FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 42 (2017) No. 2
ISSN 0867-6356
e-ISSN 2300-3405
DE GRUYTER OPEN
DOI: 10.1515/fcds-2017-0007

DIFFICULTY FACTORS AND PREPROCESSING IN IMBALANCED DATA SETS: AN EXPERIMENTAL STUDY ON ARTIFICIAL DATA

Szymon WOJCIECHOWSKI, Szymon WILK *

Abstract. In this paper we describe results of an experimental study where we checked the impact of various difficulty factors in imbalanced data sets on the performance of selected classifiers applied alone or combined with several preprocessing methods. In the study we used artificial data sets in order to systematically check factors such as dimensionality, class imbalance ratio or distribution of specific types of examples (safe, borderline, rare and outliers) in the minority class. The results revealed that the latter factor was the most critical one and it exacerbated other factors (in particular class imbalance). The best classification performance was demonstrated by non-symbolic classifiers, particular by k-NN classifiers (with 1 or 3 neighbors – 1NN and 3NN, respectively) and by SVM. Moreover, they benefited from different preprocessing methods – SVM and 1NN worked best with undersampling, while oversampling was more beneficial for 3NN.

Keywords: imbalanced data, difficulty factors, preprocessing methods, learning and classification

1 Introduction

Data characterizing many real-world classification problems manifest an imbalanced distribution of examples across decision classes, namely, one of the decision classes is underrepresented (sometimes heavily) comparing to the others. A typical example is medical diagnosis where the number of patients from a critical class requiring special management is usually much smaller than the number of patients from remaining classes [32]. Such a phenomenon is referred to as the *class imbalance*, the underrepresented class is called the *minority* class, while the other classes are referred to as the *majority* classes.

*Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland, szymon.wilk@cs.put.poznan.pl

The class imbalance poses a difficulty for many learning algorithms – induced classifiers are often biased towards the majority classes [19, 7]. Many methods have been already proposed to deal with this challenge (see [12] for a review). They can be divided into two groups operating at the *algorithm level* and *data level.* Methods from the former group adjust a learning process appropriately so it can be applied directly to imbalanced data, while methods from the latter group preprocess learning data, e.g., by resampling. Although implementing modifications at the algorithm level can potentially lead to more extensive improvement in the classification performance, preprocessing methods constitute a dominant approach.

However, the imbalanced distribution of classes itself is not the only one or the major difficulty, but there are other factors that combined together with the class imbalance can lead to a serious deterioration of classification accuracy, especially for the minority class [24]. These other *data difficulty factors* include: small data set size [14], small disjuncts [15], overlapping regions of the minority and majority classes [9], and multiple minority class examples located inside the majority classes [24].

In order to approximate the data difficulty factors we employ the taxonomy of minority class examples proposed in [25]. This taxonomy defines four types of examples based on their local characteristic: *safe*, *borderline*, *rare* and *outlier.* Safe examples lie inside the minority class and are surrounded mostly by neighbors from the same class; borderline examples are located close to class boundaries and thus their neighborhood is a mixture of the majority and minority class examples; rare examples form small islands (consisting of 2-3 examples) inside the majority classes; finally outliers are isolated examples "thrown" into the majority classes. Borderline, rare and outlier examples are considered as *unsafe*, because they are more difficult to learn.

This paper extends our previous work where we focused mostly on preprocessing borderline and outlier examples [24]. Here we consider difficulty factors captured by various distributions of minority example types and combine them with the varying class imbalance and data dimensionality (i.e., we go beyond 2-dimensional data sets). In order to examine all these factors in a systematic way we use artificial data sets created with our new data generator (specifically, we consider two non-linear data shapes and modify data characteristics according to the difficulty factors). Moreover, our evaluation involves 6 classifiers (both non-symbolic and symbolic) and 5 preprocessing methods (both random and informed).

Summarizing, this study answers the following research questions (in all these questions we primarily focus on the classification performance with respect to the minority class):

1. What is the impact of the varying class imbalance and data dimensionality on the classification performance?

2. What is the impact of the varying distributions of minority example types on the classification performance?

3. What is the impact of preprocessing methods on the classification performance and what are the best performing combinations of preprocessing methods and classifiers?

The paper is organized as follows. The next section presents related work on data difficulty factors, their impact on the classification performance and comparison of different preprocessing methods. In Section 3 we present the data generator used to create artificial data set and describe the experimental design behind our study. The results of our experiments are presented in details in Section 4. Finally, Section 5 presents a discussion and concluding remarks.

2 Related work

One of the very first papers about imbalanced data was [19]. The authors found that the performance of classifiers significantly deteriorated when the majority class included much more examples than the minority one, and they proposed a *one-sided selection* method to reduce the number of examples from the majority class. They also explained how the class imbalance affected k-nearest neighbor (k-NN), Bayesian and tree-based classifiers. These conclusions were verified and broadened by subsequent research.

Jo and Japkowicz [16] revealed that the class imbalance itself was not a problem in itself, however, it may have resulted in small disjuncts in the minority class that actually degraded the classification performance. Also García et al. [9] claimed that the class imbalance was not the only cause of deteriorated performance, but an important role was played by an overlap of the minority and majority classes. This claim came from comparison of 6 learning algorithms on 2-dimensional artificial numerical data sets with the controlled class overlap. Moreover, the authors stated that effects of the class overlap combined with class imbalance were strongly dependent on the classifier characteristics. In turn, Napierala and Stefanowski [25] stated that the distribution of example types in the minority class had greater impact on the classification performance than other difficulty factors, e.g., class imbalance or the size of a data set. Their observation was based on the analysis of 26 real-life data sets. Besides that, they found that in most considered data sets the minority class included a large portion of unsafe examples, especially outliers. This finding was consistent with our study [32] where we also found that the minority class in 5 imbalanced real-life clinical data sets consisted mostly of borderline and outlier examples.

Other researchers were interested in examining the influence of difficulty factors on the performance of classifiers constructed using different learning algorithms. Japkowicz and Stephen [14] compared a tree-based classifier (induced with the C5.0 algorithm), multi-layer perceptron and a support vector machine (SVM) on a collection of artificial data sets. They concluded that SVM was not sensitive to small disjuncts at all, while the decision tree was heavily affected. García et al. [10] found out that comparing to other types of classifiers k-NN was more sensitive to the size of the class overlap than to the overall class imbalance, but yet the most critical factor was the local imbalance ratio.

Finally, some researchers also performed comparison of various preprocessing methods. García et al. [11] conducted an experiment on real data sets to examine the influence of the imbalance ratio and classifier characteristic on the classification accuracy.

Considered data sets were divided into two groups: imbalanced and balanced. That comparison involved 4 preprocessing methods and 8 classifiers. The authors concluded that in case of high imbalance oversampling performed better than undersampling (the latter may have removed too many important examples in order to balance the class distribution), while both preprocessing methods performed comparably when data sets were balanced. They also stated that selecting a preprocessing method was more important than selecting a classifier because of the smaller impact of the latter on the observed performance. Then, in [24] we concluded that focused preprocessing methods (like neighborhood cleaning rule, NCR, or SPIDER2) outperformed both random and cluster oversampling methods when dealing with borderline and outlier examples from the minority class. This observation was made on 2-dimensional artificial numerical data sets. In turn, in our study [32] involving difficult clinical data sets, all considered classifiers performed poorly, and while applying any preprocessing method improved the classification performance, the largest increase was observed for random undersampling.

3 Methods

3.1 Data generator

In our early work on artificial data sets and their analysis [24] we used a simple data generator limited to two-class and two-dimensional problems, and to safe and borderline examples. While it was sufficient to create data sets for the first round of computational experiments, it did not allow us to simulate more complex situations, e.g., the minority class containing both rare and outlier examples.

Given the above shortcomings, we decided to design and implement a new generator with improved functionality and thus increased versatility. Specifically, the new data generator has the following capabilities:

- Support for multi-class and multi-dimensional data sets (current implementation is limited to 40 dimensions due to restrictions imposed by libraries implementing quasi-random numbers, however, it should be sufficient to generate data sets with sizes similar to these of real-life data sets [25]),
- Support for decision classes composed of one or more regions. A region is defined either as a hyper-ellipsis or a hyper-rectangle with uniform or normal distribution of safe and borderline examples. It is also possible to associate diversified weights with regions from a given class to introduce intra-class imbalance. Finally, the generator supports special so-called *integumental* regions that fill empty space between regular regions. Such regions are usually introduced for the majority class, however, they are not obligatory, and the majority class can be defined using regular regions as well,
- Support for all four types examples introduced in [25] – safe, borderline, rare and outlier. It is also possible to specify a different distribution of example

types for each decision class (e.g., it is possible to obtain data sets with two differently distributed minority classes),

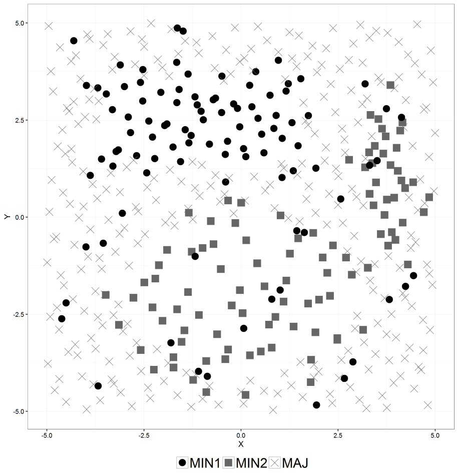
- Ability to switch between pseudo- and quasi-random numbers to obtain uniformly distributed examples. Quasi-random numbers fill the data space more uniformly than pseudo-random numbers (they result in a smaller number of empty "holes") and are often used in simulation and optimization [22]. Specifically, the new generator currently employs the Halton sequence to obtain quasi-random numbers, however, it can be easily modified to use other low-discrepancy sequences,
- Ability to generate pairs of learning and testing data sets where locations of rare and outlier examples are preserved (e.g., in both data sets such examples are located in similar positions). This allows for a more robust evaluation of learned classifiers, as we avoid situations that are quite likely in k-fold cross validation, where all rare and outlier examples from a certain area would be used for testing, thus preventing a classifier from learning them and deteriorating its accuracy.

The generator is controlled through a set of properties providing the characteristics of generated data. Listing 1 shows a sample configuration that defines a 2-dimensional 3-class data set with 500 examples categorized into two minority classes and the majority class (lines 1–7). The first minority class is composed of a single rectangular region and it includes all types of examples, including rare and outliers (lines 15–19). The second minority class consists of two elliptical regions (the former contains twice as many examples as the latter) and it is limited to safe and borderline examples (lines 21–29). Finally, the majority class is associated with a single integumental region (lines 31–35). The obtained data set is visualized in Figure 1.

Listing 1. Configuration for a sample data set

```
 1  attributes = 2
 2  classes = 3
 3  names.attributes = X, Y
 4  names.classes = MIN1, MIN2, MAJ
 5  names.decision = CLASS
 6  classRatio = 1:1:3
 7  examples = 500
 8
 9  minOutlierDistance = 0.5
10  defaultRegion.weight = 1
11  defaultRegion.distribution = U
12  defaultRegion.borderZone = 0.6
13  defaultRegion.noOutlierZone = 0.4
14
15  class.1.regions = 1
16  class.1.exampleTypeRatio = 50:20:20:10
17  class.1.region.1.shape = R
18  class.1.region.1.center = -1, 2.5
19  class.1.region.1.radius = 2.5, 1
20
21  class.2.regions = 2
22  class.2.exampleTypeRatio = 70:30:0:0
```

Figure 1. Visualization of the data set defined in Listing 1



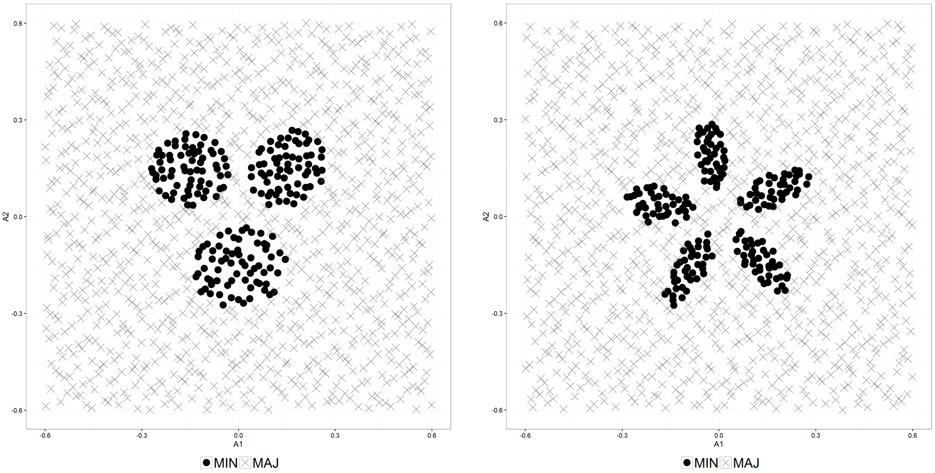
```
class.2.region.1.weight = 2
class.2.region.1.shape = C
class.2.region.1.center = 0, -2
class.2.region.1.radius = 3, 2
class.2.region.2.shape = C
class.2.region.2.center = 3.8, 1
class.2.region.2.radius = 0.5, 2

class.3.regions = 1
class.3.exapleTypeRatio = 100:0:0:0
class.3.region.1.shape = I
class.3.region.1.center = 0, 0
class.3.region.1.radius = 5, 5
```

3.2 Experimental design

An overall goal of our experimental study was to check if the characteristics of an imbalanced data set, in particular the dimensionality, the imbalance ratio and the distribution of example types, affected the performance of selected classifiers and how this performance could be improved using preprocessing methods. In the experiment we considered two data shapes – *paw3* and *flower5* – illustrated in Figure 2. Both shapes capture non-linear relation between attributes. In *flower5* the minority class resembles a flower with 5 elliptic petals, while *paw3* is composed of 3 regions and resembles a paw print. In both shapes the majority class constituted a single integumental region that filled an empty data space. Similar shapes were considered in our early study on imbalanced data [24] where they presented a challenge to classification methods comparable to the one associated with real-life data sets.

The numbers of examples for each shape were fixed throughout the experiment at



(a) *paw3* shape

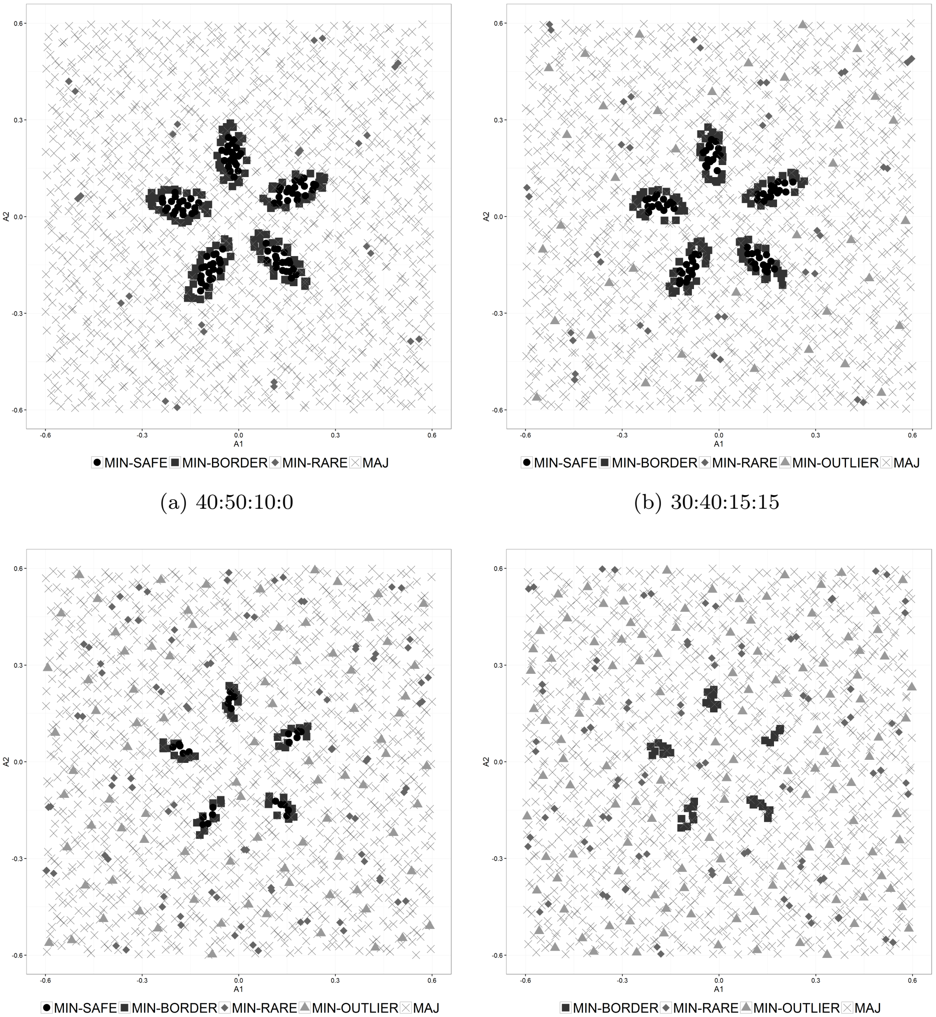
(b) *flower5* shape

Figure 2. Data shapes considered in the study

1200 for *paw3* and 1500 for *flower5*. A larger number of examples for the latter shape was associated with a larger number of regions in the minority class and allowed us to ensure reasonable number of safe examples in each region, even for larger imbalance ratios and "extreme" distributions of example types (see description below). We used 2-, 3-, 5- and 7-dimensional versions of each shape. Specifically, we systematically expanded the complexity by adding new dimensions to existing ones. Moreover, we considered four possible imbalance ratios – 1:5, 1:7, 1:9, and 1:13 that correspond to 16.7%, 12.5%, 10.0%, and 7.1% share of the minority class in a data set, respectively. Such imbalance ratios are typical for many benchmark real-life data sets (see [25, 26]).

We also considered 6 possible distributions of example types in the minority class (further in the text we refer to them as to type distributions), given in Table 1. These distributions were inspired by experimental evaluation from [25]. Specifically, the first two distributions – 100:0:0:0 and 70:30:0:0 correspond to *easy* data sets with the majority of safe examples (the former was used as a baseline). Distributions 40:50:10:0 and 30:40:15:15 correspond to *moderate* data sets where safe and borderline examples prevail in the minority class, however, the number of borderline examples is comparable to the number of safe ones. Finally, distributions 10:20:35:35 and 0:15:35:50 correspond to *difficult* data sets where most examples from the minority class are rare or outlier examples. Figure 3 illustrates how selected type distributions affect the 2-dimensional *flower5* shape with the imbalance ratio of 1:5. Summarizing, for each shape we considered 96 derived data configurations (4 dimensionalities × 4 imbalance ratios × 6 example type distributions), thus the total number of processed data configurations was equal to 192.

The considered preprocessing methods are listed in Table 2 (for their brief description see [32, 25]). We selected them to preserve consistency with previous research



(a) 40:50:10:0

(b) 30:40:15:15

(c) 10:20:35:35

(d) 0:15:35:50

Figure 3. Impact of the type example distributions on the *flower5* shape

Table 1. Considered type distributions in the minority class (S = safe, B = borderline, R = rare, O = outlier)

Distribution	S [%]	B [%]	R [%]	O [%]
100:0:0:0	100	0	0	0
70:30:0:0	70	30	0	0
40:50:10:0	40	50	10	0
30:40:15:15	30	40	15	15
10:20:35:35	10	20	35	35
0:15:35:50	0	15	25	50

Table 2. Considered preprocessing methods

Method	Description
none	no preprocessing (baseline)
RU	random undersampling
RO	random oversampling
NCR	neighborhood cleaning rule [20]
SM	SMOTE (Synthetic Minority Over-sampling TEchnique) [4]
SP2	SPIDER2 (Selective Preprocessing of Imbalanced Data, version 2) [24]

and to allow for more reliable comparison. Following our past experience [32], the RU, RO and SM methods were parametrized to produce a balanced distribution of classes in resulting data sets. Moreover, SM and SP2 were used with $k = 5$ nearest neighbors, and the latter was set for extended amplification of the minority class examples and for relabeling of the majority class examples (such settings were supported by our earlier studies [32, 24]).

The preprocessing methods were combined with classifiers given in Table 3 (all were implemented in WEKA). Such selection was driven by the consistency with previous research and also by the characteristic of specific classifiers (e.g., we selected RBF over a multi-layer perceptron due better handling of noisy data by the former [33]). For C4.5 and PART we considered both pruned and unpruned versions of induced classifiers for a comprehensive evaluation. Unpruned classifiers are generally suggested when class imbalance has been encountered [21]. However, our past results with imbalanced medical data sets showed that when combined with preprocessing methods pruned classifiers worked comparably or better than unpruned ones [32], thus, we wanted to further verify this finding.

Crucial parameters for RBF (standard deviation and the number of clusters) and SVM (gamma for the RBF kernel and complexity) were selected using a simple grid search (systematic exploration of possible combinations of parameter values [27]) over original (i.e., not preprocessed) data sets. During the search we optimized the geometric mean of sensitivity and specificity for the minority class (G-mean in short).

Table 3. Considered classifiers

Classifier	Description
1NN, 3NN, 5NN	k-NN classifier with Euclidean distance and k=1, 3 and 5 nearest neighbors, respectively
C45-P, C45-U	a tree-based classifier induced using the C4.5 algorithm with and without pruning, respectively
PART-P, PART-U	a rule-based classifier induced using the PART algorithm with and without pruning, respectively
NB	a naive Bayes classifier with a kernel density estimator
RBF	a radial basis function (RBF) neural network
SVM	a support vector machine with an RBF kernel

G-mean is typically used for imbalanced data as it avoids the bias associated with uneven distribution of classes. We considered the following ranges of parameters: standard deviation from 1e−3 to 1e−1, number of clusters from 3 to 60, gamma from 1 to 1e+3 and complexity from 1e+1 to 1e+5. We were not able to find a single set of parameters for each classifier that would have resulted in reasonable values of G-mean for all data sets. We also failed to find a single parameter configuration for a set of files with the same type distribution. Thus, we finally identified specific parameters for each of the 192 data configurations[1]. Interestingly, we observed certain patterns in the obtained configurations – more difficult sets required larger numbers of clusters for RBF and greater complexity for SVM, and the remaining parameters were less sensitive to the type distributions and thus more stable throughout the data configurations.

Most of the remaining classifiers were run with default values of parameters. Only for classifiers induced using PART and C4.5 algorithms we had to turn off the minimum description length correction for info gain. Otherwise, these algorithms were unable to handle difficult type distributions (they failed to construct tree paths or rules for the minority class). Moreover, in case of NB we decided to use a kernel density estimator, as in our preliminary tests it turned out to be better suited to numerical data than the other options (using normal distribution or internal discretization).

The classification performance was evaluated using sensitivity and specificity for the minority class and the already mentioned GM. We did not use the AUC (area under the ROC curve) measure, as the classifiers selected for our study gave deterministic predictions. Here we should also note that the minority class was set a priori and even though some of the processing methods heavily modified the class distribution (making the minority class most prevalent), we did not change it.

The above measures were estimated using a validation scheme with independent learning and testing data sets. More precisely, for each considered data configuration we generated 10 pairs of learning and testing sets and averaged results over these pairs. The size of the testing sets was fixed to 500 examples, while the learning sets

[1] A document with RBF and SVM parameters for specific data configurations is available at `http://www.cs.put.poznan.pl/swilk/fcds-generator/rbf-svm-params.pdf`

included either 1200 (*paw3*) or 1500 (*flower5*) examples. Obviously, preprocessing methods were applied to learning sets only. In order to gain better insight into differences between specific combinations of preprocessing methods and classifiers, we applied non-parametric Friedman test that globally compared their performance over multiple data sets [5]. We also carried out a post-hoc analysis (the Nemenyi test) of differences between ranks. All these tests were performed with $\alpha = 0.05$.

4 Results

4.1 Impact of the dimensionality and class imbalance on the classification performance

Here we study the impact of the dimensionality and the imbalance ratio on the performance of considered classifiers. Table 4 reports sensitivity for 4 selected classifiers applied with no preprocessing – 1NN, 3NN, PART-U and SVM (due to space limits we focused on most representative and interesting ones). To give better insight, we present results not only for the baseline 100:0:0:0 type distribution, but also for the most difficult one – 0:15:35:50.

The first observation is that for the easy type distribution lower imbalance ratios (1:5 – 1:9) had limited impact on sensitivity – a larger decrease was observed for the ratio of 1:13. This impact became more evident for the difficult type distribution where increasing class imbalance deteriorated the classification performance for the minority class (e.g., for 3NN or PART-U). Such finding is consistent with literature that associates data difficulties not only with class imbalance, but also with other factors (see discussion in Sections 1 and 2).

The obtained results also reveal that increasing dimensionality improved sensitivity (especially for difficult type distributions), although an extent of such improvement was dependent on the classifier. It was largest for 1NN (also for 3NN and RBF), and for other classifiers the improvement was more limited. While this may seem surprising, neural networks and support vector machines were already demonstrated to improve their performance with higher dimensionality [3]. Moreover, our findings are consistent with results presented in [23] where the authors explored the effect of the following three factors on classification accuracy: the length of the class boundary, the number of dimensions and the number of examples. The class boundary was established by constructing a minimum spanning over all examples and computing the ratio of edges connecting examples from different classes. A value close to 1.0 indicated that decision classes were highly interleaved, while a value close to 0.0 corresponded to their good separability. Results obtained in that study on real-life and artificial data revealed that the length of the class boundary was the most crucial factor – the other two did not affect the accuracy if the length was constant.

In our study increasing the dimensionality was associated with decreasing data density (the number of examples was constant). The drop in density was slower for the minority class, than for the majority one (when the number of dimensions increases, then the volume of a hyper-ellipsis ”shrinks” in comparison to the volume of a surrounding hyper-rectangle). In other words, examples from the minority class became more concentrated than the majority class examples, what improved the separability of both classes (and thus shortened the class boundary). Impact of the improved separability is especially visible for difficult data set and 1NN and 3NN classifiers.

In Table 5 we present specificity for the selected classifiers. It increased with the imbalance ratio. Moreover, in most cases it also increased with the dimensionality, and – as for sensitivity – the extent of change was dependent on the type distribution and the classifier. The largest increase was observed for the difficult type distribution and for the 1NN and PART-U classifiers. Similar changes were also observed for SVM, however, they were more evident for lower dimensionalities. Finally, specificity observed for 3NN was most stable in comparison to other classifiers.

We should also note that the data shape had negligible impact on the performance of classifiers — *paw3* despite apparent simplicity of (smaller number of regions in the minority class) demonstrated the same challenge as *flower5*.

4.2 Impact of the type distribution on the classification performance

In this section we discuss the impact of the type distribution on the performance of the considered classifiers applied to data sets without any preprocessing methods. While we checked the classifiers on all 192 data configurations, here we present detailed results for configurations with 3 dimensions and the imbalance ratio of 1:7. Observations for other configurations were similar – if there were any differences, we note them explicitly.

Table 6 shows sensitivity obtained by specific classifiers – in this table (and all subsequent ones) the best value in each row is marked in bold and the second best in italics. The impact of the type distribution on the performance is clear – sensitivity deteriorated when the ratio of rare cases and outliers increased. This effect was especially visible when moving from moderate to difficult data sets. For 2-dimensional data also the differences between easy and moderate sets stood out, and for larger numbers of dimensions extensive changes in sensitivity were encountered when switching within the moderate type distributions from 40:50:10:0 to 30:40:15:15. This may imply that outliers always caused problems for classifiers, while the undesirable impact of a limited number of rare cases was mitigated by increased dimensionality.

The results also allowed us to identify most competent classifiers for specific groups of data sets. For easy data sets it was 5NN (and often SVM), for moderate data sets the best sensitivity was demonstrated by SVM, 3NN and 1NN. Finally, for difficult data sets 1NN performed best. Here we need to highlight performance of SVM, which was a strong competitor for k-NN classifiers for easy and moderate data sets. It only lost to 1NN on difficult data sets with larger number of dimensions (5 and 7), however, even then it was the second best classifier. Interestingly, unlike in other related studies (e.g., [25]), none of the symbolic classifiers was able to demonstrate a competitive

Table 4. Sensitivity for selected classifiers and varying dimensionality and imbalance ratio (2d = 2 dimensions, 3d = 3 dimensions, 5d = 5 dimensions, 7d = 7 dimensions)

		1NN				3NN				PART-U				SVM			
		1:5	1:7	1:9	1:13	1:5	1:7	1:9	1:13	1:5	1:7	1:9	1:13	1:5	1:7	1:9	1:13
									paw3								
2d	100-0-0-0	0.964	0.917	0.934	0.850	0.972	0.957	0.938	0.856	0.922	0.895	0.902	0.831	0.976	0.951	0.942	0.936
	0-15-35-50	0.375	0.333	0.314	0.286	0.217	0.146	0.122	0.097	0.234	0.198	0.192	0.175	0.413	0.338	0.364	0.322
3d	100-0-0-0	0.983	0.954	0.956	0.925	1.000	0.997	0.996	0.961	0.925	0.892	0.892	0.811	0.999	0.983	0.980	0.969
	0-15-35-50	0.547	0.546	0.536	0.472	0.308	0.337	0.300	0.236	0.266	0.224	0.184	0.164	0.531	0.544	0.512	0.483
5d	100-0-0-0	1.000	0.998	0.998	0.997	1.000	1.000	1.000	1.000	0.943	0.941	0.924	0.886	1.000	0.998	0.998	1.000
	0-15-35-50	0.793	0.800	0.810	0.800	0.383	0.390	0.408	0.375	0.284	0.254	0.276	0.217	0.651	0.716	0.694	0.706
7d	100-0-0-0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.927	0.919	0.956	0.925	1.000	0.997	0.994	1.000
	0-15-35-50	0.939	0.921	0.936	0.908	0.442	0.430	0.446	0.394	0.299	0.289	0.280	0.253	0.649	0.646	0.648	0.614
									flower5								
2d	100-0-0-0	0.955	0.925	0.898	0.853	0.986	0.967	0.944	0.856	0.899	0.848	0.828	0.744	0.982	0.957	0.956	0.908
	0-15-35-50	0.367	0.344	0.354	0.306	0.227	0.198	0.170	0.103	0.270	0.197	0.238	0.194	0.414	0.373	0.408	0.344
3d	100-0-0-0	0.992	0.976	0.978	0.958	1.000	0.997	0.998	0.997	0.942	0.897	0.904	0.864	1.000	0.998	0.998	0.994
	0-15-35-50	0.559	0.557	0.552	0.511	0.282	0.297	0.312	0.261	0.292	0.244	0.240	0.167	0.518	0.519	0.546	0.528
5d	100-0-0-0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.984	0.952	0.958	0.936	1.000	1.000	1.000	1.000
	0-15-35-50	0.811	0.824	0.802	0.778	0.400	0.371	0.396	0.319	0.305	0.295	0.252	0.272	0.663	0.686	0.732	0.694
7d	100-0-0-0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.978	0.954	0.956	0.944	1.000	1.000	1.000	1.000
	0-15-35-50	0.928	0.906	0.940	0.939	0.446	0.413	0.444	0.400	0.333	0.292	0.300	0.264	0.659	0.663	0.688	0.669

Table 5. Specificity for selected classifiers and varying dimensionality and imbalance ratio (2d = 2 dimensions, 3d = 3 dimensions, 5d = 5 dimensions, 7d = 7 dimensions)

		1NN				3NN				PART-U				SVM			
		1:5	1:7	1:9	1:13	1:5	1:7	1:9	1:13	1:5	1:7	1:9	1:13	1:5	1:7	1:9	1:13
									paw3								
2d	100-0-0-0	0.981	0.985	0.986	0.991	0.982	0.989	0.990	0.995	0.984	0.987	0.983	0.988	0.992	0.992	0.993	0.994
	0-15-35-50	0.801	0.868	0.902	0.925	0.934	0.961	0.972	0.974	0.821	0.876	0.910	0.921	0.785	0.869	0.885	0.899
3d	100-0-0-0	0.972	0.976	0.974	0.984	0.964	0.972	0.974	0.983	0.985	0.986	0.989	0.988	0.988	0.992	0.990	0.994
	0-15-35-50	0.829	0.868	0.882	0.923	0.932	0.957	0.968	0.978	0.832	0.870	0.892	0.927	0.865	0.895	0.891	0.931
5d	100-0-0-0	0.987	0.983	0.984	0.988	0.981	0.979	0.979	0.986	0.989	0.989	0.990	0.990	1.000	1.000	0.999	0.997
	0-15-35-50	0.866	0.896	0.912	0.924	0.952	0.962	0.969	0.975	0.840	0.868	0.897	0.920	0.891	0.911	0.928	0.941
7d	100-0-0-0	0.986	0.989	0.988	0.989	0.981	0.987	0.984	0.986	0.989	0.989	0.992	0.994	1.000	1.000	1.000	1.000
	0-15-35-50	0.864	0.875	0.922	0.937	0.944	0.954	0.974	0.979	0.842	0.877	0.900	0.926	0.987	0.978	0.994	0.994
									flower5								
2d	100-0-0-0	0.974	0.980	0.983	0.986	0.972	0.979	0.984	0.993	0.971	0.978	0.982	0.981	0.985	0.988	0.991	0.994
	0-15-35-50	0.810	0.822	0.879	0.914	0.927	0.949	0.968	0.976	0.823	0.868	0.889	0.920	0.794	0.812	0.867	0.896
3d	100-0-0-0	0.982	0.984	0.981	0.986	0.974	0.979	0.977	0.983	0.983	0.986	0.982	0.986	0.990	0.993	0.988	0.986
	0-15-35-50	0.829	0.876	0.892	0.955	0.938	0.959	0.965	0.978	0.826	0.878	0.890	0.934	0.832	0.886	0.906	0.955
5d	100-0-0-0	0.994	0.993	0.994	0.994	0.988	0.989	0.990	0.990	0.994	0.994	0.995	0.995	1.000	1.000	1.000	1.000
	0-15-35-50	0.858	0.887	0.894	0.929	0.941	0.962	0.966	0.978	0.843	0.873	0.897	0.926	0.865	0.899	0.912	0.943
7d	100-0-0-0	0.994	0.996	0.995	0.995	0.991	0.992	0.992	0.993	0.995	0.996	0.996	0.995	1.000	1.000	1.000	1.000
	0-15-35-50	0.837	0.901	0.906	0.936	0.942	0.966	0.972	0.981	0.832	0.876	0.901	0.926	0.982	0.989	0.989	0.997

Table 6. Sensitivity for all considered classifiers and varying type distributions (3 dimensions, imbalance ratio 1:7)

	1NN	3NN	5NN	C45-P	C45-U	PART-P	PART-U	NB	RBF	SVM
					paw3					
100:0:0:0	0.954	0.997	**1.000**	0.898	0.894	0.860	0.892	0.873	0.949	*0.983*
70:30:0:0	0.951	0.986	**0.998**	0.911	0.900	0.886	0.887	0.876	0.937	*0.992*
40:50:10:0	0.883	**0.935**	0.905	0.786	0.784	0.775	0.771	0.625	0.889	*0.917*
30:40:15:15	*0.786*	0.775	0.692	0.578	0.624	0.571	0.619	0.225	0.683	**0.803**
10:20:35:35	**0.632**	0.440	0.306	0.230	0.337	0.305	0.330	0.000	0.340	*0.592*
0:15:35:50	**0.546**	0.337	0.149	0.013	0.227	0.186	0.224	0.000	0.241	*0.544*
					flower5					
100:0:0:0	0.976	0.997	**1.000**	0.930	0.903	0.916	0.897	0.987	0.925	*0.998*
70:30:0:0	0.979	*0.998*	**1.000**	0.944	0.916	0.921	0.890	0.987	0.981	0.994
40:50:10:0	0.933	*0.940*	0.905	0.822	0.814	0.822	0.810	0.890	0.908	**0.948**
30:40:15:15	*0.824*	0.759	0.700	0.605	0.649	0.637	0.654	0.659	0.708	**0.840**
10:20:35:35	**0.621**	0.449	0.302	0.249	0.362	0.325	0.384	0.000	0.319	*0.589*
0:15:35:50	**0.557**	0.297	0.151	0.068	0.251	0.198	0.244	0.000	0.200	*0.519*

performance. This may be associated with the characteristics of synthetic data and with more extensive tuning of parameters for some of non-symbolic classifiers (SVM and RBF) – we discuss it in Section 5. We should also note that unpruned symbolic classifiers generally performed better than their pruned variants. While pruning worked well for easy data sets, it became detrimental for difficult data sets, thus confirming observations from [21]. We also observed that *flower5* was somewhat easier for symbolic classifiers (especially for moderate and difficult types distributions), however, differences were minor – usually around 3–5%. Finally, we need to point out poor performance of the NB classifier. While it was acceptable for the easy type distributions, it failed for the difficult ones and it was unable to correctly recognize any example from the minority class.

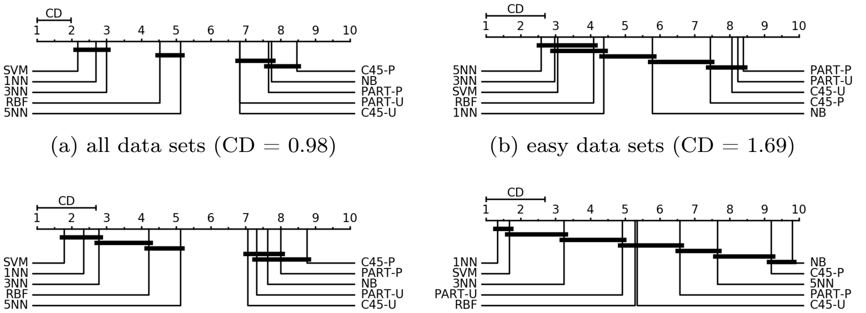
To further validate the above findings and to examine the importance of differences between specific classifiers given their sensitivity we conducted the Friedman test. It relied on ranks of classifiers (in our analysis these ranks were interpreted as positions in a ranking, i.e., the lower, the better) and involved all 192 data configurations. The *null* hypothesis saying that all classifiers performed equally was rejected ($p < 5\mathrm{e}-8$). The ranking of classifiers according to their average positions across all data sets is given in Figure 4a. The figure also gives the critical difference (CD) according to the Nemeyi test. The best classifiers were SVM, 1NN and 3NN and their sensitivity was significantly better than for the remaining classifiers. Also, there was a significant difference between two groups of classifiers – one involving k-NN, SVM and RBF) and the other with symbolic classifiers and NB. The ranking also indicates that unpruned classifiers induced using C4.5 and PART algorithms performed better than their pruned counterparts, although the differences was statistically significant only for C4.5.

We also applied the Friedman separately to easy, moderate and difficult data sets in order to get better insight into performance of the classifiers in these groups. In each test the *null* hypothesis was rejected as well. The obtained rankings and associated critical values are given in Figure 4b, 4c and 4d, respectively. The rankings for easy and moderate data sets are very similar to the ranking for all data sets, and the ranking for difficult data sets reveals several interesting observations. While the top of this ranking remained unchanged (1NN, SVM and 3NN), the PART-U and C45-U classifiers were "promoted". This is consistent with results from [25] according to which these two classifiers were well suited for data sets with many rare cases and outliers.

Additionally, in Table 7 we present G-mean for specific classifiers. As previously, we observed a larger deterioration of this measure when moving from moderate to difficult data set that was caused by the drop in sensitivity (specificity was relatively stable). Moreover, SVM was the best classifier for easy and moderate data sets – only for difficult ones 1NN took the lead due to its better sensitivity. However, SVM took advantage of its better specificity for the data sets with 5 and 7 dimensions, and the differences in G-mean between these two classifiers were smaller than for sensitivity.

Table 7. G-mean for all considered classifiers and varying type distributions (3 dimensions, imbalance ratio 1:7)

	1NN	3NN	5NN	C45-P	C45-U	PART-P	PART-U	NB	RBF	SVM
					paw3					
100:0:0:0	0.965	*0.984*	*0.984*	0.942	0.939	0.921	0.938	0.916	0.967	**0.987**
70:30:0:0	0.961	0.976	*0.982*	0.946	0.940	0.933	0.932	0.917	0.963	**0.985**
40:50:10:0	0.920	**0.946**	0.931	0.876	0.870	0.865	0.863	0.776	0.929	*0.944*
30:40:15:15	0.852	*0.861*	0.820	0.753	0.774	0.741	0.761	0.471	0.816	**0.865**
10:20:35:35	**0.741**	0.645	0.546	0.473	0.554	0.529	0.541	0.000	0.574	*0.727*
0:15:35:50	*0.686*	0.566	0.383	0.035	0.450	0.409	0.437	0.000	0.480	**0.696**
					flower5					
100:0:0:0	0.980	*0.988*	*0.988*	0.956	0.943	0.949	0.940	0.979	0.960	**0.996**
70:30:0:0	0.980	*0.987*	0.985	0.963	0.949	0.952	0.936	0.977	0.987	**0.991**
40:50:10:0	0.953	*0.956*	0.939	0.898	0.890	0.896	0.888	0.924	0.943	**0.966**
30:40:15:15	*0.884*	0.855	0.826	0.770	0.788	0.782	0.786	0.800	0.836	**0.898**
10:20:35:35	**0.757**	0.658	0.544	0.495	0.578	0.551	0.582	0.000	0.560	*0.745*
0:15:35:50	**0.697**	0.533	0.385	0.216	0.472	0.424	0.461	0.000	0.436	*0.676*



(a) all data sets (CD = 0.98)

(b) easy data sets (CD = 1.69)

(c) moderate data sets (CD = 1.69)

(d) difficult data sets (CD = 1.69)

Figure 4. Rankings of classifiers (based on their sensitivity)

4.3 Impact of preprocessing methods on the classification performance

In this section we discuss the impact of preprocessing methods on the classification performance of selected classifiers. Specifically, we focus on the classifiers identified as the best ones in the previous section – SVM and 3NN (1NN behaved similarly to SVM, therefore, we excluded it from the detailed analysis). Moreover, for better comparison with findings from [25] we also include PART-U. As previously, we present detailed results for 3 dimensions and the imbalance ratio of 1:7, however, the discussion covers other data configurations as well.

In Tables 8 and 9 we show sensitivity for SVM and 3NN, respectively. In comparison to the baseline (none) application of preprocessing methods in most cases improved the performance. An extent of this improvement was dependent on the type distribution – for easy data set the improvement was negligible due to a stellar baseline performance, however, it intensified for moderate data sets and became very strong for difficult ones. Moreover, 3NN turned out to be more prone to improvement than SVM and when combined with preprocessing, it resulted in better sensitivity than SVM. Finally, both considered data shapes were similar in terms of the observed performance of 3NN and SVM and we were not able to point at any of the shapes as more difficult than the other.

SVM and 3NN required different preprocessing methods to attain the best observable performance. In case of SVM the best method was RU that consistently led to the best sensitivity (its superiority was especially evident for moderate and difficult data sets). The other method worth mentioning was NCR. On the contrary, for 3NN the best results were observed for RO and SP2.

In Table 10 we also present sensitivity for PART-U. As previously, preprocessing always improved its performance, however, the increase was smaller than for SVM and 3NN (especially for difficult data sets). Moreover, both considered data shapes were comparable in terms of their difficulty. Similarly to SVM, the most suitable preprocessing method was RU – only for easy data sets SP2 resulted in better performance.

Table 8. Sensitivity for SVM combined with preprocessing methods and varying type distributions (3 dimensions, imbalance ratio 1:7)

	SVM					
	none	RO	RU	NCR	SM	SP2
			paw3			
100:0:0:0	0.983	0.983	*0.995*	*0.995*	0.986	**0.998**
70:30:0:0	*0.992*	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
40:50:10:0	0.917	0.916	**0.971**	*0.965*	0.941	0.949
30:40:15:15	0.803	0.832	**0.916**	*0.905*	0.878	0.848
10:20:35:35	0.592	0.606	**0.894**	*0.776*	0.714	0.635
0:15:35:50	0.544	0.562	**0.903**	*0.768*	0.697	0.589
			flower5			
100:0:0:0	*0.998*	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
70:30:0:0	0.994	*0.997*	**1.000**	**1.000**	*0.997*	**1.00**0
40:50:10:0	0.948	0.948	**0.981**	*0.971*	0.957	0.956
30:40:15:15	0.840	0.848	**0.957**	*0.910*	0.862	0.854
10:20:35:35	0.589	0.622	**0.895**	*0.825*	0.729	0.635
0:15:35:50	0.519	0.552	**0.884**	*0.737*	0.714	0.565

Table 9. Sensitivity for 3NN combined with preprocessing methods and varying type distributions (3 dimensions, imbalance ratio 1:7)

	3NN					
	none	RO	RU	NCR	SM	SP2
			paw3			
100:0:0:0	*0.997*	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
70:30:0:0	0.986	**1.000**	**1.000**	**1.000**	*0.998*	**1.000**
40:50:10:0	0.935	**0.997**	*0.995*	0.968	0.976	0.997
30:40:15:15	0.775	**0.973**	0.871	0.814	0.889	*0.970*
10:20:35:35	0.440	**0.949**	0.759	0.568	0.811	*0.941*
0:15:35:50	0.337	**0.933**	0.733	0.470	0.819	*0.932*
			flower5			
100:0:0:0	0.997	**1.000**	**1.000**	**1.000**	*0.998*	**1.000**
70:30:0:0	*0.998*	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
40:50:10:0	0.940	**0.994**	0.990	0.967	0.970	*0.992*
30:40:15:15	0.759	**0.979**	0.879	0.806	*0.883*	**0.979**
10:20:35:35	0.449	**0.957**	0.775	0.598	*0.821*	**0.957**
0:15:35:50	0.297	**0.946**	0.738	0.460	0.824	*0.940*

For all these classifiers we performed Friedman tests to check differences between specific preprocessing methods considering the observed sensitivity. In each case the *null* hypothesis was rejected (with $p < 5\mathrm{e} - 8$). Moreover, we conducted similar tests

Table 10. Sensitivity for PART-U combined with preprocessing methods and varying type distributions (3 dimensions, imbalance ratio 1:7)

	PART-U					
	none	RO	RU	NCR	SM	SP2
			paw3			
100:0:0:0	0.892	0.884	*0.948*	0.910	0.894	**0.965**
70:30:0:0	0.887	0.868	*0.959*	0.933	0.919	**0.987**
40:50:10:0	0.771	0.754	**0.913**	0.843	0.844	*0.887*
30:40:15:15	0.619	0.597	**0.798**	0.684	*0.735*	0.694
10:20:35:35	0.330	0.329	**0.702**	0.432	*0.556*	0.368
0:15:35:50	0.224	0.213	**0.667**	0.370	*0.470*	0.278
			flower5			
100:0:0:0	0.897	0.902	*0.981*	0.964	0.940	**0.992**
70:30:0:0	0.890	0.883	*0.965*	0.962	0.927	**0.987**
40:50:10:0	0.810	0.814	**0.905**	0.870	0.879	*0.897*
30:40:15:15	0.654	0.633	**0.841**	0.744	*0.768*	0.722
10:20:35:35	0.384	0.360	**0.700**	0.468	*0.570*	0.429
0:15:35:50	0.244	0.230	**0.654**	0.383	*0.522*	0.268

for specific group of data files. The results are presented in Figures 5, 6 and 7. The obtained rankings confirm and generalize our findings discussed above. In case of SVM the best method was clearly RU. It was always ranked first and its rank (with the exception of easy data sets) was statistically better than the rank of the second best method – NCR. For easy data sets SP2 became a stronger competitor that overtook NCR. For 3NN the best two methods were RO and SP2 (RO in most cases was ranked first, but the difference was statistically significant only when considering all data sets). Interestingly, for easy data sets all methods were ranked similarly (no statistical differences), and RU was located before RO. Finally, observations for PART-U are very similar to those for SVM – RU was the best method, only for easy data sets it was superseded by SP2, however, the difference in ranks was not statistically significant.

For a more comprehensive overview, in Tables 11, 12 and 13 we show G-mean for SVM, 3NN and PART-U, respectively. These tables reveal that preprocessing improved (or in case of easy data sets it did not worsen) the classification performance. For SVM the largest improvement was observed for NCR and sometimes RU (usually for difficult data sets). In case of 3NN RO and SP2 worked best. Finally, for PART-U positive impact on GM was observed for SM, SP2 and NCR. The latter method had also positive impact on GM for 1NN.

Table 11. G-mean for SVM combined with preprocessing methods and varying type distributions (3 dimensions, imbalance ratio 1:7)

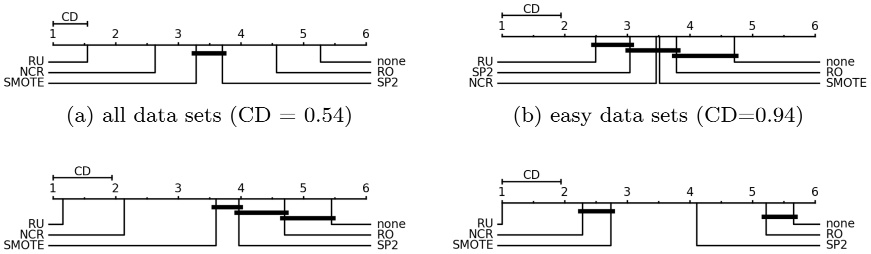
	SVM					
	none	RO	RU	NCR	SM	SP2
			paw3			
100:0:0:0	*0.987*	*0.987*	0.974	0.983	**0.988**	0.978
70:30:0:0	*0.985*	**0.987**	0.973	*0.985*	**0.987**	0.978
40:50:10:0	*0.944*	0.942	0.920	**0.954**	0.937	0.942
30:40:15:15	0.865	*0.877*	0.807	**0.898**	0.856	0.873
10:20:35:35	0.727	0.735	*0.788*	**0.818**	0.718	0.745
0:15:35:50	0.696	0.706	0.713	**0.806**	0.695	*0.718*
			flower5			
100:0:0:0	**0.996**	**0.996**	0.991	*0.993*	**0.996**	0.984
70:30:0:0	*0.991*	**0.993**	0.990	*0.991*	**0.993**	0.984
40:50:10:0	0.966	0.966	**0.975**	*0.973*	0.953	0.956
30:40:15:15	0.898	*0.901*	**0.927**	**0.927**	0.856	0.896
10:20:35:35	0.745	0.763	*0.790*	**0.855**	0.741	0.765
0:15:35:50	0.676	0.695	0.696	**0.776**	*0.701*	0.696

Table 12. G-mean for 3NN combined with preprocessing methods and varying type distributions (3 dimensions, imbalance ratio 1:7)

	3NN					
	none	RO	RU	NCR	SM	SP2
			paw3			
100:0:0:0	**0.984**	0.973	0.944	*0.980*	*0.980*	0.975
70:30:0:0	0.976	0.970	0.941	**0.978**	*0.977*	0.973
40:50:10:0	0.946	0.954	0.902	**0.955**	0.945	**0.955**
30:40:15:15	0.861	**0.907**	0.825	*0.874*	0.860	**0.907**
10:20:35:35	0.645	**0.825**	0.662	0.715	*0.741*	**0.825**
0:15:35:50	0.566	*0.792*	0.609	0.648	0.722	**0.795**
			flower5			
100:0:0:0	**0.988**	0.983	0.961	0.986	*0.987*	0.982
70:30:0:0	**0.987**	0.982	0.961	0.984	*0.985*	0.981
40:50:10:0	0.956	**0.968**	0.926	0.962	0.956	*0.965*
30:40:15:15	0.855	**0.923**	0.834	0.875	0.869	*0.922*
10:20:35:35	0.658	**0.853**	0.678	0.739	*0.761*	**0.853**
0:15:35:50	0.533	**0.791**	0.623	0.637	0.721	*0.790*

Table 13. G-mean for PART-U combined with preprocessing methods and varying type distributions (3 dimensions, imbalance ratio 1:7)

	PART-U					
	none	RO	RU	NCR	SM	SP2
			paw3			
100:0:0:0	0.938	0.934	0.936	*0.941*	0.937	**0.963**
70:30:0:0	0.932	0.923	0.944	*0.953*	0.949	**0.972**
40:50:10:0	0.863	0.851	0.879	*0.898*	0.884	**0.912**
30:40:15:15	0.761	0.750	0.726	*0.791*	0.779	**0.797**
10:20:35:35	0.541	0.541	0.582	*0.600*	**0.627**	0.567
0:15:35:50	0.437	0.432	*0.547*	*0.547*	**0.564**	0.492
			flower5			
100:0:0:0	0.940	0.943	0.967	*0.970*	0.961	**0.979**
70:30:0:0	0.936	0.931	0.958	*0.969*	0.954	**0.977**
40:50:10:0	0.888	0.889	0.892	*0.915*	0.908	**0.923**
30:40:15:15	0.786	0.772	0.760	**0.827**	0.805	*0.820*
10:20:35:35	0.582	0.568	0.598	*0.628*	**0.644**	0.615
0:15:35:50	0.461	0.442	0.540	*0.554*	**0.586**	0.479



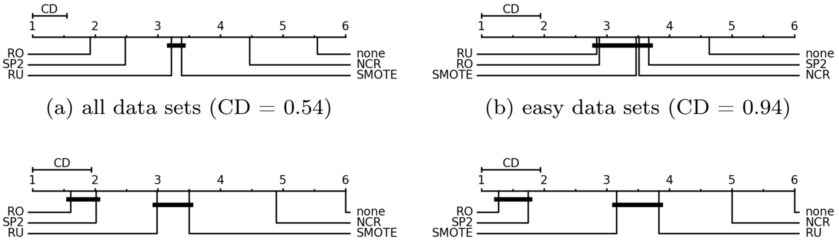
(a) all data sets (CD = 0.54)

(b) easy data sets (CD=0.94)

(c) moderate data sets (CD=0.94)

(d) difficult data sets (CD=0.94)

Figure 5. Rankings of preprocessing methods for SVM (based on their sensitivity)



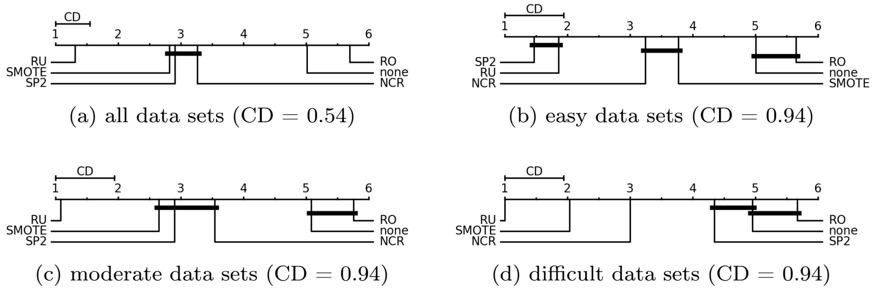
(a) all data sets (CD = 0.54)

(b) easy data sets (CD = 0.94)

(c) moderate data sets (CD = 0.94)

(d) difficult data sets (CD = 0.94)

Figure 6. Rankings of preprocessing methods for 3NN (based on their sensitivity)



(a) all data sets (CD = 0.54)

(b) easy data sets (CD = 0.94)

(c) moderate data sets (CD = 0.94)

(d) difficult data sets (CD = 0.94)

Figure 7. Rankings of preprocessing methods for PART-U (based on their sensitivity)

5 Discussion

The study described in this paper is a follow-up to our earlier work on artificial imbalanced data [24]. The new data generator allowed us to consider other types of minority class examples than safe and borderline. Moreover, we considered data sets with varying dimensionality and imbalance ratio. Finally, we expanded the set of examined preprocessing methods and classifiers. The experimental design – in particular considered distributions of safe, borderline, rare and outlier examples (corresponding to easy, moderate and difficult data sets), selected classifiers and preprocessing methods – was inspired by the work described in [25]. There were, however, several important differences that may have affected the results – they are briefly discussed below:

1. All the data sets considered in this study were numerical (as in [24]) what gave advantage to distance-based classifiers (e.g., k-NN) that demonstrated very good classification performance,

2. Instead of k-fold cross validation we employed a scheme with pairs of corresponding learning and testing sets. This may have resulted in more optimistic evaluation of performance, as such scheme saved us from situations where examples in certain data regions were underrepresented in a learning set and overrepresented in a testing set (for this reason we employed this scheme already in several experiments described in [24]). While adoption of this validation scheme does not allow for direct comparison of qualitative results, trends observed in classification performance and obtained rankings of classifiers and preprocessing methods can be still compared to other reports.

3. We performed more thorough optimization of parameters for SVM and RBF neural network classifiers. Unlike in [25] we failed to find a single set of parameters for each of these classifiers that would have resulted in reasonable performance on all data configurations, and ultimately we optimized selected parameters for each individual configuration. This gave handicap to these two classifiers and positively affected their performance.

With the above factors in mind, the major findings from our study, and at the same responses to questions formulated in Section 1, are as follows:

1. The impact of the data shape, dimensionality and imbalance ratio (within the scope considered in the study) on the classification performance was limited. Both shapes (*paw3* and *flower5*) were comparable in terms of difficulty, however, the latter presented a greater computational challenge for selected classifiers (in particular SVM) than the former. Deterioration of sensitivity associated with increasing the imbalance ratio alone from 1:5 to 1:13 in most cases did not exceed 10%. Moreover, increasing the dimensionality improved the performance (in terms of sensitivity and G-mean) – data sets with 2 and 3 dimensions were more difficult than those with 5 and 7 dimensions. This improvement was consistent with results from [23] where the authors showed the impact of the length of the class boundary (capturing how examples from various classes were interleaved) on the classification accuracy. Introduction of additional dimensions decreased data density, especially for the majority class, and improved the separability of examples from both classes, thus shortening the class boundary.

   The critical factor affecting the performance in terms of sensitivity was the distribution of example types. We observed the first large drop in performance when introducing outliers (15% of all minority class examples) and then when increasing their ratio from 15% to 35%. Introducing rare examples alone had much more limited impact (it was only visible for 2 dimensional data sets), thus we hypothesize outliers pose a special challenge for classifiers and preprocessing methods. Difficult data distributions also exacerbated the problem of class imbalance – its impact on performance was more visible in difficult data sets than in easy and moderate ones. These observations are consistent with the literature [9, 25, 16] and highlight the importance of other data difficulty factors than the sole class imbalance.

2. When no preprocessing methods were applied, the best sensitivity was demonstrated by SVM and k-NN with 1 and 3 neighbors (1NN and 3NN, respectively). Moreover, SVM was also best given G-mean, and 1NN and 3NN were second best. Further breakdown into groups of files revealed that for easy data sets 3NN (and also 5NN) was better than 1NN, however, the latter demonstrated its advantage on moderate and difficult data sets, which was consistent with findings from [10, 29].

   While symbolic classifiers were not competitive for non-symbolic ones, we observed that for difficult data sets an unpruned rule-based classifier induced by PART (PART-U) became the 4th classifier given its sensitivity. It overtook not only all other symbolic classifiers, but also some non-symbolic ones, (e.g., RBF, however, the difference was not statistically significant). Better performance on difficult data sets was also observed for an unpruned tree-based classifier induced by C4.5 (C45-U). Such behavior was consistent with findings from [25] about suitability of PART-U and C45-U for data sets with prevalent rare and outlier examples. Poorer performance of these classifiers in comparison to the leaders (1NN, 3NN and SVM), inconsistent with results from [25], may be explained by lower dimensionality of the data sets considered in our study (larger number of dimensions may deteriorate the performance of k-NN [29]) and by more extensive customization of parameters for SVM (and also for RBF). Finally, consistently with the literature [21], we observed that unpruned classifiers performed better than their pruned variants. While on easy and moderate data sets the differences were minor, they revealed their advantage on difficult data sets.

3. We observed that preprocessing methods were able to improve the performance of classifiers in terms of both sensitivity and G-mean. Moreover, the improvement of sensitivity in certain cases exceeded 60%. While it was larger than reported by other researchers (improvement of 30% or less [25, 2]), it could be attributed to the employed validation scheme (comparable improvements were observed in our earlier study [24] where we used pairs of learning and testing examples). We were able to identify certain combinations of classifiers and preprocessing methods that optimized the performance. For SVM undersampling methods worked best (this was reported by other researchers [28, 17] and we also observed it in our earlier study [32]). Specifically, random undersampling (RU) maximized sensitivity, while neighborhood cleaning rule (NCR) was most beneficial for G-mean (these two methods had the same impact on 1NN). On the other hand, 3NN worked best with oversampling – in particular with random oversampling (RO) and SPIDER2 (SP2) which improved both sensitivity and G-mean. While RU is generally preferred over RO [6], similar results showing beneficial impact of RO on 3-NN were presented in [31]. Finally, PART-U benefited from RU considering sensitivity, and from NCR, SP2 and SMOTE given G-mean (these methods were also suggested in [25] as best ones for PART-U, however given sensitivity). Interestingly, our results did not reveal clear benefits of informed resampling over random one and in this sense they are consistent with [13]. However, we can hypothesize that informed sampling is essential for obtaining a more balanced classification accuracy (as captured by G-mean) where more careful modifications of learning data are necessary.

At the end we need to mention several limitations of our study:

1. The imbalance ratios examined in the experiment were typical for many real-life data sets, however, they were far from extreme imbalance that is becoming a relevant research topic [18]. Also the dimensionality was very small (we limited it to have a better control over and insight into generated sets), especially from the perspective of extreme dimensionality where problems with hundreds or event thousands of dimensions are considered [1, 30].

2. The data shapes were relatively regular and the variance within specific regions of the minority class was limited (especially in the case of the *paw3* shape). Moreover, the majority class was defined as a single integumental region. This could have made it less susceptible to RU, as demonstrated by G-mean. In real-life data sets the majority class often forms clusters (as demonstrated, e.g., in [25]), thus RU may have a more detrimental effect on the classification performance for this class.

3. We employed a set of well-established preprocessing methods. While many of them were considered in other studies (e.g., [8]), this set did not include new developments in this field, e.g., new variants of the SMOTE method [26].

We believe that despite the above limitations, our study may be interesting for the research community. As part of our future work we plan to expand the data generator with an ability to create data sets with both numerical and symbolic attributes and make it publicly available.

Acknowledgment

The authors would like to thank anonymous reviewers for their detailed comments and suggestions and to acknowledge support by the Polish National Science Center under Grant No. DEC-2013/11/B/ST6/00963.

References

[1] Bak, B. A., Jensen, J. L.: High dimensional classifiers in the imbalanced case, Computational Statistics and Data Analysis, 2016, 98, 46–59.

[2] Batista, G., Silva, D., Prati, R.: An experimental design to evaluate class imbalance treatment methods, in: Proc. of ICMLA'12 (Vol. 2), IEEE, 2012, 95-–101.

[3] Caruana, R., Karampatziakis, N., Yessenalina, A.: An empirical evaluation of supervised learning in high dimensions, in: Proc. of the 25th International Conference on Machine Learning (ICML 2008), 2008, 96–103.

[4] Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, 16, 2002, 341–378.

[5] Demšar, J. Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research, 7, 2006, 1–30.

[6] Dittman, D. J., Khoshgoftaar, T. M., Napolitano, A.: Selecting the appropriate data sampling approach for imbalanced and high-dimensional bioinformatics datasets. in: Proc. - IEEE 14th International Conference on Bioinformatics and Bioengineering (BIBE 2014), 2014, 304–310.

[7] Drummond C., Holte R., Severe class imbalance: Why better algorithms aren't the answer, in: Proc. of the 16th European Conference on Machine Learning (ECML 2005), Springer, 2005, 539–546.

[8] Fernández, A., López, V., Galar, M., Del Jesus, M. J., Herrera, F.: Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches, Knowledge-Based Systems, 2013, 42, 97–110.

[9] García V., Sánchez J., Mollineda R., An empirical study of the behavior of classifiers on imbalanced and overlapped data sets, in: Proc. of the 12th Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications, Springer, 2007, 397–406.

[10] García V., Sánchez J., Mollineda R., On the k-NN performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications, 11, 3-4, 2008, 269–280.

[11] García V., Sánchez J., Mollineda R., On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowledge-Based Systems, 23, 1, 2012, 13–21.

[12] He H., Ma Y., Imbalanced Learning: Foundations, Algorithms and Applications, Wiley, 2013.

[13] Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A.: Experimental perspectives on learning from imbalanced data, in: Proc. of the 24th International Conference on Machine Learning (ICML 2007), 2007, 17–23.

[14] Japkowicz N., Stephen S., The class imbalance problem: A systematic study, Intelligent Data Analysis 6, 5, 2002, 429–449.

[15] Japkowicz N., Class imbalance: Are we focusing on the right issue, in: Proc. of the 2nd Workshop on Learning from Imbalanced Data Sets, ICML 2003, 2003, 17–23.

[16] Jo T., Japkowicz N., Class imbalances versus small disjuncts, ACM Sigkdd Explorations Newsletter 6, 1, 2004, 40–49.

[17] Kang, P., Cho, S.: EUS SVMs: ensemble of under-sampled SVMs for data imbalance problems, in: Proc. of the 13th International Conference on Neural Information Processing (ICONIP). Springer, 2006, 837–846.

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

[19] Kubat M., Matwin S., Addressing the curse of imbalanced training sets: one-sided selection, in: Proc. of the 14th International Conference on Machine Learning (ICML 1997), 1997, 179–186.

[20] Laurikkala, J., Improving identification of difficult small classes by balancing class distribution, in: Proc. of the 8th Conference on Artificial Intelligence in Medicine (AIME 2001). LNCS 2101, Springer, 2001, 63–66.

[21] López, V., Fernández, A., García, S., Palade, V., Herrera, F., Empirical results and current trends on using data intrinsic characteristics: Empirical results and current trends on using data intrinsic characteristics, Information Sciences, 2013, 250, 113-–141.

[22] Maaranen H., Miettinen K., Mäkelä M.M., Quasi-random initial population for genetic algorithms, Computer and Mathematics with Applications, 47, 12, 1885–1895.

[23] Maciá, M., Bernadó-Mansilla, E., Orriols-Puig, Albert On the dimensions of data complexity through synthetic data sets in: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence. IOS Press, 2008, 244–252.

[24] Napierala K., Stefanowski J., Wilk S., Learning from imbalanced data in presence of noisy and borderline examples, in: Proc. of the 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2010). LNAI 6086, Springer, 2010, 158–167.

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

[26] Sáez J.A., Krawczyk B., Woźniak M., Analyzing the oversampling of different classes and types of examples in multi-class imbalanced data sets, Pattern Recognition, 57, 2016, 164–178.

[27] Staelin, C., Parameter selection for support vector machines, Technical Report HPL-2002-354 (R.1). HP Laboratories, Israel, 2003.

[28] Tang, Y., and Zhang, Y.-Q., Chawla, N., Krasser, S.: SVMs modeling for highly imbalanced classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 39, 1, 281–288.

[29] Tomašev, N., Mladenić, D., Class imbalance and the curse of minority hubs, Knowledge-Based Systems, 2013, 53, 157–172.

[30] Triguero, I., del Río, S., López, V., Bacardit, J., Benítez, J., Herrera, F.: ROSEFW-RF: The winner algorithm for the ECBDL'14 big data competition: An extremely imbalanced big data bioinformatics problem, Knowledge-Based Systems, 2014, 87, 69–79.

[31] Wah, Y. B., Abd Rahman, H. A., He, H., Bulgiba, A.: Handling imbalanced dataset using SVM and k-NN approach, in: AIP Conference Proceedings, 2016, 1750 (1), 020023.

[32] Wilk S., Stefanowski J., Wojciechowski S., Farion K., Michalowski W., Application of preprocessing methods to imbalanced clinical data: An experimental study, in: Proc. of the 5th International Conference on Information Technologies in Biomedicine (ITiB 2016), Vol. 1. Springer, 2016, 503–515.

[33] Xie, T., Yu, H., Wilamowski, B.: Comparison between traditional neural networks and radial basis function networks, in: 2011 IEEE International Symposium on Industrial Electronics. IEEE, 2011, 1194—1199.

Received 19.10.2016, Accepted 24.04.2017