sciendo

Int. J. Appl. Math. Comput. Sci., 2019, Vol. 29, No. 4, 769–781
DOI: 10.2478/amcs-2019-0057

amcs

USING INFORMATION ON CLASS INTERRELATIONS TO IMPROVE CLASSIFICATION OF MULTICLASS IMBALANCED DATA: A NEW RESAMPLING ALGORITHM

MAŁGORZATA JANICKA [a], MATEUSZ LANGO [a,*], JERZY STEFANOWSKI [a]

[a] Institute of Computing Sciences
Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznań, Poland
e-mail: malgjanicka@gmail.com,
{mateusz.lango,jerzy.stefanowski}@cs.put.poznan.pl

The relations between multiple imbalanced classes can be handled with a specialized approach which evaluates types of examples' difficulty based on an analysis of the class distribution in the examples' neighborhood, additionally exploiting information about the similarity of neighboring classes. In this paper, we demonstrate that such an approach can be implemented as a data preprocessing technique and that it can improve the performance of various classifiers on multiclass imbalanced datasets. It has led us to the introduction of a new resampling algorithm, called Similarity Oversampling and Undersampling Preprocessing (SOUP), which resamples examples according to their difficulty. Its experimental evaluation on real and artificial datasets has shown that it is competitive with the most popular decomposition ensembles and better than specialized preprocessing techniques for multi-imbalanced problems.

Keywords: imbalanced data, multi-class learning, re-sampling, data difficulty factors, similarity degrees.

1. Introduction

In imbalanced data at least one of the target classes contains a much smaller number of examples than the other classes. This underrepresented class, called a *minority class*, gains more importance than the remaining majority class(es), and its correct recognition is particularly required in many applications. Standard learning algorithms are biased toward better recognition of the majority classes and they met difficulties (or even are unable) to classify correctly new instances from the minority class (see He and Ma, 2013).

Up to now many specialized methods for improving the classification of imbalanced data have been introduced. Nevertheless, some problems are still worth to be studied more deeply (Krawczyk, 2016; Stefanowski *et al*., 2017). One of them is dealing with *multiple decision classes*.

Note that most of the current research concerns binary classification problems, i.e., with a single minority class and a single majority class. This formulation is justified by the nature of typical imbalanced problems, where a single minority class is most important from the application perspective and it is essential to improve its recognition. Focusing on this single class usually leads to seeing all the remaining classes as one aggregated class. However, in some problems it may be reasonable to focus interest on more minority classes and such class binarization may be questionable. For instance, in some medical problems physicians may consider few different types of an illness as more critical than other less serious disorders. In such situations, aggregating classes into a binary version is unacceptable, in particular when one critical (usually also rare) disease class would be joined with a majority class of more healthy patients (Lango *et al*., 2017). On the other hand, joining minority classes together will not lead to different therapies for various types of a disease. Similar needs for distinguishing more minority classes in medical procedures are discussed by Wojciechowski *et al*. (2017).

The current approaches to deal with multiple imbalanced classes are mainly based on decomposition of the multiclass problem to special binary subtasks. The most popular are adaptations of earlier known one-versus-one (OVO) or one-versus-all (OVA) ensemble

*Corresponding author

schemes, which apply resampling methods for binary problems (Fernandez *et al*., 2013). Other simpler preprocessing methods usually straightforwardly oversample the minority classes to the size of the majority ones (Zhou and Liu, 2010) or iteratively duplicate the smallest class with the SMOTE method (Fernández-Navarro *et al*., 2011).

Although the selected minority classes are specially re-sampled in these approaches, the information about decision boundaries between various classes or its internal data distributions is lost, while in the original problem one class may influence several neighboring classes at the same time. Furthermore, these binary decompositions do not consider the mutual relations between classes that are different for majority and minority classes and increase the complexity of the learning task.

To illustrate the need for dealing with class interrelations, consider the asthma diagnosis discussed by Lango *et al*. (2017). The two types of asthma, being minority classes, are more closely related to each other and their treatment procedures do not differ too much, while the similarity of these classes to the majority class (nearly healthy patients) is much lower, which is also reflected in a simpler and less aggressive medicine therapy. Such different neighbourhood relations between classes should be taken into account while constructing new approaches to multiclass imbalances.

Following these motivations, Lango *et al*. (2017) recently introduced a new approach to examine the interrelations of multiclasses in imbalanced data. It generalizes the previous approach to study the *types of the examples' difficulty* for binary class datasets, which is based on the analysis of a class distribution in the neighborhood of the examples (Napierala and Stefanowski, 2012; 2016). In the multiclass generalization, it also exploits additional information about the similarity of neighboring classes to the class of an examined example. Lango *et al*. (2017) showed that this approach is capable of identifying data difficulty factors in multiclass imbalanced data. Nevertheless, the question of exploiting the information coming from that proposal in the design of new methods for improving classification of imbalanced data remains open.

Therefore, the aim of this paper is to study whether this approach could be used in a new preprocessing approach to multiclass imbalanced data. We will show this by introducing a new re-sampling algorithm, called Similarity Oversampling and Undersampling Preprocessing (abbreviated as SOUP), which first removes the most harmful majority class examples and than oversamples the most important minority ones according to their safe levels resulting from analyzing their neighbourhood. Furthermore, we will demonstrate that elements of SOUP can be used to modify resampling in the binarization-based ensembles, particularly those relying on the OVO principle. In order to validate the usefulness of these newly introduced methods, we will experimentally compare them with the most popular methods specialized for dealing with multiclass imbalanced data (which do not model interrelations among classes) covering both preprocessing and ensemble decomposition ones. Furthermore, we will examine how different ways of defining class similarities may influence the SOUP performance.

The paper is organized as follows. Section 2 covers the most related previous works. In Section 3 we provide background of the proposal to model class interrelations and to estimate example difficulty levels. The new resampling algorithm SOUP and modifications of the OVO ensemble are introduced in Section 4. Section 5 describes the experimental analysis of the proposed methods and their comparison with the state-of-the-art algorithms. Section 6 concludes our study.

2. Related works on multiclass imbalances

Multiclass classification problems are considered to be more difficult than their binary counterparts. For instance, Wang and Yao (2012) experimentally demonstrated that increasing the number of classes is strongly correlated with a decrease in many popular classification measures even for more balanced data. The bulk of the proposed methods for binary imbalanced data are not directly applicable for multiple classes. The current approaches to multiclass imbalances are mainly based on adapting binary preprocessing techniques, special algorithmic modifications or using misclassification costs; see their review by Fernández *et al*. (2018). Following these authors, the most popular decomposition approaches are one-vs-all (OVA) and one-vs-one (OVO) strategies.

The OVA method constructs a binary subproblem for each class, where all other classes are aggregated into one common class. More precisely, from the original dataset D, a series of datasets $D_1, D_2, \ldots, D_c$ is constructed where c is the number of classes. Each dataset D_i contains all the examples of D but the class label y is replaced by $\mathbb{I}[y = i]$ where $\mathbb{I}$ is the indicator function. On each dataset D_i a binary classifier $\mathcal{C}_i$ is trained. For a new instance x, the class is usually assigned by the component classifier with the highest confidence, i.e., $\arg\max_{i\in\{1,2,\ldots,c\}} \mathcal{C}_i(x)$.

Contrary to one-vs-all, the one-vs-one approach exploits a decomposition of multiclasses into all possible pairs of classes. For each pair of classes, a dataset D_{ij} which contains the examples of classes i and j is constructed. Again, the class label is replaced by $\mathbb{I}[y = i]$, binary classifiers $\mathcal{C}_{ij}$ are trained on corresponding datasets D_{ij}, and the final decision is made by $\arg\max_{i\in\{1,2,\ldots,c\}} \sum_{j=1}^{c} \mathcal{C}_{ij}(x)$. It is noteworthy that this method creates a quadratic number of classifiers

$(c(c-1)/2)$ with respect to the number of classes which may cause problems when the number of classes is high.

All these approaches have been adopted for the imbalanced case. The most studied extensions of OVO and OVA ensembles are those which apply resampling methods (such as random oversampling, undersampling or SMOTE) to balance class distribution in binary datasets. Their experimental evaluation was extensively studied by Fernandez *et al*. (2013). Various aggregation methods of component classifiers outputs were also studied (Galar *et al*., 2011). Other decomposition techniques are surveyed by Fernández *et al*. (2018).

There are also other ensemble approaches for multi-class imbalanced data. Abdi and Hashemi (2016) propose a Mahalanobis distance-based oversampling method and combine it with a boosting algorithm, creating MDOBoost. Other combinations of random resampling with boosting are proposed by Wang and Yao (2012). Recently, yet another extension of Roughly Balanced Bagging to multiclass imbalanced data has been proposed by Lango and Stefanowski (2018).

There are special preprocessing methods for multiclass imbalanced data. The most well-known is Global-CS which takes inspirations from rescaling approaches in cost-sensitive learning, where Zhou and Liu (2010) proposed to assign an equal weight to every class, independently of their cardinality. Fernandez *et al*. (2013) argue that the simplest way of achieving it is by the means of random oversampling. In Global-CS each class is oversampled except the class with the highest cardinality. First, each instance is copied $\lfloor n_{\max}/n_i \rfloor$ times, where n_i is the size of an example's class and $n_{\max}$ is the size of the biggest majority class. Then, $(n_{\max} \bmod n_i)$ examples are randomly oversampled for each class i. After these operations every class has an equal number of examples. Besides this uninformed preprocessing method, also some informed oversampling methods have been proposed, most notably Static-SMOTE (Fernández-Navarro *et al*., 2011), which works iteratively. In each iteration the class with the smallest cardinality is selected and duplicated with the standard SMOTE technique (treating all examples from nonselected classes as majority ones). The number of the method's iterations is set as the number of classes. There is also a limited number of works on combinations of oversampling with undersampling (Agrawal *et al*., 2015), which include a selective hybrid resampling SPIDER3 (Wojciechowski *et al*., 2017), where relations between classes are captured by predefined misclassification costs. Moreover, Seaz *et al*. (2016) have applied types of minority examples of Napierala and Stefanowski (2012) to independently oversample single minority classes, however without considering any relations between classes. For a more detailed review of methods for multi-class imbalanced learning, see the work of Fernández *et al*. (2018).

3. Modeling multiple class interrelations and data difficulty factors

The proposal presented by Lango *et al*. (2017) results from two inspirations:

(i) the need for richer modeling complex relations between classes, which is missed by current approaches, and

(ii) previous studies with handling data difficulty factors by means of the types of examples for binary imbalanced classification.

Discussing the first point, please note that one class may be a majority one when it is compared with some other classes but at the same time it may be a minority class with respect to the remaining classes (Krawczyk, 2016; Wang and Yao, 2012). The simple resampling of single classes is usually insufficient to deal with these situations. Then, distributions of many classes are quite complex and boundaries between them may overlap. As a result, examples from these overlapped regions, which belong to different classes and have similar attribute descriptions, usually negatively influence predictive accuracy. However, their influence on each class recognition may be different. Thus, when dealing with multiple classes, one may easily lose performance on one minority class while attempting to improve it at another classes (Seaz *et al*., 2016).

Moreover, the availability of expert knowledge on the classes' interpretation and their mutual relations should influence the solutions to multiclass imbalanced classification. As discussed in Section 1, some minority classes can be treated as more closely related to each other than to the majority class in some practical applications (see, e.g., the asthma diagnosis case). It may impact both the evaluation of data difficulty and the development of methods for improving classification. In the first perspective, the class similarity should be taken into account while considering which class misclassifications are better and which are worse according to an expert (it is different from the expert's misclassification costs of Wojciechowski *et al*. (2017)).

This is related to a more general problem of analyzing neighboring examples for the given class and considering which other class examples are more preferred to be the closest neighbors of this class according to the expert knowledge. Such class neighborhood analysis is particularly useful while modifying the example distribution in preprocessing techniques, where one should decide which class examples should be introduced in the given subregion of data (where examples of other classes already exist). The decomposition approaches, which treat all pairs of classes equally, do not reflect these issues properly (Seaz *et al*., 2016).

Following the second motivation point, class imbalances are often accompanied by additional *data difficulty factors*. These factors referring to internal characteristics of class distributions may be even more influential than the *global imbalance ratio* between cardinalities of minority and majority classes. They include the decomposition of the minority class into many rare subconcepts playing a role of small disjuncts (Jo and Japkowicz, 2004; Stefanowski, 2016), overlapping between the classes (Prati *et al*., 2004; Garcia *et al*., 2007) or the presence of many minority class examples inside the majority class region (Napierala *et al*., 2010). A joint combination of all these data difficulty factors with the class imbalance seriously degrades the recognition of the minority class; see, e.g., experimental studies (Lopez *et al*., 2014; Stefanowski, 2013). Napierala and Stefanowski (2012) have linked some of these data difficulty factors to distinguishing *different types of examples* forming the minority class distribution.

3.1. Types of examples in imbalanced data and the approaches for their identification. Napierala and Stefanowski (2012) proposed to distinguish the following types of examples. *Safe examples* are the ones located in homogeneous regions populated by examples from one class only. Other examples are *unsafe* (categorized into borderline, rare cases and outliers) and more difficult for learning.

To identify the type of a particular example, they analyze the ratio between the number of minority and majority examples in its neighborhood which can be modeled with either k-nearest neighbors or kernel functions. Specific thresholds on this ratio are directly related to particular example types. For instance, if all or nearly all neighbors belong to the same class, the example is treated as a safe example; if the prevalence of both classes inside the neighborhood is quite similar, the example is treated as a borderline one, etc. (Napierala and Stefanowski, 2012; 2016).

Besides using labels which depends on such thresholds, these authors also defined a coefficient expressing a *safe level* of the given example x being a local estimator of the conditional probability of its assignment to the target class as

$$p(C|x) = \frac{k_C}{k}, \tag{1}$$

where C is the class of example x, k is the number of neighbors and k_C is the number of neighbors which belongs to class C. Usually these coefficients are examined for the minority class only, as these are much diversified and smaller while majority examples often have very high safe levels (Napierala and Stefanowski, 2016).

The information about the type of examples have been already successfully applied to binary imbalanced problems, (see, e.g., Błaszczyński and Stefanowski, 2015). Therefore, new multiclass generalizations have been expected (Krawczyk, 2016). These new approaches should also take into account the complexity of different relations between multiple classes. Using existing binary class approaches to estimate data difficulty in such a case is not straightforward (Seaz *et al*., 2016).

3.2. Handling multiple class relations with similarity information. To model relations between multiple imbalanced classes, Lango *et al*. (2017) exploit information about the *similarity* between pairs of classes. This information should be acquired from users being experts in the domain problem. They should say which classes can be seen as more similar to each other than to the rest of the classes. Furthermore, this class similarity may correspond to the expert's interpretation of a mutual position of examples in the neighborhood of the example from a given class. An intuition behind this neighborhood is the following: if example x from a given class has some neighbors from other classes, then neighbors from the class with higher similarity are more preferred.

Consider an illustrative example with three classes: $M1$ and $M2$ minority ones and W majority class. Assume that example x belongs to $M1$. While looking at its 5 neighbors, consider several possible situations which are presented in Table 1.

The neighborhood (a) is the most preferred situation, as example x is surrounded only by examples from its class. In situations (b) and (c), the neighborhoods of example x include one example from class $M1$ and four examples from other classes. Within the binary analysis of example types both these situations will be treated as the same one, however, knowing relations between classes will lead to a different interpretation. If an example x has more neighbors from another minority class $M2$ (situation b) it is more preferred to neighborhood (c) where all surrounding examples come from the distant majority class W. This neighborhood (b) would let the expert consider the analyzed example to be safer—in other terms, easier recognized as a member of its class (as it will be less prone to suffer from the algorithm bias toward the majority classes). The strength of this preference could be expressed by the experts asked to define similarity between classes—stronger between minority classes and

Table 1. Different multiple classes in the neighborhood.

No.	Class $M1$	Class $M2$	Class W
a	5	0	0
b	1	2	2
c	1	0	4

much lower between the minority and majority classes[1].

More formally, it is assumed that for each pair of classes C_i, C_j the degree of their similarity is defined as a real $\mu_{ij} \in [0,1]$. The similarity of a class to itself is defined as $\mu_{ii} = 1$. The degree of similarity does not have to be symmetric, i.e., for some classes C_i, C_j it may happen that $\mu_{ij} \neq \mu_{ji}$. Although the values of μ_{ij} are defined individually for each dataset, the general recommendation of Lango *et al*. (2017) is to have higher simlarities ($\mu_{ig} \to 1$) for other minority classes C_g, while similarities to majority classes C_h should be rather low ($\mu_{ih} \to 0$). This claim is coherent with the earlier experimental studies showing that the multimajority case is more difficult than problems with many minority classes (Wang and Yao, 2012).

The degrees of similarity should be provided by an expert or can come from the domain knowledge. If neither is available, some heuristic approaches could be used. In this work, we propose such an approach which models the situations where one of minority classes suffers from imbalance with respect to the majority class but at the same time may cause imbalances to another, smaller minority class. This leads us to the following definition:

$$\mu_{ij} = \frac{\min(|C_i|, |C_j|)}{\max(|C_i|, |C_j|)}, \tag{2}$$

where $|C_i|$ is the number of examples of C_i class. To better understand our heuristics, consider the classical car UCI dataset which has the classes of the following cardinalities: $|C_{\text{good}}| = 69$, $|C_{\text{vgood}}| = 65$, $|C_{\text{unacc}}| = 1210$ and $|C_{\text{acc}}| = 384$. The similarity between two smallest minority classes is 0.94 and the similarities between the biggest "unacc" class and other minority classes is around 0.05 which is in line with our previous indications. However, the medium size "acc" class may also act as a minority one with respect to "unacc" but at the same time may play a role of a majority class in the proximity of "good" and "vgood" examples. This is reflected in the similarity values assigned by the proposed heuristic: $\mu_{\text{acc,unacc}} = 0.32$, $\mu_{\text{acc,good}} = 0.18$ and $\mu_{\text{acc,vgood}} = 0.17$.

3.3. Data difficulty with respect to a safe level of minority examples. The degrees of similarity have been applied to generalize the identification of the type of examples. Lango *et al*. (2017) generalized the *safe level* coefficient in the following way.

Considering a given example x belonging to the minority class C_i. Its safe level is defined with respect to l classes of examples in its neighborhood as:

$$\text{safe}(x_{C_i}) = \frac{1}{n}\sum_{j=1}^{l} n_{C_j}\mu_{ij} \tag{3}$$

where μ_{ij} is the degree of similarity, n_{C_j} is the number of examples from class C_j inside the considered neighborhood of x and n is a total number of neighbors.

Coming back to the illustrative example from the previous sub-section, calculate the safe level for situations (b) and (c). If we assume that similarity between minority classes $M1$ and $M2$ is equal to 0.5 while their similarity to majority class W is equal to 0, then

$$\text{safe}(x_b) = \frac{1 \times 1 + 0.5 \times 2 + 0 \times 2}{5} = 0.25$$

while

$$\text{safe}(x_c) = \frac{1 \times 1 + 0 \times 4}{5} = 0.2.$$

Thus, the situation (b) is interpreted as slightly safer than its alternative (c). If one increases the minority class similarities up to 0.8, then $\text{safe}(x_b) = 0.52$ and $\text{safe}(x_c)$ will be still 0.2. Thus, the difference in interpreting the safe neighborhood will be much higher. Note that without modeling class similarities the situations (b) and (c) are indistinguishable as their safe levels are the same and equal to 0.2

Lango *et al*. (2017) carried out few experiments with mainly artificial datasets and analyzed averaged safe levels for minority examples together with the predictions of standard classifiers. They showed that this method sufficiently well identifies difficulties in learning these classifiers from the minority classes and their distribution, in particular for class overlapping.

4. Resampling algorithm SOUP

In this paper we want further exploit the approach described in Section 3 to improve classification of multi-class imbalanced data. However, as our aim is to present a kind of a feasibility study rather than looking for the most accurate solution, we have decided to consider a relatively simple and universal pre-processing method.

As a critical motivation for this method we notice that existing multiclass oversampling methods increase class cardinalities to the sizes of majority classes (see Global-CS), which may reinforce difficulties in class distributions, in particular in the case of class overlapping or complex boundaries. Moreover, it may too strongly amplify possible noise of minority class examples with respect to more complicated relations to many other classes. On the other hand, undersampling may be more problematic for imbalanced datasets with a high disproportion between the cardinality of the biggest and the smallest class.

[1]Note that in our proposal of similarity between classes we do not directly model misclassifications between minority classes, which alternatively could be handled by yet another approach with costs of misclassifications between classes (Wojciechowski *et al*., 2017).

Therefore, we have decided to introduce a hybrid resampling algorithm, called *Similarity Oversampling and Undersampling Preprocessing* (SOUP), which combines undersampling with oversampling and exploits the information about the difficulty of examples. Its pseudocode is presented in Algorithm 1. In what follows we shall present a rationale behind SOUP algorithm and describe the proposed approach in more detail.

In SOUP, all majority classes are undersampled and all minority classes are oversampled to the cardinality being the average of the sizes of the biggest minority and the smallest majority class (*line* 3). It is partly inspired by experiences with SCUT undersampling (Agrawal *et al*., 2015). This provides us not only a dataset with a balanced class distribution, but also with a reasonable size.

SOUP exploits the knowledge about the examples' safe levels which were defined in the previous section. The undersampling of the majority classes is performed by removing the most unsafe examples until a desired class cardinality is obtained (*line* 9). In this way, the undersampling process is focused on the examples lying closely to minority examples or inside their regions, which possibly deteriorate minority class recognition. The oversampling of minority classes is performed in the opposite direction, i.e., the safest examples are duplicated first, enhancing the representation of clear minority concepts (*line* 17). In the undesirable situation that there are not enough examples to achieve the requested number of examples even by duplicating the whole class, the list of class examples is processed cyclically from the beginning.

Another aspect of this sampling scheme is that the safe level of a particular example in the final distribution is changing while performing consecutive steps of over- or under sampling for succeeding classes. This leads to establishing a particular order of performing under- and oversampling in SOUP, starting from operations which should have the biggest impact on other examples' safe levels and potentially on the recognition of the minority classes. In this way, undersampling majority classes is done from the biggest to the smallest one (*line* 4). Then, the minority classes are oversampled from the smallest to the biggest one (*line* 12). Note that after each under/oversampling of a class, safe levels of all examples are recomputed.

The calculation of the safe level which takes into account the degrees of similarities (*lines* 7 and 15) as well as the homogeneity of a k neighborhood is performed as proposed by Lango *et al*. (2017) with HVDM distance. This is the most time-consuming element of SOUP.

Furthermore, in order to check how effective our resampling technique is in the combination with the OVO and OVA decomposition approaches, we have developed two separate sampling techniques based on solutions coming from SOUP, which will be applied before learning component binary classifiers. Algorithms 2 and 3 present the pseudocodes of undersampling and oversampling approaches for binary imbalanced data. They were created by extracting and adapting the respective parts from SOUP. Note that in these approaches only two classes are considered and the reference sizes of resampled classes are defined in a new way with respect to a smaller component $D_{\min}$.

Algorithm 1. Similarity Oversampling and Undersampling Preprocessing (SOUP).

Input: D: original training set of $|D|$ examples with c classes; $C_{\min}$: indexes of minority classes; C_{maj}: indexes of majority classes; μ_{ij} similarities between classes
Output: D': balanced training set

1: Split dataset D into c homogeneous parts $D_1, D_2, \ldots, D_c$. Each D_i contains all examples from i class
2: $D' = \emptyset$
3: $m \leftarrow \text{mean}(\min_{i \in C_{\text{maj}}} |D_i|, \max_{j \in C_{\min}} |D_j|)$
4: **for all** $i \in C_{\text{maj}}$ **do**
5: **for all** $x \in D_i$ **do**
6: find k nearest neighbours of x
7: calculate safe level of x, according to Eqn. (3)
8: **end for**
9: remove $|D_i| - m$ examples with the lowest safe level values from D_i
10: $D' \leftarrow D' \cup D_i$
11: **end for**
12: **for all** $j \in C_{\min}$ **do**
13: **for all** $x \in D_j$ **do**
14: find k nearest neighbours of x
15: calculate safe level of x, according to Eqn. (3)
16: **end for**
17: duplicate $m - |D_i|$ examples with the highest safe level values in D_j
18: $D' \leftarrow D' \cup D_j$
19: **end for**
20: **return** D'

5. Experiments

5.1. Experimental setup. We want to experimentally evaluate whether SOUP (which exploits additional information on class similarities and safe levels of examples) may be competitive to existing single preprocessing methods and the ensemble specialized for multiclass imbalance data (which do not use this information). Additionally, we will examine the sensitivity of SOUP with respect to various degrees of class similarity, also including the usefulness of the automatic methods for defining these degrees.

The related standard approaches are Global-CS and Static-SMOTE as representatives of over sampling applied to single classifiers, decomposition with OVA and

Algorithm 2. Similarity Oversampling (SO).

Input: D: original training set of $|D|$ examples with two classes;
Output: D': balanced training set

1: Split dataset D in two parts containing examples from a single class, denoted $D_{\min}$ and D_{maj}
2: $D' = \emptyset$
3: diff $\leftarrow |D_{\text{maj}}| - |D_{\min}|$
4: **if** diff > 0 **then**
5: **for all** $x \in D_{\min}$ **do**
6: find k nearest neighbours of x
7: calculate safe level of x
8: **end for**
9: duplicate 'diff' examples with the highest safe level values from $D_{\min}$
10: **end if**
11: $D' = D_{\text{maj}} \cup D_{\min}$
12: **return** D'

Algorithm 3. Similarity Undersampling (SU)

Input: D: original training set of $|D|$ examples with two classes;
Output: D': balanced training set

1: Split dataset D in two parts containing examples from a single class, denoted $D_{\min}$ and D_{maj}
2: $D' = \emptyset$
3: diff $\leftarrow |D_{\text{maj}}| - |D_{\min}|$
4: **if** diff > 0 **then**
5: **for all** $x \in D_{\text{maj}}$ **do**
6: find k nearest neighbours of x
7: calculate safe level of x
8: **end for**
9: remove 'diff' examples with the lowest safe level values from D_{maj}
10: **end if**
11: $D' = D_{\text{maj}} \cup D_{\min}$
12: **return** D'

OVO ensembles with resampling of the binary classes done with random over sampling (ROS) or random under sampling (RUS) following recommendations of (Fernandez *et al*., 2013) and (Galar *et al*., 2011) and NCR as a more informative under sampling (Laurikkala, 2001), and newly introduced Multi-class Roughly Balanced Bagging, which showed good experimental results in the work of Lango and Stefanowski (2018). To learn component classifiers, we consider three popular algorithms: J4.8 tree, PART rule and k-NN. All of them were used with standard parameters except deactivating pruning options and $k = 3$ following earlier experiments on imbalanced data. All experiments were performed in the WEKA framework. Classification performance is evaluated by a stratified 10-fold cross-validation.

The predictions of all classifiers are evaluated with three measures adapted to the multiclass context: *G-mean, average_minority* and *F-score*. Let *sensitivity*$_i$ be the recognition rate of the local class C_i, then G-mean $= \sqrt[n]{\prod_{i=1}^{n} \text{sensitivity}_i}$; average_minority $= \frac{1}{n} \sum_{i \in C_{\min}} \text{sensitivity}_i$, where $C_{\min}$ denotes minority classes, while n is their number. *F-score* is macro-averaged in a standard way over the sum of *F1_scores* for all minority classes.

5.2. Multiclass datasets. Our experiments are carried out over 19 diversified datasets. Their characteristics are given in Table 2. Firstly, we choose 15 real-world imbalanced data sets coming from the UCI repository, representing a different number of classes, sizes and imbalance ratios, which have been used in the most related experimental studies (Fernandez *et al*., 2013; Galar *et al*., 2011; Seaz *et al*., 2016). We have modified some data by aggregating classes and made decisions on assigning particular classes into minority ones. It resulted in constructing two extra variants of `cleveland` data. Therefore, datasets include from 1 to 5 minority classes. Additionally, we choose 4 synthetic data sets coming from the work of Lango *et al*. (2017), where `art1` is the easiest while `art3` and `art4` more difficult ones. All the considered datasets represent different degrees of difficulty for learning standard classifiers.

5.3. Class similarities and dataset difficulty. To study the influence of modeling various potential relations between classes, we chose six different configurations of the similarity values μ_{ij}, cf. Table 3. Their values model possible various expert understanding of the safer class neighborhood.

SIM1–SIM3 are coming from the earlier study (Lango *et al*., 2017), where, e.g., SIM1 represents an expert's acceptance to potential overlapping between minority classes, while SIM2 adds more acceptance for similarity with majority classes. Then, the last three configurations cover the extreme views on which class similarity could be the most preferred.

In the first step, we evaluate the potential difficulty of class distributions in each dataset by calculating the average values of safe levels independently for minority and majority classes. Due to space limitations we present in Table 5 their values for SIM2 configuration only. One can notice that the chosen datasets represent different categories of difficulty. Following the original interpretations of Napierala and Stefanowski (2016) the datasets with high average values (close to 0.9) should be easier for recognizing classes, see, e.g., `dermatology`, `car`, `vehicle`, `thyroid` or some synthetic data. On the other hand, datasets such as `cleveland`, `cmc`,

Table 2. Characteristics of multi-class imbalanced datasets. Names of classes are given in the first row, while their cardinalities in the second row.

Dataset	Minority classes					Majority classes				
balance-scale	B					L	R			
	49					288	288			
car	good	vgood				unacc	acc			
	69	65				1210	384			
cleveland_1	1	2	3	4		0				
	55	36	35	13		164				
cleveland_2	2	3	4			0+1				
	36	35	13			219				
cmc	2					1	3			
	333					629	511			
dermatology	6					1	2	3	4	5
	20					112	61	72	49	52
ecoli	pp	imUimS	omomL			cpimL	im			
	52	37	25			145	77			
flare	4	5				1	2	3	6	
	116	51				212	287	327	396	
glass	vwf	con	tab			bwf	bwnf	head		
	17	13	9			70	76	29		
hayes_roth	3					1	2			
	31					65	64			
led7digit	5	10				1	2	3	6	
	52	49				98	94	108	99	
new_thyroid	2	3				1				
	35	30				150				
vehicle	bus	van				opel_saab				
	218	199				429				
yeast	2	3	5	6	7	1	8	9	10	
	20	30	35	44	51	463	168	244	429	
wine_quality_red	7	8				5	6			
	199	81				681	638			
art1	MIN1	MIN2				MAJ				
	120	240				840				
art2	MIN1	MIN2				MAJ				
	120	240				840				
art3	MIN1	MIN2				MAJ				
	120	240				840				
art4	MIN1	MIN2				MAJ				
	120	240				840				

`glass`, `yeast` and many others may be very difficult. Furthermore, the choice of similarity degrees influences the average safe levels, in particular for possibly more difficult datasets. For instance, the more difficult synthetic datasets `art3` and `art4` have smaller safe levels for SIM3, i.e., 0.4156 and 0.7527, respectively.

Similarly, real datasets; such as, e.g., `ecoli`, have SIM3 equal 0.4364 and SIM2 is 0.66316; hayes-roth SIM3 is 0.36129 while SIM2 is 0.4571. The similar increases do not occur for easier datasets, see, e.g., `dermatology` SIM3 0.9586 vs. SIM 2 0.96488. Therefore, modeling higher similarity degrees between classes increases safe interpretations of possibly more complex and overlapping classes, which is expected knowing the approach.

5.4. Impact of similarity degrees on the SOUP algorithm. Then we check the influence of different similarity degrees on classification results obtained with SOUP, including SOUP working with heuristic similarity values calculated according to Eqn. (2). Due to space

Table 3. Different configurations of similarity degrees.

Similarity	$\mu_{\text{min1 min2}}$	$\mu_{\text{min maj}}$	$\mu_{\text{maj1 maj2}}$
SIM1	0.8	0	0.1
SIM2	0.7	0.15	0.2
SIM3	0	0	0
SIM4	1.0	0	1.0
SIM5	0	0.5	0
SIM6	1.0	0	0

Table 4. Average safe levels for all minority and all majority classes calculated for SIM2 class similarities.

Dataset	Minority	Majority
balance_scale	0.16388	0.88009
car	0.90716	0.96745
cleveland_1	0.62374	0.83104
cleveland_2	0.51762	0.89210
cmc	0.45580	0.59982
dermatology	0.96488	0.97110
ecoli	0.66316	0.83252
flare	0.48246	0.78493
glass	0.51795	0.74309
hayes_roth	0.45710	0.66891
led7digit	0.76267	0.77206
thyroid	0.85092	0.98073
vehicle	0.89434	0.89142
wine_quality	0.46754	0.64299
yeast	0.56089	0.60316
art1	0.94994	0.96924
art2	0.77986	0.92512
art3	0.61383	0.84882
art4	0.79836	0.90873

limitations, we omit a table with precise results[2] and discuss the main observations.

The first observation is that SOUP in all configurations improves recognition of the minority classes while compared with the baseline. Then, analyzing the G-mean measure for the J4.8 classifier, we observe that differences between SIM configurations on individual datasets are high, especially on very difficult datasets. For instance, differences between SOUP with different similarity degrees on the `cleveland` dataset go up to 30%. Similar observations can be made for other classifiers: 10% for kNN and 35% for PART. Furthermore, SIM configurations influence the classifier performance for various datasets in a diversified way. For instance, using SIM1 or SIM2 leads to better results for such data as `cleveland1`, `balance scale`, while SIM6 works better for `cleveland2` or `wine quality`. Then, SIM5 is the best for `cmc`, `flare` or `glass`. In general using higher degrees of class similarity for difficult data is better than no relations in SIM3. On the other hand, for safer datasets the differences are not considerable, e.g., for `vehicle` or `dermatology` the results are the same up to the third decimal place. Such conclusions also hold for other considered classification measures they for particular datasets; go up to 11% for average minority and up to 8% for F1-measure, both reported for a tree classifier.

Although these differences occurred for particular datasets, a global statistical analysis does not clearly indicate the winning configuration. Following the Friedman rank test, the differences are not significant (the p value equal to 0.36). In further experiments, we will use only one similarity function, namely the heuristic one, as it achieves slightly better results and is adaptive to different datasets. At this point, we would like to emphasize that in practice, expert knowledge may be of key importance to model similarity between classes and to achieve the best results.

We have also tested SOUP variants with different orderings of class sampling as well as with different orders of processing examples with respect to safe levels; however, since they have not led to better results, we do not report them. We noted that changing the oversampling order for minority examples (from the most unsafe ones) is beneficial to the most difficult data, which is consistent with results of Błaszczyński and Stefanowski (2015).

5.5. Comparing related approaches. Before the final comparative experiment for SOUP, we compared only the related approaches for multi-imbalanced problems. The results of G-mean for OVO and OVA decompositions, Global-CS and Static-SMOTE preprocessing methods while using a J4.8 tree as a base classifier are presented in Table 5. In OVO and OVA we applied various resampling methods: random oversampling (ROS), random undersampling (RUS) and NCR. The results of the Friedman test are statistically significant ($p < 0.0001$) indicating differences between methods being investigated. The performed Nemenyi post-hoc analysis is summarized in Fig. 1. The results for other measures and base classifiers are available in the earlier indicated Web page.

The first observation is that OVO based approaches are winners compared with OVA and the baseline. Their advantage is particularly visible for more difficult datasets. The two best performing variants are OVO with random oversampling (average rank of 2.55) and OVO with random undersampling (2.55). Interestingly, the resampling method Global-CS also performs quite well (4.11). OVO NCR and Static-SMOTE are the last methods which have overall performance better than the baseline. The weaker performance of OVA methods goes in tandem

[2]See detailed results on the Web page accompanying this article: `www.cs.put.poznan.pl/mlango/publications/soup.html`. A SOUP implementation is also available there.

Table 5. Comparison of G-mean for decomposition methods and Global-CS for using decision trees as a basic classifier.

Dataset	baseline	Global CS	Static SMOTE	OVO ROS	OVO RUS	OVO NCR	OVA ROS	OVA RUS	OVA NCR
balance_scale	0.000	0.340	0.080	0.526	0.602	0.474	0.302	0.297	0.000
car	0.847	0.940	0.897	0.939	0.876	0.919	0.112	0.184	0.130
cleveland_1	0.227	0.000	0.052	0.255	0.287	0.262	0.254	0.259	0.000
cleveland_2	0.000	0.000	0.037	0.288	0.285	0.000	0.280	0.287	0.000
cmc	0.483	0.478	0.452	0.509	0.514	0.526	0.510	0.511	0.529
dermatology	0.945	0.952	0.927	0.921	0.929	0.948	0.000	0.000	0.000
ecoli	0.728	0.710	0.738	0.805	0.767	0.000	0.000	0.000	0.000
flare	0.446	0.570	0.421	0.544	0.568	0.522	0.000	0.000	0.000
glass	0.625	0.715	0.322	0.699	0.697	0.691	0.000	0.000	0.000
hayes_roth	0.843	0.832	0.835	0.843	0.843	0.838	0.000	0.000	0.000
led7digit	0.786	0.770	0.756	0.771	0.779	0.722	0.120	0.162	0.156
thyroid	0.889	0.922	0.879	0.922	0.886	0.913	0.904	0.927	0.898
vehicle	0.912	0.912	0.915	0.916	0.923	0.915	0.133	0.141	0.164
wine_quality	0.432	0.464	0.356	0.492	0.476	0.434	0.459	0.489	0.356
yeast	0.000	0.406	0.184	0.442	0.479	0.000	0.000	0.000	0.000
art1	0.945	0.961	0.947	0.958	0.949	0.949	0.039	0.000	0.039
art2	0.686	0.734	0.741	0.758	0.777	0.762	0.250	0.253	0.244
art3	0.410	0.534	0.535	0.615	0.612	0.559	0.307	0.304	0.236
art4	0.785	0.829	0.856	0.840	0.872	0.839	0.000	0.000	0.000

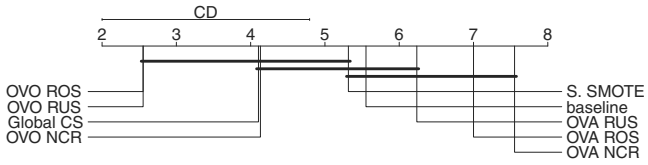

Fig. 1. Vizualization of Nemenyi post-hoc analysis results for preprocesing and decomposition methods.

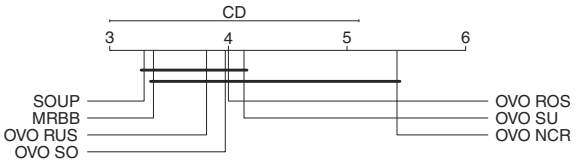

Fig. 2. Vizualization of Nemenyi post-hoc analysis results for SOUP and best ensemble methods.

with the earlier experimental studies by Fernandez *et al*. (2013). However, the differences between the OVA methods and the baseline are not statistically significant according to the post-hoc analysis. Similar observations hold for other investigated classifiers.

5.6. Comparing SOUP with other methods. We first compare SOUP's performance with other preprocessing methods and later proceed with the comparison with the best ensemble methods.

The left part of Table 6 presents the results of G-mean for a tree classifier with different preprocessing methods. Morover, Table 7 presents the results of paired Wilcoxon tests between SOUP and other preprocessing methods.

Both the G-mean values as well as the results of Wilcoxon tests indicate the superiority of SOUP over the other methods. Using typical significance level $\alpha = 5\%$, one can reject the null hypothesis about the lack of differences between SOUP and other preprocessing methods. On the average, SOUP achieves by 5.6% higher results in terms of the G-mean in comparison with the second best preprocessing method (Global-CS) with the J48 classifier. It also yields better results than Static-SMOTE of about 11.6% on the average (median 5.7%). Similarly, for k-NN SOUP outperforms Global-CS by an average of $4,9\%$ (median 1.3%) and even for PART, where the results are not statistically significant, SOUP achieves better results by about 0.8% (both median and average).

Then, we proceed with the comparison of SOUP with the best three performing methods from the previous experiment: OVO RUS, OVO ROS and OVO NCR. We have also added to this final comparison the best performing method from our earlier studies (Lango and Stefanowski, 2018; Lango, 2019), namely, the undersampling version of Multi-class Roughly Balanced Bagging (MRBB). Additionally, we ininvestigated the performance of new combinations of OVO with the introduced resampling, i.e., SO and SU variants. The

Table 6. Comparison of best methods and SOUP with the tree J48 algorithm and G-mean.

Dataset	Global CS	Static SMOTE	SOUP	OVO ROS	OVO RUS	OVO NCR	OVO SO	OVO SO	MRBB
balance_scale	0.340	0.080	0.585	0.526	0.602	0.474	0.542	0.547	0.683
car	0.940	0.897	0.941	0.939	0.876	0.919	0.940	0.794	0.907
cleveland_1	0.000	0.052	0.266	0.255	0.287	0.262	0.268	0.302	0.021
cleveland_2	0.000	0.037	0.303	0.288	0.285	0.000	0.284	0.312	0.055
cmc	0.478	0.452	0.535	0.509	0.514	0.526	0.522	0.524	0.517
dermatology	0.952	0.927	0.962	0.921	0.929	0.948	0.925	0.939	0.959
ecoli	0.710	0.738	0.735	0.805	0.767	0.000	0.791	0.739	0.768
flare	0.570	0.421	0.566	0.544	0.568	0.522	0.582	0.506	0.542
glass	0.715	0.322	0.667	0.699	0.697	0.691	0.701	0.697	0.400
hayes_roth	0.832	0.835	0.835	0.843	0.843	0.838	0.843	0.775	0.823
led7digit	0.770	0.756	0.778	0.771	0.779	0.722	0.765	0.704	0.778
thyroid	0.922	0.879	0.922	0.922	0.886	0.913	0.897	0.896	0.932
vehicle	0.912	0.915	0.915	0.916	0.923	0.915	0.904	0.880	0.943
wine_quality	0.464	0.356	0.471	0.492	0.476	0.434	0.524	0.490	0.525
yeast	0.406	0.184	0.451	0.442	0.479	0.000	0.000	0.484	0.201
art1	0.961	0.947	0.960	0.958	0.949	0.949	0.959	0.951	0.960
art2	0.734	0.741	0.777	0.758	0.777	0.762	0.754	0.804	0.808
art3	0.534	0.535	0.608	0.615	0.612	0.559	0.627	0.634	0.631
art4	0.829	0.856	0.899	0.840	0.872	0.839	0.831	0.878	0.893

Table 7. p-Values of the paired Wilcoxon signed rank test between SOUP and other preprocessing methods on G-mean measure for various classifiers.

Alg.	baseline	Global-CS	Static-SMOTE
J4.8	< 0.001	0.036	< 0.001
PART	< 0.001	0.153	< 0.001
kNN	< 0.001	0.005	0.002

Table 8. Average rank of compared algorithms (the lower, the better) from Friedman tests on G-mean measure for various classifiers.

Alg.	SOUP	MRBB	OVO RUS	OVO SO	OVO ROS	OVO SU	OVO NCR
J4.8	3.29	3.37	3.82	3.97	4.00	4.13	5.42
PART	3.8	3.05	4.05	4.0	4.45	4.2	4.45
kNN	3	3.95	4.1	4.45	4.55	3.85	4.1

results of the G-mean for a tree classifier are presented in Table 6.

Note that from the Friedman test did not reject the null hypothesis on equal performance of all classifiers with $p = 0.058$, although it is nearly at the typical confidence level. The Nemenyi post-hoc analysis of average ranks is presented on Fig. 2. The best method in our comparison, according to the average rank, is SOUP and the next is the MRBB method. Following the Wilcoxon test, the differences between these methods are insignificant ($p = 0.45$). According to average ranks, SOUP outperforms all decomposition approaches, although it uses one classifier only. The third best method is the combination of OVO with random undersampling. We also observed that new resampling is more useful for oversampling: OVO SO is always better than OVO ROS in Table 8, while it is not the case for OVO RUS.

Results for kNN are also favorable for SOUP since it has the lowest rank in such a comparison. For this component classifier, our extensions of OVO outperform their random counterparts. Interestingly, the position of MRBB is lower in this ranking. We relate this to the fact that kNN is a rather stable classifier while bagging-based algorithms work better with more unstable classifiers like trees or rules. For instance, MRBB with PART component classifiers again achieves the best ranks (even better than SOUP). Other top-performing methods for PART are SOUP and OVO SO.

We have also analyzed the size of the constructed trees (expressed by the number of nodes) for two less safe datasets: `flare` and `yeast`. The application of SOUP on the `flare` dataset resulted in the construction of trees which were, on the average, twice as large as the baseline tree without any pre-processing method. Conversely, the trees in the OVO approaches had considerably smaller sizes. However, since those methods required many trees to be constructed, the sum of tree sizes exceeded the size of SOUP's tree almost four times. Another observation was that the trees for undersampling approaches was always smaller than those for oversampling methods. Regarding the `yeast` dataset, the results were quite similar with the exception that SOUP constructed a tree

slightly smaller (341.6 nodes) than the baseline (350.8). As SOUP requires only one classifier to be constructed, it helps human to interpret it easier than complex ensembles without significantly sacrificing predictive abilities.

6. Conclusions

In this work, we have considered a new approach to deal with multiclass imbalanced data, wherein interrelations between classes are modeled by means of analyzing the neighborhood of minority examples and taking into account an expert's information about the degrees of similarity between classes. It estimates the examples' safe levels which indicate to what extent these examples are problematic for learning an accurate classifier.

The main contribution of our paper is demonstrating that those safety coefficients can be efficiently exploited in resampling techniques to improve classifiers. To this end, we have introduced a new preprocessing algorithm SOUP, whose key elements are a resampling with respect to examples' safe levels and a particular ordering of undersampling majority classes and oversampling minority ones.

Its experimental evaluation has clearly shown that defining similarity degrees influences the estimation of the multiclass dataset difficulty. Moreover, increasing these degrees between minority classes improves SOUP classification of the most unsafe datasets. SOUP with all considered configurations of similarity degrees has outperformed baseline, no-preprocessing classifiers. It also works significantly better than Static-SMOTE and Global-CS two—popular preprocessing methods for multiclass imbalances. The next comparative experiments have demonstrated that SOUP can be slightly better then MRBB—one of the best bagging ensembles for some types of component classifiers. SOUP is also better than OVO decompositions which are the most frequently recommended in the literature. Additionally, the components of SOUP preprocessing have demonstrated to be useful to improve OVO ensembles, mainly in the case of oversampling. Finally, unlike the complex structure of ensembles, SOUP results in a much smaller single classifier, which may be more interpretable for humans while using, e.g., tree classifiers.

Nevertheless, as a future research direction, we plan to use SOUP inspirations in generalizing an underbagging ensemble, such as Neighborhood Balanced Baging, in order to further improve predictive ability.

Acknowledgment

This research was supported by statutory funds of the Institute of Computing Science of the Poznan University of Technology. We are also grateful to Krystyna Napierala for her remarks on the earlier version of our work.

References

Abdi, L. and Hashemi, S. (2016). To combat multi-class imbalanced problems by means of over-sampling techniques, *IEEE Transactions on Knowledge and Data Engineering* **28**(1): 238–251.

Agrawal, A., Herna, L.V. and Paquet, E. (2015). SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling, *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Lisbon, Portugal*, Vol. 01, pp. 226–234.

Błaszczyński, J. and Stefanowski, J. (2015). Neighbourhood sampling in bagging for imbalanced data, *Neurocomputing* **150**(Part B): 184–203.

Fernandez, A., Lopez, V., Galar, M., Jesus, M. and Herrera, F. (2013). Analysing the classification of imbalanced data sets with multiple classes, binarization techniques and ad-hoc approaches, *Knowledge-Based Systems* **42**: 97–110.

Fernández, A., Garca, S., Galar, M., Prati, R., Krawczyk, B. and Herrera, H. (2018). *Learning from Imbalanced Data Sets*, Springer, Cham.

Fernandez-Navarro, F., Hervás-Martínez, C. and Gutiérrez, P. A. (2011). A dynamic over-sampling procedure based on sensitivity for multi-class problems, *Pattern Recognition* **44**(8): 1821–1833.

Galar, M., Fernndez, A., Barrenechea, E., Bustince, H. and Herrera, F.A. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, *Pattern Recognition* **44**(8): 1761 – 1776.

Garcia, V., Sanchez, J.S. and Mollineda, R.A. (2007). An empirical study of the behaviour of classifiers on imbalanced and overlapped data sets, *in* L. Rueda *et al.* (Eds), *Progress in Pattern Recognition, Image Analysis and Applications*, Lecture Notes on Computer Science, Vol. 4756, Springer, Berlin, pp. 397–406.

He, H. and Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*, Wiley, New York, NY.

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

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions, *Progress Artificial Intelligence* **5**(4): 221–232.

Lango, M. (2019). Tackling the problem of class imbalance in multi-class sentiment classification: An experimental study, *Foundations of Computing and Decision Sciences* **44**(2): 151–178.

Lango, M., Napierala, K. and Stefanowski, J. (2017). Evaluating difficulty of multi-class imbalanced data, *23rd International Symposium ISMIS, Warsaw, Poland*, pp. 312–322.

Lango, M. and Stefanowski, J. (2018). Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data, *Journal of Intelligent Information Systems* **50**(1): 97–127.

Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution, *Technical Report A-2001-2*, University of Tampere, Tampere.

Lopez, V., Fernandez, A., Garcia, S., Palade, V. and Herrera, F. (2014). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, *Information Sciences* **257**: 113–141.

Napierala, K. and Stefanowski, J. (2012). The influence of minority class distribution on learning from imbalance data, *Proceedings of the 7th Conference HAIS 2012, Salamanca, Spain*, pp. 139–150.

Napierala, K. and Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data, *Journal of Intelligent Information Systems* **46**(3): 563–597.

Napierala, K., Stefanowski, J. and Wilk, S. (2010). Learning from imbalanced data in presence of noisy and borderline examples, *in* M. Szczuka *et al.* (Eds), *Proceedings of the 7th International Conference RSCTC 2010,* Lecture Notes on Artificial Intelligence, Vol. 6086, Springer, Berlin, pp. 158–167.

Prati, R., Batista, G. and Monard, M. (2004). Class imbalance versus class overlapping: An analysis of a learning system behavior, *in* R. Monroy *et al.* (Eds), *Advances in Artificial Intelligence, MICAI 2004*, Lecture Notes in Computer Science, Vol. 2972, Springer, Berlin/Heidelberg, pp. 312–321.

Seaz, J., Krawczyk, B. and Wozniak, M. (2016). Analyzing the oversampling of different classes and types in multi-class imbalanced data, *Pattern Recognition* **57**: 164–178.

Stefanowski, J. (2013). Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data, *in* S. Ramanna *et al.* (Eds), *Emerging Paradigms in Machine Learning*, Smart Innovation, Systems and Technologies, Vol. 13, Springer, Berlin/Heidelberg, pp. 277–306.

Stefanowski, J. (2016). Dealing with data difficulty factors while learning from imbalanced data, *in* J. Mielniczuk (Eds), *Challenges in Computational Statistics and Data Mining*, Studies in Computational Intelligence, Vol. 605, Springer, Cham, pp. 333–363.

Stefanowski, J., Krawiec, K. and Wrembel, R. (2017). Exploring complex and big data, *International Journal of Applied Mathematics and Computer Science* **27**(4): 669–679, DOI: 10.1515/amcs-2017-0046.

Wang, S. and Yao, X. (2012). Mutliclass imbalance problems: Analysis and and potential solutions, *IEEE Transactions Systems, Man and Cybernetics, B* **42**(4): 1119–1130.

Wojciechowski, S., Wilk, S. and Stefanowski, J. (2017). An algorithm for selective preprocessing of multi-class imbalanced data, *International Conference on Computer Recognition Systems, CORES 2017, Polanica Zdrój, Poland*, pp. 238–247.

Zhou, Z.H. and Liu, X.Y. (2010). On multi-class cost sensitive learning, *Computational Intelligence* **26**(3): 232–257.

Małgorzata Janicka is a software engineer working on big data solutions for autonomous driving. She received her MSc degree from the Poznań University of Technology in 2018, presenting a thesis focused on multi-class imbalanced classification. Her research interests include machine learning on big datasets.

Mateusz Lango received his BScEng and MSc degrees in computer science from the Poznań University of Technology. He is currently working towards his PhD degree at this university. His research interests include learning from imbalanced data, ensemble models and its applications in natural language processing.

Jerzy Stefanowski is an associate professor in the Institute of Computing Science at the Poznań University of Technology. His research interests include machine learning, data mining and intelligent decision support, in particular rule induction, multiple classifiers, class imbalance, concept drift, classification of data streams, classification of big data and handling uncertain data.

Received: 14 December 2018
Revised: 15 June 2019
Accepted: 12 July 2019