d-10-20-35-35	14.67	20.67	24.67	0.67	36.00	3.33
paw3-3d-100-0-0-0	65.33	34.67	0.00	0.00	0.00	0.00
paw3-3d-30-40-15-15	26.00	42.67	4.67	11.33	15.33	0.00
paw3-3d-30-70-0-0	44.67	52.00	3.33	0.00	0.00	0.00
paw3-3d-50-50-0-0	57.33	40.67	2.00	0.00	0.00	0.00

One can notice that new types of examples discovered by the kernel neighbourhood are present in almost all data sets. There are two exceptions: zero type examples are not discovered in `car`; then empty type examples are not found in `car`, and `ecoli`. These type of examples are not dominant in any data set. Since they reflect poor performance of kernel neighbourhood at estimating probability of minority class, one should not expect to find a lot of them. Still, relatively high numbers of zeros and empty type examples is found in data sets: `cleveland` and `cmc`. Relatively high number of zero examples only is found in `yeast`. Furthermore, a relatively high number of empty type examples is found in `scrotal-pain`. Some relations between the numbers of discovered zero and empty type examples and the predictive performance of kernel neighbourhood (in Table 2) can be also observed.

The labeling results obtained for synthetic data sets with kernel neighbourhood and k-neighbourhood are presented in Table 8 and in Table 9, respectively.

Table 9. Labelling of minority class examples in synthetic data for tuned k

data set	safe [%]	borderline [%]	rare [%]	outlier [%]
flower5-3d-10-20-35-35	25.00	5.32	36.17	33.51
flower5-3d-100-0-0-0	87.77	12.23	0.00	0.00
flower5-3d-30-40-15-15	52.66	17.55	17.02	12.77
flower5-3d-30-70-0-0	77.13	22.87	0.00	0.00
flower5-3d-50-50-0-0	90.43	9.57	0.00	0.00
paw3-3d-10-20-35-35	18.00	12.00	34.67	35.33
paw3-3d-100-0-0-0	70.67	29.33	0.00	0.00
paw3-3d-30-40-15-15	54.00	16.00	14.67	15.33
paw3-3d-30-70-0-0	76.00	23.33	0.67	0.00
paw3-3d-50-50-0-0	66.00	34.00	0.00	0.00

We can conclude that the types of examples injected to synthetic data sets are rather well discovered by both kernel neighbourhood and k-neighbourhood. Safer distributions of examples in data sets (without rare and outlier type examples) are recognized in the best way. There is a tendency to mislabel some of safe examples as borderline (which could explained for examples located very closed to the decision boundaries that they are too dominated by neighbors from the opposite class), however, the reverse tendency (to mislabel borderline as safe) is also observable (especially for k-neighborhood). Rare and outlier types of examples are much better recognized by k-neighborhood than kernel neighborhood. We can hypothesize that the kernel neighborhood expresses a worrying tendency to discover outliers as zero type (and also sometimes empty type) examples. This result can be linked to choosing too small bandwidth by the tuning procedure for difficult distributions of examples.

To sum up, this kind of labeling analysis shows the usefulness of modeling the neighborhood to identify the level of difficulty of the studied data set. Generally speaking, the less safe examples, the more difficult could be the data set. It is

also interesting to notice that most of studied data sets do not contain too many safe examples. The percentage of rare, outlier or even empty example is quite high for some of data sets. In particular the kernel analysis may provide more information than k neighborhood approach due to new types of examples.

6 Improving Pre-processing Techniques with the Neighbourhood Analysis

One can ask whether the estimation of probability of minority class examples, which is behind the labelling of minority class, may be useful to improve pre-processing of imbalanced data sets. Therefore, we compare performance of a standard unprunned `J48` classifier trained on data sets pre-processed according to the neighbourhood analysis with kernel and k-neighbourhoods against the same classifier trained on not-processed and randomly over-sampled data sets. The choice of over-sampling is motivated by its' ease of implementing as compared to under-sampling.

Table 10. G-mean [%] for unprunned `J48` learned on base (original) and over-sampled real-world data

data set	base	random	kernel	k-NN
abalone	53.790	60.198	60.802	60.481
breast-cancer	56.495	68.139	68.764	68.791
car	89.851	90.356	90.157	89.681
cleveland	48.984	56.570	50.365	51.716
cmc	56.706	64.142	64.541	64.494
ecoli	70.489	74.011	74.080	74.401
haberman	56.060	54.559	57.394	56.492
hepatitis	63.136	72.058	66.507	68.809
scrotal-pain	69.563	70.570	70.313	71.781
solar-flare	44.249	44.522	42.867	44.110
transfusion	60.018	56.071	56.456	56.564
vehicle	91.929	94.405	93.912	92.609
yeast	54.564	53.735	55.535	57.219

The proposed over-sampling technique uses probability of minority class estimated for each of minority class example according to the frequency of examples in tuned kernel neighbourhood and k neighbourhood (we use the same tuning as comes from the analysis carried out in Section 5.3). The estimated probability is used as a weight of example in the sampling procedure. The difference with respect to the neighbourhood analysis is that, since we apply over-sampling, we want difficult examples (thus, having low value of the probability) to be more represented in the over-sampled data set than safe ones. To achieve this result we simply use inverse of the probability as the weight and replicate them proportionally to this value. In general, we want to achieve approximately balanced

classes, so we estimate the global number of need copies and divide this number among all minority examples with respect to their weights.

Classification performance of `J4.8` with pre-processing technique is measured by standard measures such as G–mean and sensitivity. G-mean results are presented in Tables 10, and 11, for real-world, and synthetic data sets, respectively.

Table 11. G-mean [%] for unprunned `J48` learned on base and over-sampled synthetic data

data set	base	random	kernel	k-NN
flower5-3d-10-20-35-35	0.000	39.627	38.835	38.426
flower5-3d-100-0-0-0	89.410	88.692	87.245	88.190
flower5-3d-30-40-15-15	72.924	72.281	70.576	73.215
flower5-3d-30-70-0-0	87.205	87.496	86.000	85.125
flower5-3d-50-50-0-0	90.530	89.306	89.834	88.442
paw3-3d-10-20-35-35	0.000	33.252	34.634	33.474
paw3-3d-100-0-0-0	88.205	89.231	89.894	88.192
paw3-3d-30-40-15-15	71.320	73.613	74.417	74.074
paw3-3d-30-70-0-0	88.491	85.650	86.153	84.993
paw3-3d-50-50-0-0	89.499	87.421	86.449	86.088

G-mean classification results on real-world data sets show rather limited influence of the proposed pre-processing on predictive performance. In general, one can observe improvements for several difficult data sets: `yeast`, `haberman`, then smaller improvements are also noted for: `abalone`, `breast-cancer`, and `ecoli`. For safer data sets like: `vehicle`, `car` one may expect that no oversampling (base) or random over-sampling may be sufficient solutions (i.e., they may perform better). Then, we acknowledge that no oversampling is best performing on `transfusion`. Moreover, random over-sampling works best on two data sets:`solar-flare`, and `cleveland`.

The results on synthetic data sets also show no significant improvement when kernel and k-neighborhood over-sampling is applied. Better performance in comparison to random over-sampling and no over-sampling (base) can be observed on some more difficult distributions. Sensitivity results confirm the observations made with respect to G-mean. Thus, we do not include tables with these results due to the page limits.

More encouraging results have been obtained for modifications of SMOTE, in particular the recent proposal called Local Neighbourhood extension of SMOTE (briefly LN-SMOTE) which is inspired by the analyzing local data characteristics of the minority examples [37]. Its comparative study against basic SMOTE and two other related generalizations applied with 3 different classifiers (`J48`, Naive Bayes and k-NN) showed that it improved G-mean and F-measure on several of real world data sets. Yet another modifications of SMOTE with respect to individual difficulty weights of examples has been also considered in [43].

7 Neighbourhood Based Ensembles

Ensembles are another kind of methods which could be improved by the neighbourhood analysis. The current proposals of ensembles dedicated to class imbalanced data are mainly extensions of known strategies as bagging, boosting or random trees. They usually either employ pre-processing methods before learning component classifiers or embed the cost-sensitive framework in the ensemble learning process; see their review in [14]. Previous comparative studies, such as [4, 14], have showed that extensions of bagging ensembles are quite promising. The most popular extensions pre-process bootstrap samples by under-sampling the majority class or over-sampling the minority class to obtain a balance of class cardinalities in each bootstrap sample. Roughly Balanced Bagging (RBBag), which is a kind of specialized under-sampling approach leads to best improvements [54].

In this section we want to show that using neighbourhood based approach to change distributions of minority class examples in bootstrap samples may improve performance of bagging ensemble classifiers and result in solutions being competitive to Roughly Balanced Bagging.

We focus on k-neighbourhoods in bagging ensembles, since they proved to better render the distribution of minority class examples in Section 5.2. Moreover, they have been already successfully integrated in the Neighbourhood Balanced Bagging (NBBag), which we have proposed [5].

Neighbourhood Balanced Bagging is based on a different principle than all known bagging extensions for class imbalance. First, instead of integrating bagging with pre-processing, it keeps the standard bagging idea. What changes are probabilities of sampling examples to bootstraps. The chance of drawing minority examples is, sometimes radically, amplified (which is controlled by a special hyper-parameter ψ). Furthermore, the amplification depends on the type of difficulty of minority example identified according to its k-neighbourhood.

We have already shown that NBBag works in both types of bagging generalizations: over-sampling and under-sampling [5]. In first type of generalization, it is similar to over-sampling minority class examples into bootstraps, however, at the same time, the probabilities of drawing majority class examples are decreased. The size of bootstrap is kept the same as the size of the original learning set. The second type is inspired by under-sampling generalizations, which predicts better than over-sampling generalizations [5]. The probabilities of drawing minority class examples are increased, while probabilities of drawing majority class examples are decreased.

Most of the extensions of bagging for imbalanced data are non-parametric [6]. They do not introduce any new parameters, which need to be adjusted during construction of an ensemble of classifiers. On the one hand, one can argue that bagging itself is a parametric method since the adequate size of the ensemble for a given problem is not known a priori. The size of the ensemble is a parameter, which may influence the performance of each of the considered extensions. On the other hand, fixing this parameter enables comparison of ensembles of the

same size, which should allow to distinguish ones which perform better than the others under the same conditions.

Different types of parameters are introduced in NBBag [5] to control the characteristics of neighbourhood: size of neighbourhood k, and amplification factor ψ. In the experiments comparing NBBag to other bagging extensions presented in [5] these two parameters were carefully selected to provide the best average performance. The previous tuning of these parameters was made post-hoc, i.e., first results were obtained for a number of promising pairs of parameter values and then the best values were chosen. On the other hand, we need to look for more appropriate approaches to tune these parameter inside learning an ensemble rather than in a post-hoc way.

Tuning of such model parameters is a known problem in machine learning [18]. However, to our best knowledge, this problem has drawn rather limited attention in the context of learning ensembles from imbalanced data. Class imbalance may limit using some more advanced parameter tuning techniques. To put it simply, minority class examples are to valuable to spare them for tuning purposes only, while majority class examples are not. Following this observation, we investigate a basic technique taken from tree learning. In the same way as reduced-error pruning uses training data [47], we divide training data set into two stratified samples. The first sample is used for training NBBag models and the second one to validate the trained models. After the best parameters are selected, NBBag classifier is constructed on the whole training set. Contrary to what was presented in [5], this technique does not allow to distinguish best values of parameters for all data sets nor even for one data set when learning of a classifier is repeated, as e.g., in cross-validation. Tuning of parameters is performed independently for each constructed component classifier.

In the following, we present performance of two variants of Neighbourhood Balanced Bagging: under-sampling (uNBBag) and over-sampling (oNBBag) with tuning of k and ψ parameters among a limited set of values (small k, and limited amplification of examples weight represented by ψ – please consult [6] for details). Tuning of best parameter values is performed on 2/3 of the training set. The remaining 1/3 of training set is used for the validation.

Now we experimentally compare classification performance of uNBBag and oNBBag to Exactly Balanced Bagging (EBBag) [23], Over-Bagging (OverBag) [58], and Roughly Balanced Bagging (RBBag) [22]. The size of ensembles is fixed to 50 components, `J48` with exactly the same parameters as in Section 6 is used as component classifier. We restrict our comparison to real-world data sets only.

The results of G-mean and sensitivity are presented in Tables 12 and 13, respectively. These results were estimated by a stratified 10-fold cross-validation repeated ten times to reduce the variance of measures.

Looking at both Tables 12 and 13, one can notice that uNBBag and RBBag stand out as the best performing classifiers. Another observation is that over-sampling extensions of bagging, represented by OverBag and oNBBag, provide worse performance that under-sampling extensions. When we compare G-mean performance of ensemble classifiers to performance of over-sampled single classi-

Table 12. G-mean [%] of NBBag and other bagging ensembles on real-world data

data set	EBBag	OverBag	uNBBag	oNBBag	RBBag
abalone	78.845	69.230	79.517	78.706	79.035
breast-cancer	58.175	60.718	58.465	58.795	60.091
car	96.668	96.959	96.356	96.851	96.568
cleveland	73.628	51.629	73.260	66.754	71.130
cmc	64.191	61.036	65.051	63.787	65.350
ecoli	88.178	83.896	88.435	85.380	88.430
haberman	64.144	63.329	63.742	61.779	63.533
hepatitis	79.137	75.816	78.035	74.762	79.457
scrotal-pain	73.679	74.038	72.923	71.997	75.618
solar-flare	83.710	64.649	83.149	79.994	83.421
transfusion	66.607	67.748	66.449	66.476	67.143
vehicle	95.038	94.934	95.440	95.115	95.417
yeast	84.018	63.167	84.475	79.557	85.016

fiers (see Table 10) it is clear that ensembles provide better performance except for `breast-cancer`, where ensembles are only better than single classifier trained on not pre-processed data (i.e., base). A more detailed comparison on G-mean shows that RBBag and uNBBag does not perform best only in case of some relatively safe data sets like: `car` (both classifiers), `scrotal-pain` (uNBBag) or more difficult `breast-cancer` (uNBBag), and `cleveland` (RBBag).

Table 13. Sensitivity [%] of NBBag and other bagging ensembles on real-world data

data set	EBBag	OverBag	uNBBag	oNBBag	RBBag
abalone	80.925	51.224	80.776	75.851	77.045
breast-cancer	63.412	54	65.176	59.059	58.471
car	100	95.652	100	95.942	100
cleveland	80.286	30.571	79.143	63.429	69.143
cmc	70.240	50.721	68.739	63.423	64.685
ecoli	92	76	92	84	90.571
haberman	56.914	59.136	63.827	66.543	55.802
hepatitis	83.438	67.188	79.062	69.688	77.500
scrotal-pain	76.271	70.169	76.441	73.051	75.763
solar-flare	88.140	46.977	86.744	81.395	85.581
transfusion	66.517	61.236	72.697	67.753	65.674
vehicle	97.236	94.523	97.286	95.477	96.935
yeast	91.765	40.980	90.392	73.529	88.431

With respect to values of sensitivity (Table 13) uNBBag and EBBag are clearly the best performing classifiers. uNBBag provides the best recognition of minority class in case of almost all of considered real-world data sets.

This analysis of classification performance of bagging extensions leads to conclusions, which are concordant with the ones presented in [5] and in [6].

RBBag and uNBBag are identified as two outstanding alternatives. Moreover, an exploitation of a relatively simple parameter tuning technique, including a dynamic adaptation of the neighborhood size, allowed us to obtain quite satisfactory predictive performance of NBBag.

8 Extensions of the Neighbourhood Analysis

In this section we briefly point out potential extensions of the neighbourhood approaches which may be useful for some applications – although they are not studied in this paper. We focus our attention on the following three issues:

Identification of class decomposition into sub-concepts

The discussed neighbourhood analysis may approximate some data difficulty factors only. In particular, it does not directly identify a decomposition of the minority class into sub-concepts. As it was discussed in the Section 2.1 research of Japkowicz and her collaborators on *within-class imbalance* showed that increasing the number of the sub-concepts decreased classification performance more than increasing the global imbalance ratio *between class imbalance* [24, 26]. The comprehensive summary of other studies on the role of such class decomposition is presented in [53].

The open question is how to automatically identify such sub-concepts in real-world data sets. In cluster-oversampling proposal, Japkowicz applied k-means clustering algorithm to examples from each class separately [44]. However, it is necessary to estimate the unknown number of expected clusters or to choose an optimization criterion (the most popular criteroa are not defined for the context of imbalanced data). Moreover, these kinds of algorithms are not appropriate for dealing with complex decision boundaries or outlier examples. In our opinion there is a need for developing a new kind of a semi-supervised algorithm (where it is necessary to deal with presence of minority vs. majority examples inside clusters).

Highly-dimensional data sets

The presented approach uses HVDM metric to calculate distances between examples. Similarly to using Euclidean metric in most of pre-processing methods it is more suitable for problems with relatively small or medium number of attributes. On the other hand, high dimensional data sets may occur in image analysis, bio-medical data analysis, genetics or other fields. The use of such dissimilarity measures and k-nearest neighbor principle on such data sets may suffer from the curse of dimensionality as it has been recently showed by Tomasev's research on, so called, *hubness*-aware shared neighbor distances for high-dimensional k-nearest neighbor classification [57].

Recall that this problem is also a challenge for standard learning of classifiers as it increases risks of over-fitting as well as spurious findings. However, considering it with class-imbalanced predictions presents an additional source of difficulties, as it biases classification towards majority class for most classifiers (see, e.g., experimental analyses from [3]). In standard balanced classification feature selection or projections techniques, such as: SVD or PCA, are often applied

to enhance predictive performance. Although these methods have been extensively studied, they mey be too biased toward majority class. Although, some new class imbalance techniques have been recently introduced, we postulate still more research also in the context of an identification of types of examples.

Multiple Imbalanced Classes

A binary classification task is mostly studied in case of imbalanced data. This formulation is justified by focus an interest on the most important class and real-world semantics, like in medical diagnosis (distinguishing sick vs. healthy patients). On the other hand, in some situations it may be reasonable to distinguish more classes with low cardinalities [59].

Considering multiple minority classes makes the learning task more difficult as relations between particular classes become more complex [59]. Internal data distributions or decision boundaries will be different than in case when some classes are aggregated. Techniques developed for binary imbalanced problems are usually not directly applicable to multi-class problems. Quite often they lose performance on one class while trying to gain it on another. A brief review of current specialized techniques is available in [49].

We could ask a question on possible generalizations of the neighbourhood analysis for more than one minority class. Although it has not been studied yet, two directions could be considered. Either one can decompose the multi-class imbalanced data set to a set of binary problems - one minority class vs. all other classes; consider them independently and somehow aggregate results. According to [28] it is a dominating strategy in specialized ensembles, see e.g., [13].

However, in such decomposition of the multiple imbalanced classes, pairwise relations between two classes may be too strongly over-simplified and they do not reflect more complex relations / interactions between several of classes, as one class influences several neighboring classes at the same time. Therefore, it may be more interesting to consider interaction of examples from various minority classes while defining types of examples or exploiting other information from the neighbourhood analysis – however, it is still a topic for further research.

9 Final Remarks

In this paper we follow earlier research on studying the internal characteristics of class imbalanced data and its consequences for difficulties while learning classifiers. We share opinions of researches [25, 26, 15, 36] who showed that the high imbalance ratio between the minority and majority classes (measured on the global level of the data) is not the only and not even the main reason of these difficulties. Other data difficulty factors, such as decomposition of the minority class into many rare sub-concepts, the effect of too strong overlapping between the classes or a presence of too many minority examples inside the majority class region, referring to more local characteristics of class distributions, are more influential.

Our current study on these local data characteristics and difficulties goes along research lines introduced by Napierala and Stefanowski in [40, 42]. They

have proposed to capture the aforementioned data difficulty factors by considering the local characteristics of learning examples from the minority class and by an identification of four basic types of examples: safe, borderline, rare case and outlier. It has been achieved by analyzing the class distribution of examples from different classes inside a *local neighborhood* of the considered example which could be modeled either by means of k-neighbours or kernels.

As the tuning the size of these two kinds of neighbourhoods with respect to characteristics of given data sets have not been sufficiently studied yet, the first contribution of this paper is discussing tuning methods. In our opinion simple rules of thumb are simply not suitable. We have rather promoted tuning bandwidth of a kernel neighbourhood or number k of nearest neighbours using the adapted version of cross validation optimization methods.

Results of many experiments presented in Section 5 have confirmed usefulness of these tuning methods. Moreover, they were sufficiently consistent with earlier results of establishing categories of data set difficulty with respect to dominating types of minority class examples [40, 42]. However, unlike the earlier studies, in this paper we have managed to find an individual size of neighbourhood for each data sets. A general observation is that this size is larger for easier imbalanced data while it becomes smaller for data sets treated as more difficult to be learned.

The other contribution of the current paper is to promote incorporating the results of analyzing this neighbourhood of minority class examples in construction of new methods for learning classifiers from imbalanced data. We have "implemented" this postulate by considering two main categories of methods specialized for imbalanced data: (1) the most popular over-sampling and (2) the generalization of bagging ensembles which incorporates the results of an analyzing the local neighbourhood to re-sample examples into bootstrap samples.

The experiments presented in Section 7 have demonstrated that Nearest Balanced Bagging in the version of under-sampling with local tuning the size of neighbourhoods and the level of re-sampling achieved the best predictive results. Furthermore, experiments presented in Sections 5.2, and 6 have shown that the k nearest neighbours variant has led to better predictions than the kernel neighbourhood. On the other hand, the kernel analysis allows to identify new types of minority class examples: singletons in empty sub-regions (which is an extreme rarity situation being different to single examples surrounded by k-neighbours from opposite classes - this extension may be valuable in studying medical complex data with many untypical cases of disease, see [45])

Issues of dealing with the local characteristics of imbalanced data may still open several lines of future research. Besides already mentioned semi-supervised clustering for detecting small disjuncts, re-considering the neighbourhood based methods in highly dimensional spaces or multi-class imbalanced problems one could look for other tasks such as:

- Other, more sophisticated proposals of dynamic re-sampling (also under-sampling) of both classes with respect to identified different, local characteristics of sub-regions of imbalanced data.

- Considering a new type of cost-sensitive re-sampling where costs of misclassification between classes will be taken into account while defining types of the minority examples; Then the cost post-posterior probability should be joined together with an estimation of different density of examples in various sub-regions.
- Studying differences between outliers and real noise in imbalanced data; detecting them, developing a new method for dealing with such noisy examples.
- Exploiting information about types of examples in modifications of other algorithms, see e.g., promising results of the rule induction algorithm, called BRACID [41].
- Studying imbalanced data streams affected by concept drifts, i.e., changes in definitions of target classes over time [65]; In particular, recent studies have shown needs for developing new kinds of ensembles for the imbalanced and evolving data streams.

Acknowledgment. The research was funded by the the Polish National Science Center, grant no. DEC-2013/11/B/ST6/00963. Close co-operation with Krystyna Napierala in research on modeling types of examples and with Mateusz Lango in research on ensemble models is also acknowledged.

References

1. Aggarwal, C.C. (Ed.): Data classification: algorithms and applications. Chapman & Hall / CRC (2015).
2. Batista, G.,Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1), 20–29 (2004).
3. Blagus, R., Lusa, L.: Class prediction for high- dimensional class-imbalanced data. BMC Bioinformatics, 11, 523 (2010).
4. 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 226, 269–278 (2013).
5. Błaszczyński, J., Stefanowski, J.: Neighbourhood sampling in bagging for imbalanced data. Neurocomputing, vol. 150 A, 184–203 (2015).
6. Błaszczyński, J., Lango, M.: Diversity Analysis on Imbalanced Data Using Neighbourhood and Roughly Balanced Bagging Ensembles. In Proc. of ICAISC 2016. Lecture Notes in Computer Science, vol. 9692, 552–562 (2016).
7. Branco, P., Torgo, L., Ribeiro, R.: A survey of predictive modeling under imbalanced distributions. ACM Computing Surveys (CSUR), 49(2), 31 (2016) - to appear.
8. Chawla, N.: Data mining for imbalanced datasets: An overview. In Maimon O., Rokach L. (eds.): The Data Mining and Knowledge Discovery Handbook, Springer, 853–867 (2005).
9. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-sampling Technique. J. of Artificial Intelligence Research, 16, 341-378 (2002).
10. Cost, S., Salzberg, S.: A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. Machine Learning Journal, vol. 10 (1), 1213–1228 (1993).

11. Dasarathy, B.V.: NN concepts and techniques: an introductory survey. In: Nearest Neighbor Norms, NN Pattern Classification Techniques. IEEE Press, 1–30 (1991).
12. Enas, G., Chai, S.: Choice of the smoothing parameter and efficiency of the k-nearest neighbour classification. Computers and Mathematics with Applications, vol 12, 308–317 (1986).
13. Fernandez, A., Lopez, V., Galar, M., Jesus M., Herrera, F.: Analysis the classification of imbalanced data sets with multiple classes, binarization techniques and ad-hoc approaches. Knowledge Based Systems, 42, 97-110 (2013).
14. 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 Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 99, 1–22 (2011).
15. Garcia, V., Sanchez, J.S., Mollineda, R.A.: An empirical study of the behaviour of classifiers on imbalanced and overlapped data sets. In Proc. of Progress in Pattern Recognition, Image Analysis and Applications 2007, Springer, LNCS, vol. 4756, 397–406 (2007).
16. Gatnar, E.: Multimodel approach to discrimination and regression issues.(In Polish), PWN Warszawa, (2008).
17. Goldstein, M.: K_n-Nearest Neighbour Classification. IEEE Transactions on Information Theory, 627–630, (1972).
18. Guyon, I., Saffari A., Dror, G., Cawley, G..: Model Selection : Beyond the Bayesian / Frequentist Divide, Journal Machine Learning Research, 11, 61–87, (2010).
19. Han, H., Wang, W., Mao, B.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In Proc. ICIC, Springer LNCS vol. 3644, 878-887 (2005).
20. He, H., Yungian, Ma (eds): Imbalanced Learning. Foundations, Algorithms and Applications. IEEE - Wiley, (2013).
21. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering, vol. 21 (9), 1263–1284 (2009).
22. Hido S., Kashima H.: Roughly balanced bagging for imbalance data. In Proc. of the SIAM Int. Conference on Data Mining, 143-152 (2008) - an extended version in Statistical Analysis and Data Mining, vol. 2 (5-6), 412–426 (2009).
23. Hoens, T., Chawla, N.: Generating Diverse Ensembles to Counter the Problem of Class Imbalance. In Proc. PAKDD 2010, 488–499 (2010).
24. Japkowicz, N.: Concept-Learning in the Presence of Between-Class and Within-Class Imbalances. In Proc. Canadian Conference on AI 2001: 67-77 (2001).
25. Japkowicz, N., Stephen, S.: Class imbalance problem: a systematic study. Intelligent Data Analysis Journal, vol. 6 (5), 429-450 (2002).
26. Jo, T., Japkowicz, N.: Class Imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter, 6 (1), 40–49 (2004).
27. Jones, M.C., Marron, J.S., Sheather, S. J.: A Brief Survey of Bandwidth Selection for Density Estimation. Journal of the American Statistical Association, vol. 91, no. 433, 401–407 (1996).
28. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress Artificial Intelligence (2016) accepted for publication.
29. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-side selection. In Proc. of the 14th Int. Conf. on Machine Learning ICML-97, 179-186 (1997).
30. Lango, M., Stefanowski, J.: The Usefulness of Roughly Balanced Bagging for Complex and High-dimensional Imbalanced Data. In Proc. of Int. ECML PKDD Work-

shop on New Frontiers in Mining Complex Patterns NFmCP 2015, Springer LNAI 9607, 93-107, (2015).
31. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Tech. Report A-2001-2, University of Tampere, (2001).
32. Lumijarvi, J., Laurikkala, J., Juhola, M.: A comparison of different heterogeneous proximity functions and Euclidean distance. Stud Health Technol. Inform., 107 (Part 2), 1362–1366 (2004).
33. Ledl, T.: Kernel Density Estimation: Theory and Application in Discriminant Analysis. Austrian Journal of Statistics. vol. 33 (3), 267–279 (2004).
34. Liu, B., Yang, Y., Webb, GT., Boughton, J.: A Comparative Study of Bandwidth Choice in Kernel Density Estimation for Naive Bayesian Classiffication. In Proc. of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD '09, Springer LNCS vol. 5476, 302–313 (2009).
35. Lin, W., Chen, J.: Class–imbalanced classifiers for high-dimensional data. Briefings in Bioinformatics, 14(1), 13–26 (2013).
36. 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. Information Sciences 257, 113-141,(2014).
37. Maciejewski, T., Stefanowski, J.: Local neighbourhood extension of SMOTE for mining imbalanced data. In Proc. IEEE Symp. on Computational Intelligence and Data Mining, 104–111 (2011).
38. Napierala, K.: Improving rule classifiers for imbalanced data. Ph.D. Thesis. Poznan University of Technology, (2013).
39. Napierala, K., Stefanowski, J., Wilk, Sz.: Learning from imbalanced data in presence of noisy and borderline Examples. In Proc. of 7th Int. Conf. RSCTC 2010, Springer, LNAI vol. 6086, 158–167 (2010).
40. Napierala, K., Stefanowski, J.: The influence of minority class distribution on learning from imbalance data. In Proc. 7th Conf. HAIS 2012, LNAI vol. 7209, Springer, 139–150 (2012).
41. Napierala, K., Stefanowski, J.: BRACID: a comprehensive approach to learning rules from imbalanced data. Journal of Intelligent Information Systems, vol. 39 (2), 335-373 (2012).
42. Napierala, K., Stefanowski, J.: Types of Minority Class Examples and Their Influence on Learning Classifiers from Imbalanced Data. Journal of Intelligent Information Systems, 46 (3), 563–597 (2016).
43. Napierala, K., Stefanowski, J., Trzcielinska, M.: Local Characteristics of Minority Examples in Pre-processing of Imbalanced Data. In: T. Andreasen et al. (eds.): Proc. ISMIS 2014, LNAI vol. 8502, Springer, 123–132 (2014).
44. Nickerson, A., Japkowicz, N., Milios, E.: Using unsupervised learning to guide resampling in imbalanced data sets. In Proc. of the 8th Int. Workshop on Artificial Intelligence and Statistics, 261–265 (2001).
45. Niemann, U., Spiliopoulou, Volzke, H., Kuhn, J.P.: Subpopulation discovery in epidemiological data with subspace clustering. Foundations of Computing and Decision Sciences, vol. 39 (4), 271–300, (2014).
46. Prati, R., Batista, G., Monard, M.: Class imbalance versus class overlapping: an analysis of a learning system behavior. In Proc. 3rd Mexican Int. Conf. on Artificial Intelligence, 312–321 (2004).
47. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. (1993)

48. Saez, J., Luengo, J., Stefanowski, J., Herrera, F.: Addressing the noisy and borderline examples problem in classification with imbalanced datasets via a class noise filtering method-based re-sampling technique. Information Sciences, 291, 184–203 (2015).
49. Seaz, J., Krawczyk, B., Wozniak, M.: Analyzing the oversampling of different classes and types in multi-class imbalanced data. Pattern Recognition (2016), DOI 10.1016/j.atcog.2016.03.012.
50. Sheather, S.J.: Density estimation. Statistical Science, 19(4), 588–597 (2004).
51. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall/CRC (1986).
52. Stefanowski, J.: Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In S.Ramanna, L.C. Jain, and R.J. Howlett (eds), Emerging Paradigms in Machine Learning, 277-306 (2013).
53. Stefanowski, J.: Dealing with data difficulty factors while learning from imbalanced data. In Mielniczuk, J., Matwin, S. (eds) Challenges in Computational Statistics and Data Mining, Springer, 333–363 (2016).
54. Stefanowski, J.: On Properties of Under-sampling Bagging and its Extensions for Imbalanced Data, In Proc. of the 9th Int. Conference on Computer Recognition Systems CORES 2015, Springer, 407–417 (2016).
55. Stefanowski, J., Wilk, Sz.: Selective pre-processing of imbalanced data for improving classification performance. In Proc. of the 10th Int. Conf. DaWaK 2008. LNCS vol. 5182. Springer, 283–292 (2008).
56. Sun, Y., Wong, A., Kamel, M.: Classification of imbalanced data: a review. Int. J. Pattern Recognition Artificial Intelligence 23(4), 687–719 (2009).
57. Tomasev, N., Mladenic, D.: Class imbalance and the curse of minority hubs. Knowledge-Based Systems. 53, 157–172 (2013).
58. Wang, S., Yao, T.: Diversity analysis on imbalanced data sets by using ensemble models. In Proc. IEEE Symp. Comput. Intell. Data Mining, 324–331 (2009).
59. Wang, S., Yao, X.: Mutliclass imbalance problems: analysis and potential solutions. IEEE Trans. System Man Cybern., Part B. 42 (4), 1119–1130 (2012).
60. Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter, vol. 6 (1), 7-19 (2004).
61. Weiss, G.M., Provost, F.: Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research, 19, 315–354 (2003).
62. Wilk, S., Stefanowski, J., Wojciechowski, S., Farion, K.J, Michalowski, W.: Application of Preprocessing Methods to Imbalanced Clinical Data: An Experimental Study. In Pietka E.(ed.) Information Technologies in Medicine, Springer, 503–515 (2016).
63. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research, 6, 1-34 (1997).
64. Wojciechowski, S., Wilk, Sz.: The generator of synthetic multi-dimensional data. Poznan Univ. of Technology Report RB-16/14 (2014).
65. Zliobaite, I., Pechenizkiy, M., Gama, J.: An overview of concept drift applications. In Japkowicz, N., Stefanowski, J. (eds) Big Data Analysis: New Algorithms for a New Society, Springer Studies in Big Data Series, 91–11, (2016).