ImWeights: Classifying Imbalanced Data Using Local and Neighborhood Information

Mateusz Lango MLANGO@CS.PUT.EDU.PL
Dariusz Brzezinski DBRZEZINSKI@CS.PUT.EDU.PL
Jerzy Stefanowski JSTEFANOWSKI@CS.PUT.EDU.PL
Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznań, Poland

Editors: Luís Torgo, Stan Matwin, Nathalie Japkowicz, Bartosz Krawczyk, Nuno Moniz, and Paula Branco

Abstract

Preprocessing methods for imbalanced data transform the training data to a form more suitable for learning classifiers. Most of these methods either focus on local relationships between single training examples or analyze the global characteristics of the data, such as the class imbalance ratio in the dataset. However, they do not sufficiently exploit the combination of both these views. In this paper, we put forward a new data preprocessing method called ImWeights, which weights training examples according to their local difficulty (safety) and the vicinity of larger minority clusters (gravity). Experiments with real-world datasets show that ImWeights is on par with local and global preprocessing methods, while being the least memory intensive. The introduced notion of minority cluster gravity opens new lines of research for specialized preprocessing methods and classifier modifications for imbalanced data.

Keywords: Imbalanced data, example weighting, grid clustering, over-sampling

1. Introduction

The predictive performance of most classifiers considerably deteriorates when learned from imbalanced data. In response to this issue, over the last decades researchers have proposed various specialized data preprocessing methods and classifier modifications that tackle skewed data distributions (Branco et al., 2016; He and Yungian, 2013). Nevertheless, the problem of learning from imbalanced data is still considered a challenge, both from a research and application perspective (Krawczyk, 2016).

Although first studies on imbalanced data have focused on the *global* view at the data, expressed e.g. by the imbalance ratio between the minority and majority class, it has since been shown that *local* data difficulty factors also play a crucial part in the challenging nature of skewed datasets. Local difficulty factors refer to internal characteristics of class distributions in the sub-regions of data or example neighborhoods, such as the decomposition of the minority class into many sub-concepts (Japkowicz and Stephen, 2002; Jo and Japkowicz, 2004), overlapping between the classes (Garcia et al., 2007), and presence of many minority class examples inside the majority class region (Napierała et al., 2010).

Both global as well as local characteristics have been used to propose specialized data preprocessing methods. Simple resampling methods, such as Random Oversampling, use the global imbalance ratio to level class cardinalities by multiplicating examples. On the other hand, methods such as variants of SMOTE (Chawla et al., 2002) or ADASYN (He et al., 2008) analyze pairs of examples to control the level of oversampling in different regions of the attribute space. However, local and global information has been mostly used separately or combined without providing any smooth spectrum between knowledge about pairs of examples and entire datasets.

Here, we consider yet another view at exploiting local and global information. We propose a new method for preprocessing imbalanced data called ImWeights, which uses example weighting to combine information about the local difficulty of single examples with knowledge about neighboring clusters and class proportions. To achieve this goal, we use a recently proposed clustering algorithm called ImGrid (Lango et al., 2017) to calculate example *safety* based on class proportions and combine it with the concept of *gravity*, which is emitted by neighboring minority class regions. We will show that ImWeights provides a smooth spectrum of weights by using local and global data characteristics. Moreover, we will experimentally compare ImWeights with popular global and local oversampling strategies on benchmark imbalanced datasets.

The remainder of the paper is organized as follows: related literature is discussed in Section 2, the proposed ImWeights algorithm is described in Section 3, experimental results are discussed in Section 4, and finally conclusions and lines of future research are drawn in Section 5.

2. Related works

The proposed algorithm is a data preprocessing method that takes into account local neighborhood information and data difficulty factors. Section 2.1 discusses existing works on local preprocessing methods for imbalanced data, whereas Section 2.2 describes taxonomies and clustering algorithms concerning data difficulty factors.

2.1. Preprocessing methods using local information

Preprocessing of imbalanced data transforms it to a form (example distribution) more suitable for learning accurate classifiers. According to (Branco et al., 2016), the main preprocessing strategies can be categorized into: re-sampling methods, adaptations of active learning, and example weighting. In this paper, we focus on re-sampling and weighting strategies that modify the available data distribution, usually into a more class-balanced one. According to several studies, such balancing may improve classifier predictions (Weiss and Provost, 2003).

Up to now, many re-sampling techniques have been introduced; for their comprehensive review see (Branco et al., 2016; He and Garcia, 2009). The most popular approaches are random under- and over-sampling. The first approach simply removes examples from the majority classes until a required degree of balance between class cardinalities is reached. On the other hand, over-sampling randomly adds copies of minority class examples or generates new synthetic minority examples. Although simple random re-sampling methods help in some problems and can be quite efficient, particularly in changing training sets for specialized ensembles for imbalanced data, in general it is claimed that they are not sufficiently good at improving the recognition of imbalanced classes of single classifiers.

For instance, random under-sampling may potentially remove some important examples, whereas simple over-sampling may lead to overfitting (He and Garcia, 2009). Therefore, several researchers have focused their interests on *informed re-sampling* methods, which take into account local information about particular example positions in attribute space.

For instance, informed under-sampling is often realized by removing potentially harmful majority examples, in particular noisy or class overlapping instances. Such an approach is often based on exploiting local information by analyzing relations between minority and majority examples, e.g., with Tomek-links (Tomek, 1976) or Edited Nearest Cleaning Rule (Laurikkala, 2001). The Tomek-links method removes examples from overlapping minority-majority regions. On the other hand, the Nearest Cleaning Rule removes majority outliers and examples leading to wrong re-classification of minority examples. These ideas also appear in hybrid methods such as SPIDER (Napierała et al., 2010), which selectively filters out harmful examples from the majority class and amplifies difficult minority examples.

Probably the best-known oversampling method is SMOTE (Chawla et al., 2002). SMOTE oversamples the minority class by generating new synthetic examples also in the local perspective, although with a global parameter referring to the imbalance ratio. It considers each minority class example as a seed and finds its k-nearest neighbors from the minority class. Then, according to the user-defined over-sampling ratio, for each minority example SMOTE randomly selects one of its k neighbors and introduces a new example along the line connecting the seed example with the selected neighbor.

As the basic version of SMOTE blindly generates these minority examples without considering positions of the majority examples, it has been generalized in many ways (Fernández et al., 2018). Some of these generalizations also exploit local neighborhood. For instance, Borderline SMOTE (Han et al., 2005) focuses on oversampling the difficult examples located around decision boundaries, and skips examples that are far from this borderline. The borderline examples are identified by using the local ratio between the majority and minority examples within the neighborhood of each minority candidate for oversampling. The other perspective of exploiting such local ratios is presented in ADASYN (He et al., 2008), which dynamically modifies the amount of over-sampling depending on the difficulty of the minority examples. More precisely, for each minority example its difficulty is defined based on the ratio of majority examples in its neighborhood. Then, given a global balancing rate, the number of new synthetic examples to be generated around each minority class is calculated with respect to its difficulty ratio.

Yet another idea of exploiting local information is to force the algorithm to focus on examples which are most difficult to learn and use an energy-based analogy for pushing the majority examples out of the minority example neighborhood. Such an approach was recently put forward in the Combined Cleaning and Resampling (CCR) algorithm (Koziarski and Woźniak, 2017).

Finally, several other approaches for class imbalanced data use clustering algorithms. First proposals concern cluster-based oversampling (Jo and Japkowicz, 2004), which addresses both the global imbalance between classes and internal class decompositions into small disjuncts. More recently, other clustering algorithms were applied in informed resampling, e.g. MWMOTE or DBSMOTE (Fernández et al., 2018).

On the other hand, example weighting for imbalanced data is typically used in the context of ensemble models and often related to the cost-sensitive learning methodology. Chen et al. (2004) used a more global concept of class weight to improve minority class recognition by Random Forests. Class weight was used in the calculation of tree split criterion to prefer clearer splits of the minority class. It was estimated by out-of-bag estimate or manually tuned. Example weighting is also used in boosting approaches for imbalanced data. For example, Wang and Japkowicz (2010) proposed Boosting-SVM with Asymmetric Cost and a classifier-independent cost-sensitive boosting is described in (Sun et al., 2007). Our weighting approach significantly differs from those proposed in the literature since 1) it is not a modification of any specific learning algorithm and 2) it does not use the notion of classification cost.

2.2. Algorithms for discovering local difficulty factors from imbalanced data

Most research on improving classifiers learned from imbalanced data has been focused on developing new algorithms, while less effort has been put into studying the data *characteristics* that make learning from imbalanced data so difficult. Nonetheless, researchers have already demonstrated the high impact of the following factors: decomposition of the minority class into many sub-concepts, overlapping between classes, and presence of many minority class examples inside the majority class region. When these data difficulty factors occur *together* with class imbalance, they may seriously deteriorate the recognition of the minority class (Lopez et al., 2014; Napierała et al., 2010).

The authors of (Napierala and Stefanowski, 2012) differentiate between *safe* and *unsafe* minority instances. Unsafe examples are further categorized into *borderline*, *rare cases*, and *outliers*. Experimental studies (Napierala and Stefanowski, 2016) show that the identification of dominating types of examples may be useful during assessing the difficulty of imbalanced datasets, interpreting differences between preprocessing methods, and developing new specialized algorithms for imbalanced data (Stefanowski, 2016).

However, most current algorithms for detecting data difficulties focus on single factors, rather than on discovering multiple difficulties at once. Therefore, we direct our interest to the recent proposal of the ImGrid (Imbalanced Grid) algorithm that attempts to simultaneously uncover sub-concepts in complex imbalanced data and categorize types of examples inside these detected clusters (Lango et al., 2017). Since ImGrid is a crucial part of the proposed ImWeights algorithm, we will describe it in more detail.

ImGrid is inspired by grid clustering algorithms and involves: 1) dividing the attribute space into grid cells, 2) joining similar adjacent cells taking into account their minority class distributions, 3) labeling examples according to difficulty factors, 4) forming minority sub-clusters.

First, Imgrid divides the attribute space into equally wide intervals. In order to obtain enough instances for joining adjacent cells, the authors of ImGrid propose to estimate the number of intervals as $\lceil \sqrt[m]{|D|/10} \rceil$, where $|D|$ is the number of examples in the dataset and m is the number of dimensions of the attribute space. Next, the joining step of ImGrid takes into account the probability distribution of minority and majority class examples in adjacent grid cells. It merges them only if the distributions of the classes are similar according to Jeffreys' Bayesian test (Jeffreys, 1935). As a result merged cells constitute candidate clusters. In the next step of ImGrid, each cluster is assigned to one of four difficulty labels: safe, borderline, rare, or outlier (Napierala and Stefanowski, 2016). These labels are estimated based on the local proportion of minority examples to all examples in the cluster. Finally, having a preliminary clustering that divides the data into sub-regions of different difficulties, adjacent cells containing minority examples are joined into minority sub-clusters.

ImGrid has been experimentally validated and compared with imbalanced adaptations of DBSCAN and k-means clustering algorithms on a large collection of artificial datasets with hidden class distribution structures. These results have demonstrated that ImGrid, rediscovers simulated clusters and types of minority examples on par with competing methods, while being the least sensitive to parameter tuning. However it has not been applied to real data and its output information has not been considered in any data preprocessing method for imbalanced data. Therefore, in the following section we put forward an algorithm that uses ImGrid to enhance classifier performance on real-world imbalanced data.

3. ImWeights

We propose a new preprocessing algorithm based on example weighting which we call ImWeights. The proposed method combines information about the local difficulty of examples with knowledge about the vicinity of safe minority clusters. The example weights are determined by using a concept of *safety*, which is defined by the ImGrid algorithm (Lango et al., 2017), and *gravity*, which is emitted by neighboring regions with a force proportional to their safety. The safety of an the example is defined as the ratio of the minority examples to all examples in a grid cell computed by ImGrid. The proposed ImWeights algorithm combines the concepts of safety and gravity using the following formula:

$$w_x = 1 + f(safety(x))(1 + gravity(x)) \tag{1}$$

where x is the example being weighed, and $f()$ is a scaling function. Both the $gravity()$ and $f(safety())$ take values from zero to one, therefore the final weight of a minority example is from one to three. The pseudo-code for ImWeights is presented in Algorithm 1.

First, the proposed method executes the ImGrid clustering algorithm (line 1), which outputs a grid of cells containing information about example difficulty factors (types of minority examples based on class proportions), minority clusters, and relations between them. Next, for each minority example the gravity force is calculated (lines 3–10) and used to compute the examples' weights according to Eq. 1 (line 11). Lastly, the majority examples receive weights that, in sum, balance out all the weights of minority examples (lines 15–17).

The weight formula (Eq. 1) has essentially two parts. The first part describes the weight amplification which depends on the cluster's safety level. The weight is further augmented by the second part which depends on the gravity coming from neighboring cells. Note that this formula guarantees that the gravity can only amplify the example's weight. On the other hand, locally calculated safety has the ability to entirely eliminate the impact of gravity.

```
Algorithm 1: ImWeights
Input: $D$: $m$-dimensional dataset, $\alpha$: threshold for statistical test (ImGrid parameter)
Output: dataset $D$ with assigned weights
$grid \leftarrow$ ImGrid(D, $\alpha$)
for $cell \in grid$ do
    for $x \in cell.minority\_examples$ do
        $x.gravity \leftarrow 0$
        for $k \in \{1, 2, .., m\}$ do
            $position \leftarrow \frac{x[k] - cell.min\_value[k]}{cell.max\_value[k] - cell.min\_value[k]}$
            $x.gravity \leftarrow x.gravity + position * cell.right\_neighbour[k].safety$
            $x.gravity \leftarrow x.gravity + (1 - position) * cell.left\_neighbour[k].safety$
        end
        $x.gravity \leftarrow \frac{1}{m} * x.gravity$
        $x.weight \leftarrow 1 + f(cell.safety) * (1 + x.gravity)$
    end
end
$minority\_weights\_sum \leftarrow \sum_{x \in grid.minority\_examples} x.weight$
for $x \in grid.majority\_examples$ do
    $x.weight \leftarrow \frac{minority\_weights\_sum}{|grid.majority\_examples|}$
end
return grid.majority_examples ∪ grid.minority_examples
```

To control the effect of the example's safety level on its weight, we use a scaling function defined as follows:

$$f(x) = \max\{0, \min\{1, -2.5x + 1.75\}\} \tag{2}$$

This function is equal to 1 for safety ranging from 0 to 0.3, then drops linearly to achieve its minimum equal to 0 for a safety of 0.7. The rationale behind this function is the assumption that the more unsafe a cluster is, the more the example weight should be increased. However, focusing the learning algorithm on outliers too extensively can lead to overfitting and cause an adverse effect. Similarly, safe examples usually do not pose a big challenge to the learning methods, hence, their weight does not need to be inflated. Following previous works (Napierala and Stefanowski, 2016), a local neighborhood with safety higher than 0.7 is considered a safe one. For such values of safety the scaling function (Eq. 2) returns zero and, as a consequence, does not boost the weight of safe examples in any way. Then, the scaling function linearly increases in the range of safety between 0.7 and 0.3, which is usually attributed to borderline examples. Finally, the function is truncated at 1 for examples with safety lower then 0.3 i.e. rare and outlier examples.

The gravity for a given example is calculated by taking the average of gravities emitted by all adjacent cells. The influence of a single cell's gravity on a particular example depends on two factors: the cell safety and the distance between the cell border and the example. At its borders a cell emits gravity which is equal to its safety. As the gap between the position of the weighted example and the cell border increases, its intensity drops linearly until the border of the next cell is reached.

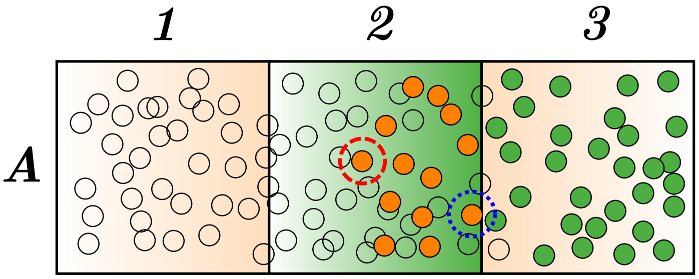

Figure 1: An example dataset divided into grid cells. The minority examples (filled points) are colored according to their types: safe (green) and borderline (orange). The intensity of the background indicates the value of gravity.

Let us illustrate the concept of gravity with an example (Figure 1). We will compute the gravity that affects the leftmost minority instance in cell $A2$ (red dashed circle). Cell $A2$ contains 11 minority examples and 20 majority example, hence its safety is equal to $\frac{11}{20+11} = 0.35$. This cell has two neighbors $A1$ and $A3$ with safety equal 0% and 97%, respectively. Hence, the cell $A1$ does not emit any gravity. The leftmost minority example is located in approximately 40% of the cell's width, thus the influence of gravity is equal to $40\% \cdot 0.97 + 60\% \cdot 0 = 0.39$. For comparison, the gravity force on the rightmost example of $A2$ (blue dotted circle), which is closer to the $A3$ gravity emitter, will be 0.95.

The main purpose of using gravity to augment weights is to identify and highlight to the classifier examples which lie in a dangerous region with safe vicinity. Those examples are especially important because they lie between a homogeneous minority concept and majority examples, marking a border between classes. Also, they can exhibit an underrepresented part of a safe minority concept which can be influential for the construction of a classifier. By taking into account both the local class proportions and the distance to neighboring minority clusters ImWeights attempts to provide richer information than traditional local preprocessing methods. Figure 2 exemplifies this in cell $C2$ where examples will be treated differently than in $A2$ as the neighbors of $C2$ belong to quite safe regions.

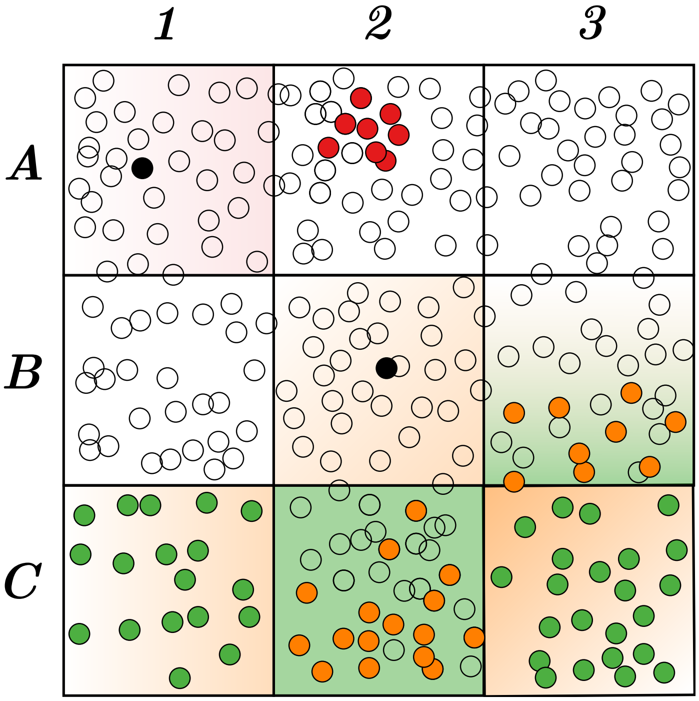

Figure 2: The vizualization of ImGrid clustering with detected types of minority examples: safe (green), borderline (yellow), rare (red) and outlier (black). The intensity of the background indicates the value of gravity.

Contrary to several approaches that use local information about an example's difficulty, ImWeights is not based on k nearest example analysis, making it less susceptible to potential noise. Moreover, the used ImGrid clustering does not require the definition of the expected number of clusters and defines a neighborhood relation between clusters. This relation is exploited by ImWeights to enhance local information about an example's difficulty with the concept of gravity. Furthermore, unlike approaches such as SMOTE and ADASYN the proposed concept of gravity can smoothly weight minority examples, and does so also for regions neighboring with a safe area. Finally, since ImWeights is a weighting scheme it is much more memory efficient than the aforementioned oversampling methods.

4. Experimental evaluation

Since ImWeights exploits both a local data view and more general information about neighboring clusters, we have investigated its performance against popular approaches which use knowledge about global and local data characteristics. Random Oversampling (ROS) represents a simple approach which makes use of the global imbalance ratio only. We have chosen Borderline-SMOTE2 (Han et al., 2005) and ADASYN (He et al., 2008) as representatives of methods which also exploit local data characteristics by analyzing k nearest neighbors.

The goal of the experiment is to verify whether combining both global and local information with additional knowledge about characteristics of adjacent clusters may positively influence classification performance of single classifiers.

As classifiers we have chosen Logistic Regression and Naive Bayes with default scikit-learn parameters. Both algorithms incorporate examples' weights into the learning procedure in a different way. Logistic Regression optimizes a loss function which is basically an average of losses attributed to the classification errors of each example. The weights are integrated into the loss function by simply changing the average into a weighted one. On the other hand, the examples' weights are used in Naive Bayes to modify the estimates of a class prior and data likelihood probability. In the classical version of the classifier, the class prior probability is calculated as the number of class examples divided by the size of the dataset. In the weighted version, the class prior probability is computed by taking a proportion of the sum of examples' weights which belong to a given class to the total sum of all weights. The weights are incorporated into likelihood probability estimates in an analogous way.

Experiments were carried out on 12 real-word datasets from the UCI repository (Table 1). The selected datasets represent a variety of different characteristics of imbalanced data and were often used in earlier papers. Due to the limitations of the ImGrid clustering algorithm, only numerical features of the selected datasets were used in experiments.

Dataset	# examples	# attrib.	IR	Difficulty type
breast-w	699	9	1.90	safe
vehicle	846	18	3.25	safe
new-thyroid	215	5	5.14	safe
pima	768	8	1.87	borderline
haberman	306	3	2.78	borderline
ecoli	336	7	8.60	borderline
transfusion	748	4	3.20	rare
yeast	1484	8	28.10	rare
glass	214	9	12.59	rare
seismic-bumps	2584	11	14.2	rare/outlier
abalone	4177	7	11.47	outlier
balance-scale	625	4	11.76	outlier

Table 1: Datasets characteristics.

We decided to use two common measures of classifier performance for imbalance data, namely G-mean and the recall of the minority class (Japkowicz and Shah, 2011). The recall of minority class was selected as a measure of the classifier's ability to predict minority examples, which are usually of particular interest for the user. On the other hand, G-mean computes the trade-off between the satisfactory recognition of majority and minority class examples. G-mean is considered to be easily interpretable and has better theoretical properties than other classification measures for class imbalanced problems (Brzezinski et al., 2018). The definition of minority class recall and G-mean can be found e.g. in (He and Garcia, 2009).

All reported results were averaged over 10 runs of 5-fold cross-validation. The experiments were performed using the *scikit-learn* (Pedregosa et al., 2011) library and its extension for imbalanced data *imbalanced-learn* (Lemaître et al., 2017).

	G-mean					Difference with ImWeights			
Dataset	Baseline	ImWt.	ROS	B-SM.	ADA.	Baseline	ROS	B-SM.	ADA.
abalone	0.189	0.744	**0.769**	0.760	0.766	0.555	-0.025	-0.016	-0.022
balance-scale	0.000	0.265	0.328	**0.516**	0.387	0.265	-0.063	-0.251	-0.123
breast-w	0.957	0.962	0.961	**0.969**	0.967	0.005	0.001	-0.007	-0.004
ecoli	0.169	0.863	**0.875**	0.841	0.867	0.695	-0.012	0.023	-0.004
glass	0.000	**0.673**	0.569	0.609	0.578	0.673	0.103	0.064	0.095
haberman	0.392	0.640	0.643	0.622	**0.650**	0.249	-0.003	0.018	-0.010
new-thyroid	**0.997**	0.989	0.994	0.976	0.994	-0.008	-0.006	0.013	-0.005
pima	0.694	**0.761**	0.752	0.743	0.748	0.067	0.009	0.018	0.013
seismic-bumps	0.448	**0.582**	0.306	0.333	0.344	0.134	0.276	0.249	0.238
transfusion	0.504	0.664	0.650	0.656	**0.668**	0.160	0.014	0.008	-0.005
vehicle	**0.965**	0.963	0.962	0.952	0.963	-0.002	0.000	0.011	-0.001
yeast	0.000	**0.846**	0.831	0.833	0.815	0.846	0.016	0.014	0.031

Table 2: G-mean for different preprocessing methods and Logistic Regression, averaged over 10 runs of 5-fold cross-validation. The right side of the table shows the difference between the compared methods and ImWeights (positive difference means better result of ImWeights).

Table 2 presents the values of G-mean obtained by Logistic Regression with different preprocessing methods[1]: no preprocessing (Baseline), ImWeights (ImWt.), Random Oversampling (ROS), Borderline-SMOTE (B-SM.), and ADASYN (ADA.). Note that ImWeights is almost always better than the baseline (no data preprocessing), and when it is not the difference is probably not practically significant (below 1%). Concerning the competitive preprocessing methods, ImWeights performs better than Borderline-SMOTE and ROS, and is comparable with ADASYN. After ranking the results (computing average ranks as in the Friedman test), ImWeights gets the second lowest average rank equal to 2.45 outperforming Borderline-SMOTE (3.16) and Random Oversampling (2.95). ADASYN ranks slightly better (2.33), however, all the differences are not statistically significant. By taking into account only considerable differences (G-mean > 1%), we found that ImWeights surpasses

[1] More detailed results (including other classification measures) can be found at http://www.cs.put.poznan.pl/mlango/publications/imweights/

ADASYN on four datasets while ADASYN outperforms our approach three times. An analogous analysis performed between ImWeights and other methods in the experiment reveals that ImWeights has always more favorable differences than its counterparts. Especially for SMOTE, ImWeights substantially outpaces it on 8 datasets. It also seems that ImWeights achieves the best performance on datasets categorized as rare and partly borderline according to (Napierala and Stefanowski, 2012). For instance, on datasets such as seismic-bumps or glass it outperforms all the other methods, most notably on the first one where improvements are over 10%.

We also preformed experiments with several modifications of ImWeights, which use different scaling functions. Besides the scaling function defined by Eq. 2, we also tested a simple non-truncated linear function and a non-linear function given by $f(x) = (1 + \exp(-20x + 9))^{-1}$ which takes the form of a shifted sigmoid. ImWeights with a linear function achieved slightly worse results than its versions with the other functions. The non-linear function was comparable with the truncated linear function (Eq. 2), however the latter was finally selected due to its simplicity.

In terms of minority class recall (Table 3) ImWeights is considerably better than the baseline and Random Oversampling, but weaker than SMOTE and ADASYN. Taking into account the results on G-mean, one can hypothesize that using Borderline-SMOTE or ADASYN can lead to an overly extensive generalization of the minority class which causes substantial deterioration of majority class recognition.

	Recall					Difference with ImWeights			
Dataset	Baseline	ImWt.	ROS	B-SM.	ADA.	Baseline	ROS	B-SM.	ADA.
abalone	0.036	0.696	0.721	0.708	**0.733**	0.660	-0.025	-0.013	-0.038
balance-scale	0.000	0.184	0.276	**0.471**	0.361	0.184	-0.092	-0.288	-0.178
breast-w	0.937	0.954	0.950	**0.988**	0.970	0.017	0.004	-0.034	-0.016
ecoli	0.029	0.943	0.943	0.943	0.943	0.914	0.000	0.000	0.000
glass	0.000	**0.882**	0.606	0.629	0.688	0.882	0.276	0.253	0.194
haberman	0.160	0.519	0.522	0.472	**0.602**	0.358	-0.004	0.047	-0.084
new-thyroid	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
pima	0.541	0.754	0.732	**0.770**	0.753	0.213	0.022	-0.016	0.001
seismic-bumps	0.218	0.795	**0.914**	0.866	0.889	0.578	-0.119	-0.071	-0.094
transfusion	0.270	0.719	0.690	0.701	**0.737**	0.449	0.029	0.018	-0.018
vehicle	0.950	0.955	0.955	**0.968**	0.955	0.005	-0.001	-0.013	0.000
yeast	0.000	**0.843**	0.812	0.796	0.790	0.843	0.031	0.047	0.053

Table 3: Recall for different preprocessing methods and Logistic Regression, averaged over 10 runs of 5-fold cross-validation. The right side of the table shows the difference between the compared methods and ImWeights (positive difference means better result of ImWeights).

G-mean and Recall for Naive Bayes were comparable to those obtained by Logistic Regression (Tables 4 and 5).

It is also important to note that approaches which, similarly to ImWeights, improve classifier performance by assigning weights to examples have lower memory requirements than

	G-mean					Difference with ImWeights			
Dataset	Baseline	ImWt.	ROS	B-SM.	ADA.	Baseline	ROS	B-SM.	ADA.
abalone	0.590	0.557	**0.592**	0.394	0.348	-0.033	-0.035	0.163	0.210
balance-scale	0.000	0.278	0.342	**0.457**	0.394	0.278	-0.064	-0.180	-0.117
breast-w	0.962	0.965	0.964	0.964	**0.966**	0.003	0.000	0.000	-0.001
ecoli	**0.851**	0.830	0.833	0.798	0.847	-0.021	-0.003	0.032	-0.017
glass	0.594	0.589	0.588	0.601	**0.603**	-0.005	0.001	-0.011	-0.014
haberman	0.433	0.546	0.551	0.550	**0.568**	0.112	-0.005	-0.004	-0.023
new-thyroid	0.966	0.960	0.968	0.865	**0.969**	-0.006	-0.008	0.095	-0.009
pima	0.701	**0.740**	0.732	0.725	0.728	0.038	0.007	0.014	0.012
seismic-bumps	0.473	0.604	0.605	**0.606**	0.564	0.131	-0.001	-0.002	0.040
transfusion	0.454	0.574	0.541	**0.605**	0.571	0.120	0.033	-0.031	0.003
vehicle	0.719	0.710	0.729	**0.747**	0.708	-0.009	-0.020	-0.037	0.001
yeast	0.358	0.265	0.272	**0.574**	0.373	-0.093	-0.007	-0.309	-0.107

Table 4: G-mean for different preprocessing methods and Naive Bayes, averaged over 10 runs of 5-fold cross-validation. The right side of the table shows the difference between the compared methods and ImWeights (positive difference means better result of ImWeights).

	Recall					Difference with ImWeights			
Dataset	Baseline	ImWt.	ROS	B-SM.	ADA.	Baseline	ROS	B-SM.	ADA.
abalone	0.460	0.716	0.690	0.850	**0.966**	0.257	0.026	-0.134	-0.250
balance-scale	0.000	0.163	0.257	**0.378**	0.255	0.163	-0.094	-0.214	-0.092
breast-w	0.971	0.979	0.976	**0.992**	0.991	0.008	0.003	-0.013	-0.012
ecoli	0.943	0.943	0.943	0.943	0.943	0.000	0.000	0.000	0.000
glass	0.765	**0.824**	0.818	0.765	0.765	0.059	0.006	0.059	0.059
haberman	0.198	0.333	0.336	0.343	**0.365**	0.136	-0.002	-0.010	-0.032
new-thyroid	0.971	1.000	1.000	1.000	1.000	0.029	0.000	0.000	0.000
pima	0.586	0.698	0.687	**0.716**	0.695	0.112	0.011	-0.018	0.003
seismic-bumps	0.247	0.429	0.433	0.519	**0.587**	0.182	-0.004	-0.089	-0.158
transfusion	0.225	0.517	0.404	**0.587**	0.474	0.292	0.112	-0.070	0.043
vehicle	0.874	0.869	**0.929**	0.917	0.845	-0.005	-0.059	-0.047	0.024
yeast	0.961	0.961	0.961	0.925	0.945	0.000	0.000	0.035	0.016

Table 5: Recall for different preprocessing methods and Naive Bayes, averaged over 10 runs of 5-fold cross-validation. The right side of the table shows the difference between the compared methods and ImWeights (positive difference means better result of ImWeights).

standard oversampling methods. When the data imbalance is formidable, the dataset after oversampling can grow considerably due to new, artificially generated minority examples. For instance, storing the seismic-bumps dataset requires 242 kB of RAM memory in our implementation. The same dataset after oversampling requires almost twice this memory (454kB) while storing weights requires only 1% of memory overhead (244 kB). Therefore, the growing size of the oversampled dataset can become a serious issue, especially when processing massive data.

5. Conclusions

In this paper, we have attempted to provide a new perspective on combining local and global information about an imbalanced dataset. A critical discussion of existing data preprocessing methods has led us to the proposal of the ImWeights algorithm. The proposed approach weights examples according to the local safety of each data region and augments this weight by using a novel concept of gravity emitted by neighboring minority clusters. Experiments on real-world benchmark datasets have demonstrated that ImWeights achieves results comparable to several specialized preprocessing methods for imbalance data, outperforming them on unsafe datasets with dominating borderline or rare minority examples.

ImWeights can be still generalized in future research. For instance, our approach does not exploit all the data characteristics provided by the ImGrid clustering algorithm. The number of minority clusters, their size, or the distribution of majority clusters can be employed to better address different difficulties of data distribution and to, hopefully, improve classifier performance. There is also room for improvements in the data clustering phase. The current version of ImGrid uses a grid with all cells of the same size, constructed by dividing each dimension into sub-intervals of equal width. As the distribution of examples may vary for each attribute domain, the grid splitting procedure is susceptible to outliers and groups of values concentrated around distant example sub-centers. Finally, the topic of supporting qualitative attributes is also an open issue for future research.

Our view of exploiting both local and more global information of dataset difficulty can be generalized beyond weighting or sampling approaches for imbalanced data. The knowledge about the number of clusters and their difficulty can be incorporated into the classifier's construction procedure, adapting the learning process to the specific data distribution in particular regions. One can also think about an ensemble model which divides the feature space according to the clusters' positions and trains a base classifier for each of them. In such an approach, the learning procedure can benefit from the awareness of the clusters' characteristics by creating a meta-classifier that exploits component classifiers depending on the characteristics of the attribute region of a testing example.

Acknowledgments

We acknowledge the support from the Institute of Computing Science Statutory Funds.

References

P. Branco, L. Torgo, and R. Ribeiro. A survey of predictive modeling under imbalanced distributions. *ACM Comput Surv*, 49(2):31, 2016.

D. Brzezinski, J. Stefanowski, R. Susmaga, and I. Szczech. Visual-based analysis of classification measures and their properties for class imbalanced problems. *Information Sciences*, 462:242–261, 2018.

N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. SMOTE: Synthetic Minority Oversampling Technique. *J Artif Intell Res*, 16:341–378, 2002. ISSN 1076-9757.

C. Chen, A. Liaw, and L. Breiman. *Using random forest to learn imbalanced data.* Technical Report 666, 2004.

A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 2018.

V. Garcia, J.S. Sanchez, and R.A. Mollineda. 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 , LNCS*, volume 4756, pages 397–406. Springer, 2007.

H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887. Springer, 2005.

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

H. He and M. Yungian. *Imbalanced Learning. Foundations, Algorithms and Applications.* IEEE - Wiley, 2013.

H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks*, pages 1322–1328. IEEE, 2008.

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

N. Japkowicz and S. Stephen. Class imbalance problem: a systematic study. *Intelligent Data Analysis Journal*, 6(5):429–450, 2002.

H. Jeffreys. Some tests of significance, treated by the theory of probability. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 203–222. Cambridge University Press, 1935.

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

M. Koziarski and M. Woźniak. Ccr: A combined cleaning and resampling algorithm for imbalanced data classification. *International Journal of Applied Mathematics and Computer Science*, 27(4):727–736, 2017.

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

M. Lango, D. Brzezinski, S. Firlik, and J. Stefanowski. Discovering minority sub-clusters and local difficulty factors from imbalanced data. In Akihiro Yamamoto, Takuya Kida, Takeaki Uno, and Tetsuji Kuboyama, editors, *Discovery Science*, pages 324–339. Springer, 2017.

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

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

V. Lopez, A. Fernandez, S. Garcia, V. Palade, and F. Herrera. An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. *Inf Sci*, 257:113–141, 2014.

K. Napierala and J. Stefanowski. Identification of different types of minority class examples in imbalanced data. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 139–150. Springer, 2012.

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

K. Napierała, J. Stefanowski, and S. Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *Proc. of 7th Int. Conf. RSCTC 2010, LNAI*, volume 6086, pages 158–167. Springer, 2010.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Stefanowski. Dealing with data difficulty factors while learning from imbalanced data. In J. Mielniczuk and S. Matwin, editors, *Challenges in Computational Statistics and Data Mining*, pages 333–363. Springer, 2016.

Y. Sun, M. S. Kamel, A. Wong, and Y. Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(6):448–452, 1976.

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

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