Influence of minority class instance types on SMOTE imbalanced data oversampling

Przemysław Skryjomski SKRYJOMSKIPL@MYMAIL.VCU.EDU
Bartosz Krawczyk BKRAWCZYK@VCU.EDU
Department of Computer Science
Virginia Commonwealth University
Richmond, VA, USA

Editors: Luís Torgo, Bartosz Krawczyk, Paula Branco and Nuno Moniz.

Abstract

Despite more than two decades of intense research, learning from imbalanced data still remains as one of the major difficulties posed for computational intelligence systems. Among plethora of techniques dedicated to alleviating this problem, preprocessing algorithms are considered among the most efficient ones. They aim at re-balancing the training set by either undersampling of the majority class, or oversampling of the minority one. Here, Synthetic Minority Oversampling Technique, commonly known as SMOTE, stands as the most popular solution that introduces artificial instances on the basis of minority class neighborhood distribution. However, many recent works point out to the fact that the imbalanced ratio itself is not the sole source of learning difficulties in such scenarios. One should take a deeper look into the minority class structure in order to identify which instances influence the performance of classifiers in most significant manner. In this paper, we propose to investigate the role of minority class instance types on the performance of SMOTE. To achieve this, instead of oversampling uniformly the minority class, we preprocess only selected subsets of instances, based on their individual difficulties. Experimental study proves that such a selective oversampling leads to improved classification performance.

Keywords: Machine learning, Imbalanced data, Data preprocessing, Oversampling, SMOTE, Data complexity

1. Introduction

Most of existing classifiers work with an underlying assumption that classes given in the training set are roughly balanced. Their training performance is estimated using predictive accuracy or 0-1 loss function, both of which assume uniform importance of instances in the supplied dataset. This approach is however no longer valid when classes are imbalanced, i.e., instances from some class(es) are predominant. In such scenarios classifiers will be biased towards the majority class, while degrading their performance on minority cases. However, in many real-life scenarios such rare instances are of higher interest than abundant ones, e.g., in medicine or intrusion detection. Therefore, we must strive for obtaining a skew-insensitive classification systems that will offer high performance on minority classes, while not loosing significantly their predictive power on majority instances (Branco et al., 2016).

There are two main approaches for tackling the imbalance issue: data-level and algorithm-level (Krawczyk, 2016). The former ones focus on preprocessing solutions that aim at balancing the training set. The latter ones aim at identifying what causes a specific classifier to fail in imbalanced scenarios and improve those drawbacks to make new model skew-insensitive (Cano et al., 2013; Czarnecki and Tabor, 2017). Additionally, ensemble approaches have gained popularity, by allowing a hybrid between classifier combination and use of one of the mentioned techniques (Woźniak et al., 2014; Ksieniewicz et al., 2017).

Data-level solutions are very popular, as they are relatively easy to apply and are independent from the choice of a classifier. Among them Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) became a most frequently used algorithm. However, it is not free of several severe limitations. One of them is the uniform approach to oversampling that assumes equal importance of all minority instances. As recent studies show (Krawczyk, 2016), instances within the minority class may pose varying learning difficulties to classifiers. Thus, using more selective approach for imbalanced data preprocessing should lead to improved efficacy of the constructed system. There exist some algorithms that followed this line of thought. Borderline-SMOTE (Han et al., 2005) concentrates on oversampling instances that lie close to the class boundaries. This idea is further developed in ADASYN (He et al., 2008) that dynamically determine which instances may pose higher challenge for a classifier. Safe-Level-SMOTE (Bunkhumpornpat et al., 2009) assigns weights to instances depending on how 'safe' they are from being affected by majority class and use these weights to guide the introduction of artificial examples. SPIDER (Napierala et al., 2010) specializes at emphasizing difficult instances, especially those overlapping with the majority class.

In this paper, we propose to empower SMOTE by analyzing the structure of minority class and identifying individual difficulties posed by its instances. Then, we conduct selective SMOTE on subset of instances in order to enhance the presence of most challenging instances. The main contributions of this paper include:

- Incorporating the background information on learning difficulties embedded in the minority class into SMOTE-based oversampling.
- A minority class instance selection based on its individual properties that allows for oversampling only specific instances and offers guided alleviation of the bias towards the majority class.
- An insight into the role of different types of minority class instances in imbalanced data preprocessing.

The proposed algorithm is examined by conducting an experimental study on a set of diverse benchmarks. Obtained results clearly show that embedding knowledge about the minority class in SMOTE may lead to an improved performance in comparison to the canonical version of this algorithm.

2. Imbalanced data oversampling with SMOTE

In 2002 an intelligent approach to oversampling was introduced. As opposed to randomized solutions that oversampled by duplicating existing instances of minority class, it allowed to create new artificial instances class with use of knowledge about neighbours that surrounds each sample in minority class. This method is called *Synthetic Minority Over-Sampling Technique* (Chawla et al., 2002) (*SMOTE*), pseudocode for which is shown in Alg. 1.

Algorithm 1: Synthetic Minority Over-Sampling Technique aka "SMOTE"

```
Function SMOTE ($D_{minority}$, $N_{percent}$, $k$)
    $D_{smoted} \longleftarrow [\,]$
    for $i \leftarrow 1$ to $nrow(D_{minority})$ do
        $nn \longleftarrow kNN(D_i,\, D_{minority},\, k)$
        $N_i \longleftarrow \lfloor N_{percent}/100 \rfloor$
        while $N_i$ != 0 do
            $neighbour \longleftarrow select\_random(nn)$
            $gap \longleftarrow range\_random(0,\, 1)$
            $diff \longleftarrow neighbour$ - $D_i$
            $synth \longleftarrow D_i + gap * diff$
            $D_{smoted} \longleftarrow append(D_{smoted},\, synth)$
            $N_i \longleftarrow N_i$ - 1
        end
    end
    return $\longleftarrow D_{smoted}$
```

This preprocessing method works in conjunction with *k-Nearest Neighbour* (k-NN) method that allows to find nearest neighbours for a given object from the neighbourhood provided as an input. Distance can be computed with the use of different metrics, although for most cases Euclidean distance or its normalized equivalent is being used. As a result, static number of k neighbours is returned for each given object.

Artificially created instances are relying on the computed neighbourhood from which one neighbour is selected randomly for new object. Amount of synthetic instances formed per original instance is controlled by the $N_{percent}$ argument provided as an input to the algorithm. Each new instance is created by adding to the original object computed difference between randomly selected neighbour and the source instance, which is additionally multiplied by randomly chosen value in $0 - 1$ range. This allows to control the final position of the artificial instance, which can be located in the same position of original one, selected neighbour or in between them depending on the randomly generated value. This increases the diversity of artificial instance set, allowing for better exploitation of the given decision space.

It is worth mentioning that SMOTE is characterized by a significant computational complexity and memory requirements, which become visible when handling large-scale imbalanced data (Krawczyk, 2016). Therefore, some attention has been paid to developing high-performance implementations of this algorithm to make it suitable for ever-increasing volumes of data. Recent work shows the GPU-based implementation, allowing for efficient preprocessing of millions of instances (Gutiérrez et al., 2017).

3. Identifying types of minority class instances

When dealing with imbalanced data class disproportion is not the main source of learning difficulty, as local characteristics of minority class are also important. Objects from minority class often forms heterogeneous structures within the whole dataset and creates *characteristic regions* that can be defined as a types of minority class instances. Examples of them are presented on Fig. 1.

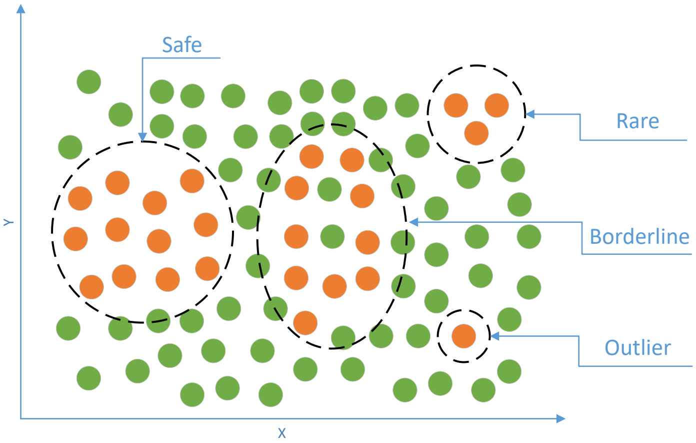

Figure 1: Types of minority class instances

As seen above, it is possible to specify several types of minority class instances by relying on their neighbourhood and position relative to the objects forming homogenous space known as *safe region* (Napierala and Stefanowski, 2016). Safe objects are clearly separated from other class instances and often represent largest fraction of the minority class instances. All objects that formed *safe region* are defined as a *safe* type. Instances outside this *safe* region can be characterized with use of three remaining types. Objects which are located closely to the border between two or more classes, often having some overlapping regions are identified as a *borderline* type. The next region type consists of *rare* samples which form small structures, although distant to the formed safe regions and expressed in a low number of minority class instances within. Last region consists of individual objects that are known as *outliers*. They may stand for noise or otherwise provide unique information to the classification problem.

In order to determine to which type each minority instance belongs, specific method for instance difficulty analysis needs to be applied. In this article we will cover algorithm that allows to classify and label group membership of minority objects (Napierala and Stefanowski, 2016). This method depends on neighbourhood computation with a use of *k-Nearest Neighbours* algorithm. Types of minority class instances are determined by analyzing neighbours that were found with specified limiting preset and their corresponding class membership. Pseudocode of this algorithm can be seen in Algorithm 2.

Proposed solution extracts minority class instances from the original dataset, then computes neighbourhood for each minority sample, while taking majority class into account. Specified number of neighbours is found with use of *k-NN* and its built-in *Heterogenous Value Difference Metric* (Napierala and Stefanowski, 2016). Processing this algorithm on given dataset results in a vector in which each entry contains identified types for the *subsequent* minority sample that was found in the original dataset. Minimum number of minority class instances that needs to exist in order to classify sample to the specific region is computed by multiplying neighbourhood size by static, predefined values. In this paper we use 0.8, 0.5 and 0.2 for Safe, Borderline and Rare regions respectively. This approach allows to use different k values. However, $k = 5$ is most widely selected size of neighbourhood (Napierala and Stefanowski, 2016).

As mentioned before, this algorithm makes use of Heterogenous Value Difference Metric. While Euclidean distance metric allows to calculate distance between two objects in n dimensional space, using this metric introduces some issues regarding imbalanced data and performance problems regarding computation or approximation of the square root (Wilson and Martinez, 1996).

Algorithm 2: Neighbourhood analysis for determining types of minority class instances

```
Function Types (D, k)
    $types \longleftarrow [\,]$
    $D_{minority} \longleftarrow get\_minority(D)$
    $D_{majority} \longleftarrow get\_majority(D)$
    foreach $x_i$ in $D_{minority}$ do
        $neighbours \longleftarrow kNN\_HVDM(x_i,\ D,\ k)$
        $N_{minority} \longleftarrow minority\_samples(neighbours)$
        if $N_{minority} \geqslant \lfloor 0.8k \rfloor$ then
            $types_i \longleftarrow$ "safe"
        end
        else if $N_{minority} \geqslant \lfloor 0.5k \rfloor$ then
            $types_i \longleftarrow$ "borderline"
        end
        else if $N_{minority} \geqslant \lfloor 0.2k \rfloor$ then
            $types_i \longleftarrow$ "rare"
        end
        else
            $types_i \longleftarrow$ "outlier"
        end
    end
    return $\longleftarrow types$
```

Many datasets contains both nominal and numerical features. One major weakness of the Euclidean distance is that when some features have large range of values as opposed to remaining attributes, they may introduce bigger impact on the computed distance, while attributes with lower range of values will have lesser impact on the results. As a solution to this problem, normalization is done on the Euclidean distance metric. Although this allows to overcome major problems, there is another issue with inability of appropriate handling of nominal features (Wilson and Martinez, 1996). *Heterogenous Value Difference Metric* (*HVDM*) allows handling both normalized nominal and numerical values, as shown in Eq. 1.

$$d_a(x,y) = \begin{cases} 1 & \text{if } x \text{ or } y \text{ is unknown} \\ nvdm_a(x,y) & \text{if } a \text{ is nominal} \\ ndiff_a(x,y) & \text{if } a \text{ is linear} \end{cases} \tag{1a}$$

$$nvdm_a(x,y) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^2} \tag{1b}$$

$$ndiff_a(x,y) = \frac{|x-y|}{4\sigma_a} \tag{1c}$$

$$D(x,y) = \sqrt{\sum_{a=1}^{n} d_a(x_a,y_a)^2} \tag{1d}$$

This metric can be used as any other distance metric, but the major difference is that the algorithm needs to determine which features are nominal and which are linear. There are two different functions for computing distance, $nvdm_a$ and $ndiff_a$ for nominal and linear features respectively. As for nominal attributes, distance is computed by basing on number of instances that exists for specific case, for example $N_{a,x}$ equals number of instances that has value x for attribute a and $N_{a,x,c}$ when specific class c is only taken into account. Applying this distance metric allows to identify minority samples as accurately as possible. After determining specific types of minority class instances, preprocessing can be applied to specific instance groups.

Result of minority instance type analysis algorithm on selected datasets can be seen on Fig. 2 where *Multidimensional Scaling* was done in order to present results of the algorithm on a 2D plot.

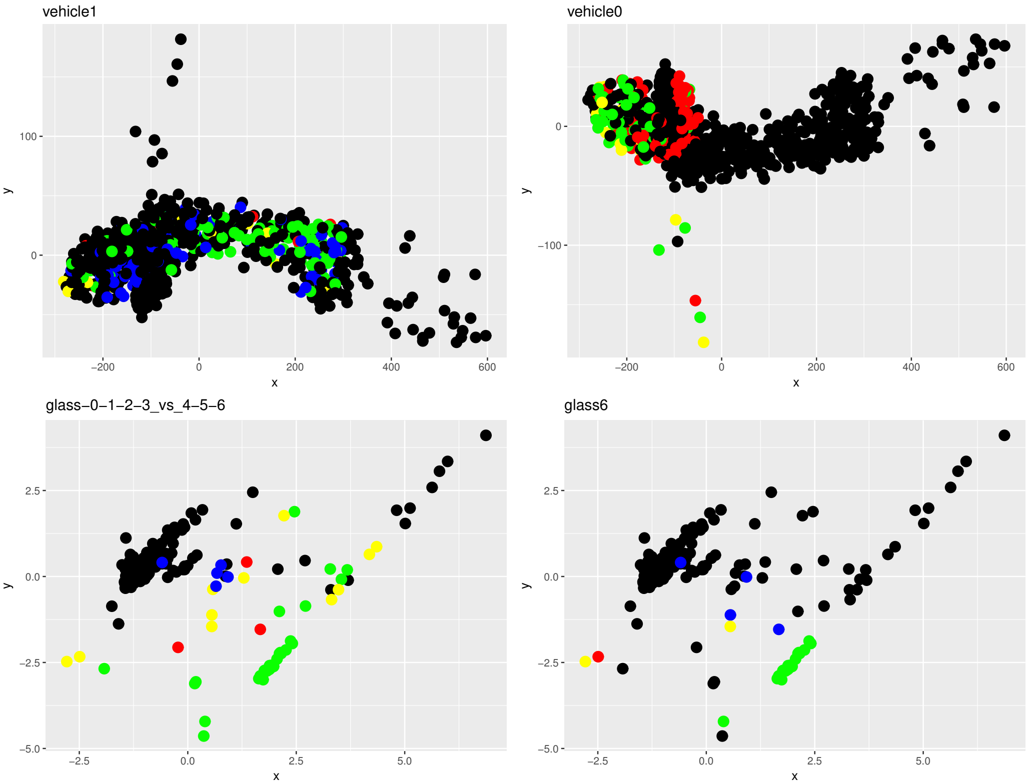

Figure 2: Multidimensional Scaling of the neighbourhood analysis results

Each color represents proper minority type which is shown in Table 1 with exception of black color which corresponds to the majority class instances shown in the plot of *MDS*. It is worth to pay attention to the borderline objects located near and inside safe region. The main reason for this behavior is that this is a low-dimensional projection of the multi-dimensional dataset, therefore some objects may be deceiving as opposed to their original placement.

Table 1: Determining types of dataset objects

Color	Type	Ratio
🟩	Safe	$M \geqslant \lfloor 0.8k \rfloor$
🟨	Borderline	$M \geqslant \lfloor 0.5k \rfloor$
🟥	Rare	$M \geqslant \lfloor 0.2k \rfloor$
🟦	Outlier	$M \geqslant 0$
⬛	Majority class	*Not applicable*

Ratio given in Table 1 allows adapting to different k values supplied to the algorithm. For the most common $k = 5$, there need to be 4 *or more* minority samples in neighbourhood in order to identify them as a *safe* instances, then for *borderline* type neighbourhood needs to contain 2 *or* 3 minority samples and for *rare* type there need to be 1 minority instance in the nearest location. Minority samples are identified as *outliers* where there are no other minority samples in the neighbourhood.

4. Combining SMOTE with specific instance types

After covering algorithm that allows to analyze neighbourhood and determine types of minority class instances, it is possible to perform experiment in which selected base classifier will be tested in different preprocessing configurations that depends on oversampling only specific and selected types of minority class instances. Proposed experiment relies on classification performance assessment on several combinations which are formed by concatenating specific, oversampled minority class types. Each possible combination is shown in Table 2.

Table 2: Combinations of minority types in oversampling

Safe	Borderline	Rare	Outlier
✓	-	-	-
-	✓	-	-
-	-	✓	-
-	-	-	✓
✓	✓	-	-
✓	-	✓	-
✓	-	-	✓
-	✓	✓	-
-	✓	-	✓
-	-	✓	✓
✓	✓	✓	-
✓	✓	-	✓
✓	-	✓	✓
-	✓	✓	✓
✓	✓	✓	✓

This experiment allows to determine if by oversampling specific cases through artificial generation of samples can lead towards classification performance improvement. Results of this experiment are shown in Section 5.3 with direct comparison to the application of SMOTE on the whole training set.

5. Experimental study

For experimentation purposes, a number of diverse benchmark datasets were selected from the public KEEL Imbalanced Data repository. Two-class problem related datasets already prepared for 5-Fold Cross Validation were selected with specific Imbalanced Ratio (IR) with mind and covered in Section 5.1. Evaluation methodology, used algorithms implementations and their configuration is shown in Section 5.2. Results of the experiment including analysis of obtained performance can be seen in Section 5.3.

5.1. Datasets

Datasets that were used in evaluation of proposed solutions are shown in Table 3 and sorted by the value of Imbalance Ratio. Datasets are classified to one of the possible Imbalance Ratio tiers depending on the degree of imbalance. Each dataset is described by the Imbalance Ratio, number of features, amount of instances including majority and minority class instances contained in the table separately. Additionally information about percentage amount of safe, borderline, rare and outlier examples on unsplit imbalanced dataset is supplied. Volume of the dataset is considered high if it contains at least 1000 instances with respect to the size of the datasets used in this study. Tier levels and datasets belonging to them are determined by the Imbalance Ratio, where the higher Tier the more imbalanced datasets are in particular group. Columns named ”S”, ”B”, ”R” and ”O” corresponds to the percentage amount of Safe, Borderline, Rare and Outlier instances respectively.

Table 3: Selected datasets for evaluation

Dataset	Tier	Vol.	IR	Feat.	Inst.	Maj.	Min.	S [%]	B [%]	R [%]	O [%]
glass1	1	Low	1.82	10	214	138	76	2.63	11.84	6.58	78.95
wisconsin	1	Low	1.86	10	683	444	239	61.09	18.83	5.02	15.06
pima	1	Low	1.87	9	768	500	268	32.09	41.04	13.81	13.06
haberman	1	Low	2.78	4	306	225	81	4.94	4.94	2.47	87.65
vehicle2	2	Low	2.88	19	846	628	218	34.86	9.63	55.05	0.46
vehicle1	2	Low	2.90	19	846	629	217	41.01	7.37	3.23	48.39
vehicle3	2	Low	2.99	19	846	634	212	46.70	8.02	3.30	41.98
glass-0-1-2-3_vs_4-5-6	2	Low	3.20	10	214	163	51	62.75	21.57	5.88	9.80
vehicle0	3	Low	3.25	19	846	647	199	29.65	10.05	59.80	0.50
new-thyroid1	3	Low	5.14	6	215	180	35	37.14	34.29	20.00	8.57
segment0	3	High	6.02	20	2308	1979	329	55.32	31.31	8.81	4.56
glass6	3	Low	6.38	10	214	185	29	75.86	6.90	3.45	13.79
vowel0	4	Low	9.98	14	988	898	90	26.67	41.11	8.89	23.33
cleveland-0_vs_4	4	Low	12.31	14	173	160	13	61.54	0.00	15.38	23.08
abalone9-18	4	Low	16.40	9	731	689	42	2.38	2.38	92.86	2.38
glass5	4	Low	22.78	10	214	205	9	0.00	33.33	22.22	44.44
lymphography-normal-fibrosis	5	Low	23.67	19	148	142	6	0.00	50.00	16.67	33.33
winequality-red-4	5	High	29.17	12	1599	1546	53	0.00	1.89	13.21	84.91
winequality-white-3_vs_7	5	Low	44.00	12	900	880	20	0.00	10.00	20.00	70.00
kddcup-buffer_overflow_vs_back	5	High	73.43	42	2233	2203	30	96.67	0.00	3.33	0.00

5.2. Set-up

In order to fairly assess performance of proposed solution, multiple independent validations are required. In our experiment *5-Fold Cross Validation* on selected datasets was done on original KEEL datasets. As a base classifier we have selected C5.0 decision tree, due to its highly efficient implementation that exists in selected testing environment and due to the fact that decision trees are popular choice for dealing with imbalanced data (Pandya and Pandya, 2015).

This paper focuses on binary imbalanced classification tasks. The basic metrics for evaluation are true positive TP, true negative TN, false positive FP and false negative FN which can be deducted from confusion matrix. However, one needs aggregated measures to compare classifiers. For experimental study we will use the following ones.

Balanced Accuracy shown in Eqn. 2 is a metric that was used for performance evaluation and can be described as an average accuracy received from both minority and majority class (Brodersen et al., 2010).

$$BAccuracy = \frac{1}{2}\left(\frac{TP}{TP+FP} + \frac{TN}{TN+FN}\right) \quad (2)$$

Information about proper classification of minority class can be obtained by "Sensitivity" metric also known as "Recall", shown in Eqn. 3.

$$Senstivity = \frac{TP}{TP+FN} \quad (3)$$

As the above metric takes only one class into consideration, Geometric Mean shown in Eqn. 4 is used as it balances between classification accuracy over the instances from both minority and majority classes at the same time.

$$GM = \sqrt{TP_{rate} * TN_{rate}} \quad (4)$$

F-Measure shown in Eqn. 5 can be considered as a harmonic mean of both precision and sensitivity which can measure accuracy of the test.

$$FMeasure = \frac{2*TP}{2*TP+FP+FN} \quad (5)$$

Another metric that is connected with classification of imbalanced data is the Area Under the ROC Curve (AUC) shown in Eqn. 6. ROC allows to visualize tradeoff between benefits (TP_{rate}) and costs (FP_{rate}) regarding minority class and classification task. Computed AUC value allows to measure performance of the classifier and corresponds to the probability that the proposed solution will favor minority class instance instead of majority.

$$AUC = \frac{1}{2}\left(1 + TP_{rate} - FP_{rate}\right) \quad (6)$$

Experiment was done in *R programming language* with use of *C50*, *Rcpp* and *MASS* libraries. SMOTE method, neighbourhood analysis, *k-NN* algorithm and HVDM metric covered in Section 3 are custom implementations which will be released in upcoming *R* library. Base classifier was run with default configuration ie. *information gain* was used, no tree pruning. For neighbourhood analysis $k = 5$ neighbours was used and $k = 3$ for SMOTE preprocessing respectively. There are other approaches that allows to determine which value would be optimal for the given dataset, but for most datasets fixed amount of neighbours set like seen above proved to fulfill the task and this values are commonly used in other articles. Essential, core part of code used for the processed experiment can be seen at:
http://redtux.rocik.net/public/lidta2017.zip

5.3. Results and discussion

Results from the preprocessing methods are shown in Tables 4 – 8, containing obtained classification performance for the datasets in corresponding Tier with concatenated minority instance types named as "S", "B", "R", "O" for Safe, Borderline, Rare, Outlier respectively and "-" for excluding specific region. As baseline we present results for original SMOTE. Additionally, results of the Friedman rank test with Shaffer Post-Hoc test(Garcia and Herrera, 2008) are shown in Figure 3 in which instance typecombinations are named by the first letters of the used minority class types and "All" refers to the original SMOTE method done uniformly on minority class. Results plotted with *red* color are considered as lack of statistically significant differences between two methods. As can be seen in Table 4, for datasets with low IR like *glass1* and *haberman* original SMOTE oversampling on all minority class instances gives the best performance. Although in *wisconsin* and *pima* datasets taking types into account gives some benefits as it allows to boost classification accuracy of both classes including minority class sensitivity while maintaining high overall accuracy. Classification performance of more imbalanced datasets from Tier 2 and 3 shown in Tables 5 – 6 is greatly improved by oversampling done on small specific regions of minority class and their combinations. Taking local characteristic into consideration allows to increase senstivitity along with the overall accuracy making oversampling of certain combination the best solution for given datasets with higher IR as shown for *vehicle2* and *new-thyroid1*. Even more imbalanced datasets, shown in Table 7 from Tier 4 benefits from a more sophisticated process of forming synthetic samples. Not only several combinations allows to improve sensitivty rate to 100% and overall classification rate to higher levels as opposed to original SMOTE as shown in *vowel0*, they allows to at least keep performance of base solution while improving specific validation criteria as shown in *glass5*. Although oversampling of certain types may slightly harm accuracy in some cases as shown in results from the *abalone9-18* dataset.

Table 4: Preprocessing results for Tier 1 datasets

	glass1					wisconsin				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	**94.93**	**96.39**	**95.83**	**94.07**	**95.84**	97.18	98.85	98.01	96.88	98.02
S / - / - / -	89.59	79.27	86.61	83.88	87.18	97.06	98.12	97.70	96.61	97.71
- / B / - / -	92.26	87.82	91.04	88.88	91.28	97.32	98.54	98.00	96.97	98.00
- / - / R / -	92.21	83.20	89.59	87.77	89.88	**97.63**	98.22	98.07	**97.21**	98.07
- / - / - / O	93.36	93.08	93.80	91.73	93.82	97.19	98.74	97.99	96.87	97.99
S / B / - / -	92.91	87.16	91.37	89.51	91.59	96.68	98.12	97.48	96.21	97.48
S / - / R / -	91.40	83.84	89.33	87.09	89.65	96.86	98.12	97.59	96.40	97.59
S / - / - / O	93.16	92.75	93.53	91.41	93.57	96.97	98.64	97.82	96.62	97.83
- / B / R / -	91.65	91.77	92.20	89.59	92.26	96.95	97.59	97.47	96.39	97.47
- / B / - / O	92.87	94.08	93.86	91.51	93.87	97.35	98.95	**98.15**	97.08	**98.15**
- / - / R / O	91.83	93.09	92.78	90.11	92.83	97.11	98.95	98.01	96.83	98.01
S / B / R / -	91.93	90.79	92.07	89.66	92.13	96.73	98.12	97.51	96.26	97.51
S / B / - / O	92.46	94.08	93.59	91.08	93.60	96.97	98.64	97.82	96.62	97.83
S / - / R / O	93.11	94.73	94.07	91.78	94.10	96.97	98.64	97.82	96.62	97.83
- / B / R / O	92.79	95.40	94.15	91.65	94.16	97.06	**99.27**	98.08	96.84	98.09
S / B / R / O	92.79	95.73	94.21	91.68	94.24	96.92	98.64	97.80	96.57	97.80
	pima					haberman				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	84.50	93.00	87.21	82.01	87.40	**72.86**	**85.51**	**77.80**	**64.69**	**78.37**
S / - / - / -	**86.58**	87.50	87.80	83.57	87.85	71.65	38.92	58.86	46.69	64.74
- / B / - / -	83.75	86.48	85.48	80.36	85.56	72.62	37.38	58.14	46.29	64.52
- / - / R / -	86.43	88.53	88.04	83.67	88.06	57.16	35.23	49.74	40.12	63.28
- / - / - / O	82.76	87.22	84.87	79.48	84.94	70.11	75.86	73.45	59.93	74.09
S / B / - / -	84.56	88.72	86.60	81.66	86.66	72.79	38.30	58.87	47.12	65.09
S / - / R / -	86.30	90.11	**88.30**	**83.80**	**88.33**	70.54	39.83	59.77	47.56	65.08
S / - / - / O	84.79	91.14	87.01	82.11	87.17	69.14	76.16	72.88	58.86	73.47
- / B / R / -	85.57	91.05	87.78	83.04	87.87	72.61	41.10	60.79	49.20	66.16
- / B / - / O	84.03	87.78	86.11	81.02	86.14	70.21	76.78	73.88	60.24	74.50
- / - / R / O	83.80	86.95	85.66	80.54	85.72	69.97	77.09	73.60	59.89	74.27
S / B / R / -	85.35	90.30	87.49	82.69	87.58	71.83	42.33	61.32	49.44	66.27
S / B / - / O	84.97	91.51	87.39	82.35	87.53	70.26	76.47	74.25	60.43	74.74
S / - / R / O	84.31	92.91	86.35	81.38	86.75	70.50	78.32	74.55	60.79	75.16
- / B / R / O	84.04	94.12	86.87	81.48	87.16	70.60	77.09	74.33	60.78	74.93
S / B / R / O	83.87	**94.40**	86.50	81.18	86.88	71.29	77.09	75.34	61.80	75.82

Table 5: Preprocessing results for Tier 2 datasets

	vehicle2					vehicle1				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	98.59	98.40	98.80	98.07	98.80	91.36	94.36	94.16	89.20	94.18
S / - / - / -	**98.96**	97.36	98.47	98.09	98.48	91.11	91.81	93.05	88.24	93.08
- / B / - / -	98.86	**98.97**	**99.14**	**98.52**	**99.15**	92.46	85.59	90.83	87.49	91.10
- / - / R / -	98.82	97.82	98.63	98.10	98.63	91.35	85.96	90.46	86.28	90.71
- / - / - / O	98.90	98.63	99.01	98.45	99.01	91.74	88.48	91.82	87.78	91.94
S / B / - / -	98.30	98.28	98.64	97.71	98.64	90.00	92.16	92.33	86.75	92.43
S / - / R / -	98.66	98.17	98.72	98.06	98.72	91.82	92.97	93.88	89.40	93.88
S / - / - / O	98.62	97.94	98.61	97.93	98.61	91.55	93.44	93.88	89.15	93.90
- / B / R / -	98.39	98.51	98.78	97.89	98.78	90.07	90.91	91.87	86.51	91.96
- / B / - / O	98.46	98.05	98.60	97.82	98.61	89.98	91.70	92.44	86.93	92.47
- / - / R / O	98.29	98.28	98.64	97.72	98.64	91.13	91.36	92.73	87.97	92.80
S / B / R / -	97.71	98.62	98.57	97.23	98.58	90.98	91.93	93.03	88.12	93.06
S / B / - / O	98.21	97.82	98.41	97.48	98.41	90.79	93.78	93.70	88.46	93.71
S / - / R / O	98.29	98.28	98.64	97.72	98.64	**92.87**	94.70	**94.96**	**90.93**	**94.97**
- / B / R / O	98.06	98.51	98.66	97.56	98.66	91.43	90.09	92.48	88.09	92.52
S / B / R / O	97.57	98.74	98.57	97.12	98.57	91.20	**94.82**	94.27	89.14	94.29
	vehicle3					glass-0-1-2-3_vs_4-5-6				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	88.67	93.87	92.82	86.02	92.84	97.52	97.55	98.07	96.61	98.08
S / - / - / -	89.35	92.33	92.50	86.40	92.52	98.23	93.60	96.51	95.94	96.57
- / B / - / -	**93.52**	90.22	93.32	**90.06**	93.47	97.20	96.09	97.34	95.84	97.35
- / - / R / -	91.06	80.90	88.08	84.04	88.65	97.12	95.57	97.08	95.59	97.10
- / - / - / O	87.12	90.08	90.56	83.30	90.61	97.44	96.09	97.41	96.08	97.43
S / B / - / -	90.06	91.39	92.29	86.71	92.34	96.95	94.60	96.59	95.07	96.61
S / - / R / -	90.65	92.57	93.12	87.82	93.15	**98.31**	94.09	96.76	96.19	96.81
S / - / - / O	90.05	93.99	93.31	87.42	93.35	98.26	98.04	98.56	**97.57**	98.56
- / B / R / -	93.42	89.87	92.98	89.58	93.18	95.84	96.06	96.87	94.45	96.88
- / B / - / O	91.57	93.04	93.77	88.98	93.80	97.49	99.02	**98.74**	97.11	**98.75**
- / - / R / O	89.01	89.14	91.12	85.19	91.18	97.62	97.06	97.91	96.60	97.92
S / B / R / -	90.26	89.86	91.88	86.64	91.95	96.88	95.57	97.01	95.34	97.02
S / B / - / O	89.81	**95.76**	**94.03**	87.74	**94.05**	97.10	98.02	98.17	96.39	98.17
S / - / R / O	89.11	94.10	93.07	86.53	93.09	97.76	97.55	98.15	96.86	98.16
- / B / R / O	91.49	92.58	93.60	88.82	93.63	96.55	**99.51**	98.60	96.23	98.61
S / B / R / O	89.31	94.81	93.48	86.99	93.50	96.89	99.02	98.51	96.44	98.52

Table 6: Preprocessing results for Tier 3 datasets

	vehicle0					new-thyroid1				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	97.34	98.37	98.41	96.74	98.41	99.09	97.86	98.78	98.21	98.79
S / - / - / -	98.64	97.99	98.66	97.93	98.67	98.76	97.86	98.71	97.87	98.72
- / B / - / -	98.13	96.74	97.93	96.98	97.94	99.09	97.86	98.78	98.21	98.79
- / - / R / -	98.76	96.86	98.19	97.65	98.20	99.31	96.43	98.12	97.82	98.14
- / - / - / O	**98.88**	97.61	98.57	98.04	98.57	98.09	97.86	98.57	97.18	98.58
S / B / - / -	98.35	**99.12**	99.08	98.02	99.08	99.45	97.86	98.85	98.56	98.86
S / - / R / -	98.25	97.86	98.48	97.50	98.49	99.38	97.14	98.49	98.19	98.50
S / - / - / O	98.87	98.24	98.85	98.25	98.85	98.33	97.14	98.28	97.15	98.29
- / B / R / -	98.44	97.24	98.26	97.48	98.27	99.59	95.71	97.83	97.80	97.86
- / B / - / O	98.39	97.24	98.24	97.42	98.25	99.02	97.14	98.42	97.84	98.43
- / - / R / O	98.82	97.24	98.38	97.85	98.39	99.24	95.71	97.75	97.44	97.79
S / B / R / -	97.32	98.62	98.52	96.80	98.52	99.79	97.86	98.92	98.91	98.93
S / B / - / O	98.62	98.99	**99.11**	**98.26**	**99.11**	99.52	**98.57**	99.21	98.92	99.22
S / - / R / O	98.25	97.86	98.48	97.50	98.49	99.38	97.14	98.48	98.18	98.50
- / B / R / O	98.73	97.11	98.30	97.72	98.31	99.38	97.14	98.48	98.18	98.50
S / B / R / O	97.70	98.37	98.53	97.10	98.53	**99.86**	**98.57**	**99.28**	**99.27**	**99.29**
	segment0					glass6				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	99.69	99.47	99.69	99.47	99.69	**99.27**	95.65	97.71	97.33	97.76
S / - / - / -	99.66	98.25	99.09	98.93	99.09	98.62	87.93	93.68	93.12	93.90
- / B / - / -	99.79	98.86	99.41	99.31	99.41	98.33	89.67	94.53	93.65	94.70
- / - / R / -	99.60	98.86	99.38	99.12	99.38	98.46	91.41	95.45	94.62	95.57
- / - / - / O	**99.85**	99.09	99.53	99.47	99.53	97.33	97.43	98.30	96.20	98.31
S / B / - / -	99.37	98.86	99.34	98.90	99.34	98.68	88.80	94.12	93.57	94.33
S / - / R / -	99.71	99.24	99.58	99.39	99.58	98.36	90.54	94.98	94.11	95.14
S / - / - / O	99.65	99.32	99.61	99.36	99.61	97.33	97.43	98.30	96.20	98.31
- / B / R / -	99.77	99.47	99.70	99.54	99.70	98.46	91.41	95.45	94.62	95.57
- / B / - / O	99.82	99.24	99.60	99.50	99.60	97.33	97.43	98.30	96.20	98.31
- / - / R / O	99.80	99.39	99.67	99.54	99.67	98.23	98.30	98.87	**97.48**	98.88
S / B / R / -	99.62	99.47	99.68	99.39	99.68	98.82	90.54	95.04	94.54	95.20
S / B / - / O	99.67	99.62	99.76	99.51	99.76	97.33	97.43	98.30	96.20	98.31
S / - / R / O	99.72	99.77	99.84	**99.62**	99.84	97.14	**99.17**	**99.11**	96.70	**99.11**
- / B / R / O	99.77	99.54	99.74	99.58	99.74	98.23	98.30	98.87	**97.48**	98.88
S / B / R / O	99.57	**99.85**	**99.85**	99.51	**99.85**	98.23	98.30	98.87	**97.48**	98.88

Table 7: Preprocessing results for Tier 4 datasets

	vowel0					cleveland-0_vs_4				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	99.42	99.44	99.67	99.17	99.67	96.28	82.55	90.61	87.72	91.04
S / - / - / -	99.36	98.33	99.11	98.61	99.11	94.26	90.36	94.50	89.38	94.71
- / B / - / -	99.23	98.33	99.09	98.46	99.10	94.08	80.55	89.06	84.23	89.88
- / - / R / -	99.68	99.17	99.55	99.30	99.56	94.93	74.55	85.54	80.86	86.96
- / - / - / O	**99.86**	**100.00**	**99.99**	**99.86**	**99.99**	94.92	78.36	87.45	82.84	88.87
S / B / - / -	99.15	99.44	99.64	98.89	99.64	**96.68**	88.36	93.64	**90.80**	93.95
S / - / R / -	99.65	98.61	99.27	99.01	99.28	94.87	90.36	94.58	90.18	94.79
S / - / - / O	**99.86**	**100.00**	**99.99**	**99.86**	**99.99**	92.30	**92.55**	**95.45**	88.26	**95.57**
- / B / R / -	98.89	99.44	99.61	98.63	99.61	95.07	78.55	88.13	84.00	88.96
- / B / - / O	99.85	99.72	99.85	99.72	99.85	95.14	84.36	91.19	87.29	91.87
- / - / R / O	**99.86**	**100.00**	**99.99**	**99.86**	**99.99**	94.28	78.36	87.58	82.34	88.79
S / B / R / -	99.44	99.72	99.81	99.31	99.81	95.79	88.36	93.57	89.96	93.87
S / B / - / O	**99.86**	**100.00**	**99.99**	**99.86**	**99.99**	94.56	88.55	93.57	88.63	93.80
S / - / R / O	**99.86**	**100.00**	**99.99**	**99.86**	**99.99**	92.07	84.55	91.33	84.67	91.65
- / B / R / O	99.85	99.72	99.85	99.72	99.85	94.43	82.36	90.17	85.48	90.79
S / B / R / O	99.73	**100.00**	99.97	99.72	99.97	93.17	82.55	90.32	84.59	90.72
	abalone9-18					glass5				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	91.98	**89.16**	**93.91**	**86.77**	**94.09**	95.14	**100.00**	**99.76**	**94.82**	**99.76**
S / - / - / -	95.04	34.01	55.95	46.76	66.90	–	–	–	–	–
- / B / - / -	92.65	46.04	67.21	59.57	72.82	**98.71**	91.43	95.49	94.21	95.65
- / - / R / -	91.95	68.95	82.25	75.18	84.09	95.14	**100.00**	**99.76**	**94.82**	**99.76**
- / - / - / O	92.18	53.10	71.76	64.33	76.30	97.52	94.29	96.91	94.41	97.02
S / B / - / -	94.42	41.91	64.24	56.99	70.84	–	–	–	–	–
S / - / R / -	90.95	69.55	82.43	74.63	84.34	–	–	–	–	–
S / - / - / O	91.00	50.14	69.02	60.69	74.80	–	–	–	–	–
- / B / R / -	90.70	71.96	83.90	75.61	85.49	95.14	**100.00**	**99.76**	**94.82**	**99.76**
- / B / - / O	91.47	47.11	66.74	58.02	73.30	96.33	97.14	98.33	94.62	98.39
- / - / R / O	91.36	83.83	91.08	83.68	91.41	95.14	**100.00**	**99.76**	**94.82**	**99.76**
S / B / R / -	91.20	71.98	83.84	76.20	85.55	–	–	–	–	–
S / B / - / O	**96.33**	50.05	69.30	62.62	74.90	–	–	–	–	–
S / - / R / O	90.85	85.61	92.00	83.97	92.25	–	–	–	–	–
- / B / R / O	91.12	85.03	91.70	83.93	91.97	95.14	**100.00**	**99.76**	**94.82**	**99.76**
S / B / R / O	91.00	86.22	92.32	84.37	92.55	–	–	–	–	–

Last but not least for the most imbalanced datasets from Table 8 like *lymphography-normal-fibrosis* proposed oversampling gives better recognition of minority class without violating classification of the instances from the second class.

Table 8: Preprocessing results for Tier 5 datasets

	lymphography-normal-fibrosis					winequality-red-4				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	96.07	88.00	93.50	89.74	93.82	77.97	**82.54**	**89.56**	66.01	**90.00**
S / - / - / -	–	–	–	–	–	–	–	–	–	–
- / B / - / -	**99.48**	74.00	85.13	83.33	87.00	39.42	4.76	13.09	8.22	52.38
- / - / R / -	92.90	79.00	88.36	82.07	89.24	88.67	32.18	55.18	44.19	65.95
- / - / - / O	98.07	87.00	93.01	90.88	93.41	**90.58**	69.82	83.32	75.37	84.64
S / B / - / -	–	–	–	–	–	–	–	–	–	–
S / - / R / -	–	–	–	–	–	–	–	–	–	–
S / - / - / O	–	–	–	–	–	–	–	–	–	–
- / B / R / -	92.72	71.00	83.56	76.52	85.24	85.11	31.23	54.12	42.05	65.41
- / B / - / O	98.07	87.00	93.01	90.88	93.41	89.74	71.73	84.39	**75.51**	85.55
- / - / R / O	98.16	**92.00**	**95.69**	**93.74**	**95.91**	82.93	71.35	83.41	66.33	84.83
S / B / R / -	–	–	–	–	–	–	–	–	–	–
S / B / - / O	–	–	–	–	–	–	–	–	–	–
S / - / R / O	–	–	–	–	–	–	–	–	–	–
- / B / R / O	98.16	**92.00**	**95.69**	**93.74**	**95.91**	82.10	73.73	84.67	66.58	85.97
S / B / R / O	–	–	–	–	–	–	–	–	–	–
	winequality-white-3_vs_7					kddcup-buffer_overflow_vs_back				
Method	**BAcc**	**Sen**	**GM**	**FM**	**AUC**	**BAcc**	**Sen**	**GM**	**FM**	**AUC**
SMOTE	94.76	**96.25**	**97.95**	**92.76**	**98.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
S / - / - / -	–	–	–	–	–	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
- / B / - / -	99.18	26.25	50.23	40.58	63.12	–	–	–	–	–
- / - / R / -	**99.43**	48.75	66.77	61.11	74.38	99.20	**100.00**	99.99	99.18	99.99
- / - / - / O	96.83	82.50	90.64	87.81	91.19	–	–	–	–	–
S / B / - / -	–	–	–	–	–	–	–	–	–	–
S / - / R / -	–	–	–	–	–	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
S / - / - / O	–	–	–	–	–	–	–	–	–	–
- / B / R / -	96.37	43.75	66.05	59.36	71.83	–	–	–	–	–
- / B / - / O	96.06	86.25	92.71	89.13	93.04	–	–	–	–	–
- / - / R / O	93.75	87.50	93.29	87.38	93.61	–	–	–	–	–
S / B / R / -	–	–	–	–	–	–	–	–	–	–
S / B / - / O	–	–	–	–	–	–	–	–	–	–
S / - / R / O	–	–	–	–	–	–	–	–	–	–
- / B / R / O	92.07	85.00	91.70	84.27	92.32	–	–	–	–	–
S / B / R / O	–	–	–	–	–	–	–	–	–	–

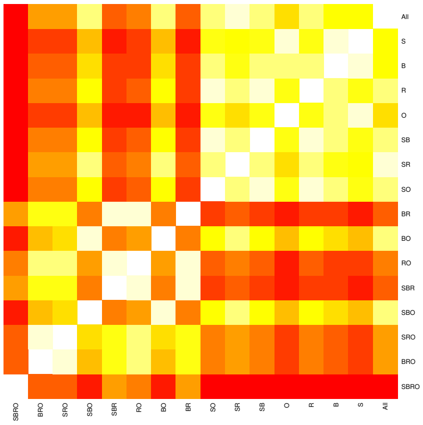

Figure 3: Shaffer post-hoc test results for comaprison among oversampling methods with G-mean as selected metric - yellow stands for statistically significant differences between a pair of considered approaches, red for a lack of such

In terms of *Geometric Mean* oversampling done on all regions, but separately proves to provide better performance than SMOTE done on minority class instances treated as a homogenous space. Taking into account results from Figure 3 similarity between *Safe / Borderline / Rare / Outlier* and *SMOTE* can be observed, as well as for *Safe / Borderline / Outlier* and *Safe / Rare / Outlier*. It proves that class disproportion is not the main source of difficulty regarding imbalanced data, local characteristics of minority class are also important and should be considered while dealing with imbalanced data.

6. Conclusions

In this paper we have examined the role of minority class instance types on the SMOTE imbalanced data preprocessing. Based on the fact that the imbalance ratio is not the sole source of learning difficulties, we have analyzed the structure of minority classes to identify properties of each object within. This information was used to empower SMOTE by changing it into a selective oversampling algorithm that focused on certain types of instances. We departed from an uniform oversampling in favor of enhancing the presence of only most challenging instances. Provided experimental study clearly showed that such data-driven oversampling may lead to an improved performance of base classifier. In our future works we plan to propose a fully automatic instance selection procedure for SMOTE, based on their local characteristics. Additionally, we work on combining clustering (Spurek, 2017) with our approach in order to more efficiently manage minority class sub-concepts.

Acknowledgments

This work was partially supported by the Polish National Science Center under the grant no. UMO-2015/19/B/ST6/01597

References

Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2):31:1–31:50, 2016.

Kay H. Brodersen, Cheng Soon Ong, Klaas E. Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. *International Conference on Pattern Recognition*, pages 1 – 4, 2010.

Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, pages 475–482, 2009.

Alberto Cano, Amelia Zafra, and Sebastián Ventura. Weighted data gravitation classification for standard and imbalanced data. *IEEE Trans. Cybernetics*, 43(6):1672–1687, 2013.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(16):321 – 357, 2002.

Wojciech Marian Czarnecki and Jacek Tabor. Extreme entropy machines: robust information theoretic classification. *Pattern Anal. Appl.*, 20(2):383–400, 2017.

Salvador Garcia and Francisco Herrera. An extension on ”statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677 – 2694, 2008.

Pablo D. Gutiérrez, Miguel Lastra, José M. Benítez, and Francisco Herrera. Smote-gpu: Big data preprocessing on commodity hardware for imbalanced classification. *Progress in Artificial Intelligence*, 2017. doi: 10.1007/s13748-017-0128-2. URL https://doi.org/10.1007/s13748-017-0128-2.

Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I*, pages 878–887, 2005.

Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, pages 1322–1328, 2008.

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

Pawel Ksieniewicz, Manuel Graña, and Michal Woźniak. Paired feature multilayer ensemble - concept and evaluation of a classifier. *Journal of Intelligent and Fuzzy Systems*, 32(2): 1427–1436, 2017.

Krystyna Napierala and Jerzy 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.

Krystyna Napierala, Jerzy Stefanowski, and Szymon Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *Rough Sets and Current Trends in Computing - 7th International Conference, RSCTC 2010, Warsaw, Poland, June 28-30,2010. Proceedings*, pages 158–167, 2010.

Rutvija Pandya and Jayati Pandya. C5.0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, 117(16):18 – 21, 2015.

Przemyslaw Spurek. General split gaussian cross-entropy clustering. *Expert Systems with Applications*, 68:58–68, 2017.

D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1 – 34, 1996.

Michal Woźniak, Manuel Graña, and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.