DE GRUYTER
OPEN

Int. J. Appl. Math. Comput. Sci., 2017, Vol. 27, No. 4, 727–736
DOI: 10.1515/amcs-2017-0050

amcs

CCR: A COMBINED CLEANING AND RESAMPLING ALGORITHM FOR IMBALANCED DATA CLASSIFICATION

MICHAŁ KOZIARSKI [a], MICHAŁ WOŹNIAK [a,*]

[a] Department of Systems and Computer Networks
Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: {michal.koziarski,michal.wozniak}@pwr.edu.pl

Imbalanced data classification is one of the most widespread challenges in contemporary pattern recognition. Varying levels of imbalance may be observed in most real datasets, affecting the performance of classification algorithms. Particularly, high levels of imbalance make serious difficulties, often requiring the use of specially designed methods. In such cases the most important issue is often to properly detect minority examples, but at the same time the performance on the majority class cannot be neglected. In this paper we describe a novel resampling technique focused on proper detection of minority examples in a two-class imbalanced data task. The proposed method combines cleaning the decision border around minority objects with guided synthetic oversampling. Results of the conducted experimental study indicate that the proposed algorithm usually outperforms the conventional oversampling approaches, especially when the detection of minority examples is considered.

Keywords: machine learning, classification, imbalanced data, preprocessing, oversampling.

1. Introduction

The imbalanced data problem occurs whenever there is a significant disproportion among the number of instances in the classes considered. It is ubiquitous in contemporary machine learning and heavily influences many practical applications, including biological data analysis (Yu *et al.*, 2013; Hao *et al.*, 2014), medical diagnosis (Mazurowski *et al.*, 2008), neuroimaging (Dubey *et al.*, 2014), anomaly detection (Khreich *et al.*, 2010), face recognition (Liu and Chen, 2005), fraud detection (Wei *et al.*, 2013), and financing (Sanz *et al.*, 2015), to cite only a few. Due to its prevalence, imbalanced data have received a great amount of attention from the scientific community. Notably, in recent years the problem of imbalanced data has been considered in the context of big data (Triguero *et al.*, 2015), data streams (Hoens *et al.*, 2012) and multi-class classification (Fernández *et al.*, 2013). Nevertheless, many open problems still remain unsolved.

Various techniques of dealing with imbalanced data have been proposed, which may be grouped into the following categories:

1. preprocessing methods,
2. classification algorithms,
3. hybrid approaches.

Preprocessing methods focus on altering the original distributions in an attempt to reduce the imbalance ratio. Most often this is achieved by either over- or undersampling. More sophisticated approaches can also implicitly deal with other data difficulty factors, such as the presence of noise (Sáez *et al.*, 2013) or overlapping distributions.

Classification algorithms, designed for dealing with imbalanced data, are usually extensions of the existing learning methods, which aim at reducing the bias towards the majority class.

Finally, hybrid methods try to combine the two previous approaches to maximize their strengths and minimize their weaknesses.

In addition to the above categorization, methods of dealing with imbalance in data may be grouped on the basis of the priority they give to the detection of minority examples. Methods prioritizing high recall are desired in many practical applications, in which the cost associated with not detecting a minority object is especially high. While the existing approaches may be tuned to prioritize

[*] Corresponding author

recall, this significantly affects the precision in many cases. The motivation behind proposing a new approach was to design a method capable of achieving high recall, at the same time not completely overlooking precision. Therefore, we propose a novel, energy-based approach to clean the neighborhoods of minority examples, which we later combine with an oversampling procedure.

The main contributions of this work are as follows:

- proposition of a novel oversampling method called CCR (*combined cleaning and resampling*),
- exhaustive experimental evaluation of the proposed algorithm backed up by statistical tests.

2. Related works

Several excellent survey papers have been published in the area of imbalanced data learning. Sun *et al*. (2009) give a systematic overview of the imbalanced data problem and the methodological approaches to solve it. He and Garcia (2009) describe the state-of-the art methods in the field. López *et al*. (2013) provide a deep insight into the nature of imbalanced data. Galar *et al*. (2012) give a more specialized review of applying classifier ensembles to the problem, whereas Wang and Yao (2012) focus on the issue of multi-class imbalance. It is also worth mentioning the work by Zhang *et al*. (2016), who deal with efficient decomposition of the multi-class imbalance task using the one-versus-one scheme. Despite the amount of attention given by the scientific community, imbalanced data still pose many open problems, as discussed by Krawczyk (2016).

It is, however, important to note that imbalance in data usually does not pose a problem by itself. Only when combined with other data difficulty factors (Stefanowski, 2016) does it negatively affect recognition of the minority class. Defining and understanding such factors is therefore a crucial task when designing new methodologies for dealing with imbalanced data. Some research has been done in this area. Jo and Japkowicz (2004) analyze the issue of small disjuncts caused by data imbalance. The problem of class overlapping combined with imbalanced distributions is tackled by Prati *et al*. (2004) and García *et al*. (2007). Napierała *et al*. (2010) conduct an experimental study, measuring the impact of noisy and borderline examples on the imbalanced data learning task. Napierała and Stefanowski (2012) also propose a method of assessing safety of an example based on its local neighborhood.

Several resampling techniques have been proposed to combat the issue of imbalanced data and the related difficulty factors. By far the most prevalent is SMOTE (Chawla *et al*., 2002), based on generating synthetic examples instead of oversampling with replacement. It has been extensively studied and improved. SMOTEBoost (Chawla *et al*., 2003) combines the original SMOTE algorithm with a boosting procedure. The borderline-SMOTE (Han *et al*., 2005) family of methods focuses on oversampling unsafe minority examples. The safe-level-SMOTE (Bunkhumpornpat *et al*., 2009) takes the opposite approach and focuses on the safest objects. LN-SMOTE (Maciejewski and Stefanowski, 2011) exploits the local information about the neighborhoods of oversampled examples. MWMOTE (Barua *et al*., 2014) expands SMOTE by modifying the synthetic generation procedure. It creates new samples using a clustering approach. RWO-sampling (Zhang and Li, 2014) employs a random walk mechanism during the synthesis of new examples.

Ramentol *et al*. (2012) and Verbiest *et al*. (2014) take advantage of fuzzy rough set theory. Fernández-Navarro *et al*. (2011) propose an extension of SMOTE to the multi-class case. Finally, another important extension to the original SMOTE algorithm is the ADASYN (He *et al*., 2008) technique. It uses the idea of prioritizing the most difficult examples. It synthesizes larger proportion of new samples in the vicinity of a unsafe objects. It is also worth mentioning the SPIDER algorithm (Stefanowski and Wilk, 2008). It identifies the local characteristics of examples, and then removes those majority examples that may result in misclassifying examples from the minority class. It also uses local over-sampling of the objects from the minority class that are in a dense cloud of majority class objects.

While oversampling the minority class is the dominant approach, some work has been done in the area of undersampling the majority class. The neighborhood cleaning rule (Laurikkala, 2001) removes difficult examples based on their neighborhood. The EasyEnsemble and BalanceCascad (Liu *et al*., 2009) algorithms combine undersampling with classifier ensembles. García and Herrera (2009) propose to use undersampling together with the evolutionary algorithms. This idea is later expanded in the form of EUSBoost (Galar *et al*., 2013), in which boosting is additionally applied.

Finally, a family of methods combining oversampling and undersampling could be distinguished. Estabrooks *et al*. (2004) experimentally investigate the possibility of integrating these two approaches to resampling. Batista *et al*. (2004) propose to use SMOTE in combination with two data cleaning methods: Tomek links (Tomek, 1976) and the edited nearest-neighbor rule (Wilson, 1972). More recently, Bunkhumpornpat and Sinapiromsaran (2015) have proposed the CORE technique, in which oversampling is performed in combination with undersampling of the borderline examples.

3. CCR algorithm

To address the issue of data imbalance, we propose a novel algorithm for oversampling the minority class. We base it on two observations. Firstly, class imbalance does not make the classification problem difficult by itself. This might be easily illustrated by an example of a highly imbalanced, but linearly separable dataset. In such a case finding the decision border leading to the perfect accuracy will not be a problem for most classifiers. It is only when we deal with noisy data, complicated distributions or insufficient number of observations that imbalance further exacerbates the difficulty of the classification task. Secondly, in most problems achieving better accuracy on the minority class is the most pressing issue. Data imbalance mainly lowers classifiers' performance on examples from the minority class, leaving precision in a large part unaffected. At the same time, misclassification of minority examples is often more costly in practical applications such as medical diagnosis or fraud detection. Therefore, while we would like to achieve the highest possible accuracy for all classes, in practice sacrificing some precision to improve recall is often desirable.

Based on these observations, we propose a novel *combined cleaning and resampling* (CCR) algorithm. As the name indicates, it consists of two operations. Firstly, cleaning the neighborhoods of minority samples from majority objects. The aim of this step is to simplify the task of classification of examples from the minority class. Secondly, selectively generating synthetic samples, with the highest number of synthetic objects created near the least safe observations. In that we force the algorithm to focus on examples which are the most difficult to learn. A detailed pseudocode of the proposed algorithm is presented in Algorithm 1. The visualization of its behavior is presented in Fig. 1. In the remainder of this section we give a thorough description of both the cleaning and sample generating steps.

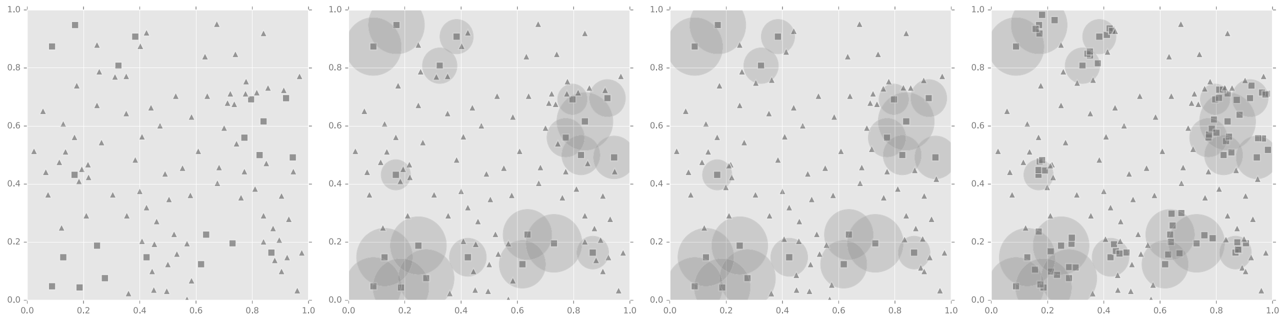
Algorithm 1. CCR algorithm.

```
function NoP($point$, $radius$):
$h \leftarrow$ number of majority points within $radius$ of $point$
return $h + 1$ {incremented to avoid division by zero}

function CCR($energy$):
for all minority points $m_i$ do
    $e_i \leftarrow energy$ {remaining energy budget}
    $r_i \leftarrow 0$
    while $e_i > 0$ do
        $\Delta r \leftarrow \dfrac{e_i}{\text{NoP}(m_i, r_i)}$
        if NoP($m_i$, $r_i + \Delta r$) $>$ NoP($m_i$, $r_i$) then
            $\Delta r \leftarrow$ dist. to the nearest majority point not within $r_i$
        end if
        $r_i \leftarrow r_i + \Delta r$
        $e_i \leftarrow e_i$ - $\Delta r \cdot$ NoP($m_i$, $r_i$)
    end while
    for all majority points $M_j$ within $r_i$ of $m_i$ do
        $d \leftarrow \|M_j - m_i\|_1$
        $t_j \leftarrow t_j + \dfrac{r_i - d}{d} \cdot (M_j - m_i)$ {translation of $M_j$}
    end for
end for
apply accumulated translations to all majority points
$G \leftarrow |M| - |m|$ {no. of synthetic samples to be generated}
for all minority points $m_i$ do
    $g_i \leftarrow \dfrac{r_i^{-1}}{\sum_k r_k^{-1}} \cdot G$ {proportion of $G$ for $m_i$}
    for $g_i$ times do
        $p \leftarrow$ random point inside a sphere with radius $r_i$
        generate synthetic point $m_i + p$
    end for
end for
```

3.1. Cleaning the minority samples neighborhood.

The problem noise present in data is especially difficult in the case of imbalanced distributions. Distortions can significantly deteriorate classifiers' performance, especially on examples from the minority class (called later minority examples, objects or points) (Van Hulse *et al.*, 2007). We propose a method of overcoming this issue by cleaning a minority object neighborhood out of the examples from the majority class (called later majority examples, objects or points). Intuitively, what we try to achieve is to expand the decision borders in favor of minority examples. By doing so, we reduce the impact of noise in the majority examples on the minority class detection. At the same time, accounting for minority outliers is necessary, at least to some extent. To satisfy these conflicting requirements, we propose an energy-based method of neighborhood cleaning. The visualization of the approach is presented in Fig. 2. Every minority example has an associated energy budget, defined as a parameter of the algorithm. With every minority object there is also associated a sphere, a region that will be later cleared of majority objects. Starting from the minority point, we try to expand the radius of the sphere, expending the available energy. However, every majority example we reach increases the cost of growing the sphere linearly, blocking the expansion process. We introduce this limitation to decrease the impact of minority outliers. In the case of the minority samples surrounded by a large number of majority objects,



(a) (b) (c) (d)

Fig. 1. Visualized steps of the CCR algorithm: original imbalanced data (a), the sphere radius calculated for every minority point (b), the occurrence of majority points within the radius limits its final length—majority points within the radius pushed out of the spheres (c), generated synthetic minority samples (d). The number of generated samples near the minority point is inversely proportional to its radius.

the resulting spheres will be small. This corresponds to lower confidence we have about proper labeling of the object considered. Finally, the original majority examples are pushed out of the spheres.

A modification of the original data is a possibly dangerous operation, because some information might be lost. Most conventional resampling techniques do not do it explicitly. However, we would like to argue that extensive oversampling of the minority class may lead to a similar conclusion.

We chose the spherical shape of the clean-out region due to the computational simplicity associated with it. It should, however be noted that in many cases it may be suboptimal, for instance, if majority objects are grouped on one side of the minority example. It would be preferable to limit the growth of the region on this side, expanding it much further on the other.

Two objections should be stated with regard to algorithm implementation. First of all, in some cases majority points could be affected by a number of spheres simultaneously. Several approaches to this problem may be employed, e.g., sphere expansion could be conducted interactively, taking into account previous translations of majority points. This could, however, lead to pushing majority examples right into the neighborhood of the minority points already considered. Alternatively, a more drastic approach to such majority points could be taken, in which they would be deleted. We opted for a strategy in which the translations are accumulated on an unchanging distribution and later applied all at once. In this paradigm it is possible for a cluster of minority points to push out a majority point with combined, possibly unwarranted energy. Furthermore, majority points can still be pushed out into the spheres associated with neighboring minority objects. While this approach does pose its own issues, we decided that they were least severe.

Secondly, we are faced with the choice of a distance measure used to calculate the cleaning regions. The influence of the different distance metrics applied to high-dimensional data has been well studied. It has been shown that the Manhattan distance is usually preferable to the Euclidean distance when operating on data with a large number of dimensions (Aggarwal *et al.*, 2001). We therefore used the L_1 norm as a distance metric in the implementation of the algorithm.

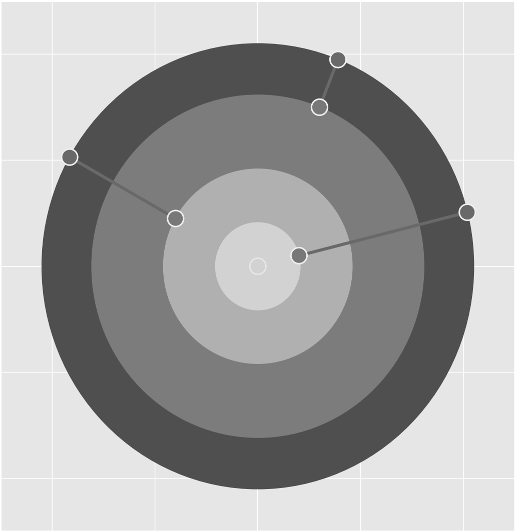


Fig. 2. Visualized sphere radius calculation. Every example from the minority class (inside the sphere) has obtained energy budget. Starting from the minority point, the sphere radius is expanded, decreasing available energy. Upon reaching a majority example, the energy cost of the expansion is increased. The consecutive orbits, depicted with an increasing color intensity, reflect higher expansion cost. Finally, the available energy is depleted and the expansion stops. The original majority examples are pushed out of the sphere.

3.2. Selectively generating synthetic samples. In the second step of the algorithm we perform oversampling to artificially balance the majority and the minority class. We rely heavily on the spheres produced during the cleaning process. For every sample, a synthetic sample is generated randomly within the area of the sphere. The proportion of samples generated for every minority example is reversely proportional to its associated radius. Let r_i be the radius of the i-th minority point, N the number of minority points and G the total number of synthetic minority points we wish to generate. We define the number of synthetic points generated around the i-th point as

$$g_i = \frac{r_i^{-1}}{\sum_{k=1}^{N} r_k^{-1}} G. \tag{1}$$

Thereby we force the classification algorithm to focus on the most difficult examples. Since minority points with the smallest spheres associated to them are the ones surrounded by most majority examples, their correct classification would be normally most difficult. At the same time, however, since spheres associated with them are relatively small, the synthetic samples are generated in a close proximity. This forces the classifier to constrain the area assigned to the minority class in such regions, reducing the negative impact of minority outliers.

4. Experimental study

To evaluate the performance of the proposed CCR algorithm, we conducted an experimental study divided into two stages. In the first one, preliminary analysis, we measured the impact of the CCR energy parameter on the performance of the algorithm. The goal of this part of the study was establishing what value of the energy parameter, if any, is optimal for the algorithm. In the second one, final analysis, we compared the proposed method with the state-of-the-art techniques based on synthetic oversampling and performed statistical analysis of the results. In this section we describe the set-up of the conducted experiments and discuss the achieved outcomes.

4.1. Datasets. Evaluation was performed on 42 datasets taken from the KEEL (Alcalá *et al.*, 2010) imbalanced data repository. The datasets were randomly divided into two partitions: the first one, consisting of 10 datasets, was dedicated to the preliminary analysis, whereas the second one, consisting of 32 datasets, was used during the final analysis. Both partitions contained diverse datasets, with diversification measured by parameters such as the the imbalance ratio, the number of features and the number of samples. Details of the datasets are presented in Tables 1 and 2 for the preliminary and the final partition, respectively. Only two-class datasets composed solely of numerical data were used.

Table 1. Details of datasets used during preliminary evaluation.

No.	Name	IR	Features	Samples
1	pima	1.87	8	768
2	yeast1	2.46	8	1484
3	haberman	2.78	3	306
4	vehicle2	2.88	18	846
5	led7digit02456789vs1	10.97	7	443
6	yeast1vs7	14.30	7	459
7	winequalityred4	29.17	11	1599
8	poker9vs7	29.50	10	244
9	abalone3vs11	32.47	8	502
10	winequalitywhite9vs4	32.60	11	168

4.2. Implementation and reproducibility. Experiments were implemented in the Python programming language. The code is publicly available at `https://github.com/michalkoziarski/CCR`. Additionally, cross-validation folds used throughout the experimental study were supplied together with the code. Whenever possible, existing implementations of the algorithms were used to limit the risk of programming errors. Notably, classification algorithms provided in the scikit-learn (Pedregosa *et al.*, 2011) library were used, as well as data resampling methods provided in the imbalanced-learn (Lemaitre *et al.*, 2017) library.

4.3. Preliminary analysis. During the first stage of the experimental study, preliminary analysis, we evaluated the impact of the CCR energy parameter on its performance. To this end, we measured the values of AUC, G-mean and F-measure for 10 distinct datasets while adjusting the value of energy, chosen from $\{0.001, 0.0025, 0.005, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1.0, 2.5, 5.0, 10.0, 25.0, 50.0, 100.0\}$. The datasets were partitioned into folds and the 5×2 cross-validation procedure was employed, with the average values of the metrics being reported. During this stage of the experiment a single classifier, the CART decision tree, was used. The achieved results are displayed in Fig. 3. Based on the observed results, we conclude that there is no single value of the energy parameter optimal for all the tested datasets. For Datasets 1–5 and 9, performance was relatively stable for lower values of energy, whereas the choice of a higher value led to a decrease in performance. In contrast, the behavior was less stable for Datasets 6–8 and 10, for which setting the higher value of energy led to better performance, especially when AUC and G-mean were considered. Despite the fact that higher values of energy were preferred for 4 out of 5 datasets with a higher imbalance ratio, fine-tuning of the parameters was still necessary to achieve optimal performance.

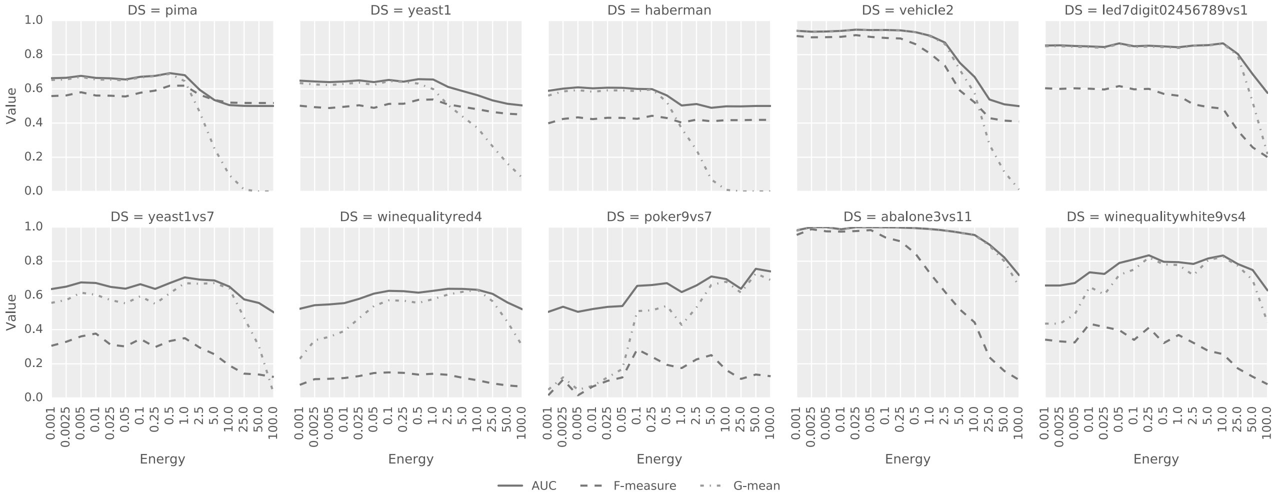


Fig. 3. Results of preliminary analysis: the impact of the energy parameter on AUC, G-mean and F-measure is evaluated for 10 datasets (DS); a CART decision tree was used as a classification algorithm.

4.4. Final analysis.

In the second stage of the experimental study, final analysis, the CCR algorithm proposed in this paper was compared with the state-of-the-art resampling techniques. To provide a wide range of relevant methods, we considered the following reference algorithms: SMOTE and ADASYN as general, widely used methods of dealing with imbalance in data; borderline-SMOTE (Bord) as an example of a method designed specifically to deal with borderline minority examples; SMOTE in combination with Tomek links (SMOTE+TL) and the edited nearest-neighbor rule (SMOTE+ENN) as an example of methods combining oversampling with the cleaning of difficult examples; and the neighborhood cleaning rule (NCL), an undersampling technique focused on cleaning the neighborhood of difficult minority objects. Additionally, as a baseline we evaluated the case in which no form of resampling was applied to the data (Base). Four different classifiers were considered: a CART decision tree, k-nearest neighbors (k-NN), a support vector machine (SVM) with linear kernel and naive Bayes (NB). Additionally, we evaluated the case in which bagging was used with CART (CART+Bag) and k-NN (k-NN+Bag).

Based on the results of the preliminary analysis, we decided to choose the value of the energy parameter separately for each dataset. To this end, we performed a 5-fold cross-validation on the training data. Similarly to the preliminary analysis, the values of energy from $\{0.001, 0.0025, 0.005, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1.0, 2.5, 5.0, 10.0, 25.0, 50.0, 100.0\}$ were considered. The value of energy for which the average AUC was maximized was selected and used for final resampling on the full training set. For the remaining resampling algorithms, as well as all of the classification algorithms, the default values of the parameters contained in the corresponding libraries were used. In all the cases oversampling was applied up to the point of achieving balanced distributions. Prior to classification, data were normalized to the range from 0 to 1. No further preprocessing was applied. 5×2 cross-validation was used in all the tests.

The results of the conducted experimental study are presented in Table 3. It contains average rankings for various classification algorithms and performance measures, obtained by applying the Friedman procedure, as well as the results of Shaffer's post-hoc procedure. Additionally, complete results of this part of the experimental study, that is, tables containing the precise values of the evaluation measures as well as the results of the conducted statistical analysis, are provided as supplementary material at `https://github.com/michalkoziarski/CCR`. Since the choice of the performance measure is ambiguous when dealing with imbalanced data, all of the most common metrics were used, namely accuracy, precision, recall, F-measure, G-mean and AUC. The resampling algorithm proposed in this paper, CCR, scored the highest average ranking in recall for all of the classifiers except NB. When combined with CART and CART+Bag, the highest average rank was achieved in G-mean and AUC as well, with significantly better results than most of the reference methods. For AUC, the highest average rank was observed also for k-NN+Bag. In general, using the CCR algorithm resulted in achieving high recall at the cost of precision for all of the classifiers except NB. When combined performance metrics were considered, this led to a better G-mean and AUC at the cost of a worse F-measure. Interestingly, the trend of

Table 2. Details of datasets used during the final evaluation.

No.	Name	IR	Features	Samples
1	glass1	1.82	9	214
2	ecoli0vs1	1.86	7	220
3	wisconsin	1.86	9	683
4	glass0	2.06	9	214
5	vehicle1	2.90	18	846
6	vehicle3	2.99	18	846
7	glass0123vs456	3.20	9	214
8	vehicle0	3.25	18	846
9	ecoli1	3.36	7	336
10	newthyroid1	5.14	5	215
11	ecoli2	5.46	7	336
12	segment0	6.02	19	2308
13	glass6	6.38	9	214
14	yeast3	8.10	8	1484
15	ecoli3	8.60	7	336
16	pageblocks0	8.79	10	5472
17	yeast2vs4	9.08	8	514
18	yeast05679vs4	9.35	8	528
19	vowel0	9.98	13	988
20	glass016vs2	10.29	9	192
21	glass2	11.59	9	214
22	ecoli4	15.80	7	336
23	pageblocks13vs4	15.86	10	472
24	abalone918	16.40	8	731
25	yeast1458vs7	22.10	8	693
26	yeast2vs8	23.10	8	482
27	yeast4	28.10	8	1484
28	yeast1289vs7	30.57	8	947
29	yeast5	32.73	8	1484
30	yeast6	41.40	8	1484
31	poker89vs6	58.40	10	1485
32	abalone19	129.44	8	4174

achieving higher recall at the price of lower precision was reversed for the NB classifier. In this case, CCR achieved the highest average rank in precision, as well as all three of the combined performance metrics, at the same time having worse recall.

One of the most important questions we have to ask when dealing with imbalanced data is what performance measure should we optimize for. This, of course, depends heavily on the specific problem domain. We would argue, however, that correct detection of minority examples is often the most pressing issue, especially when dealing with extreme levels of imbalance. The results of the conducted experimental study seem to indicate that the proposed CCR method is particularly well suited for such a task. At the same time it would be trivial to construct an algorithm achieving perfect recall with no regard for precision. In the conducted study, the CCR algorithm proved to strike a right balance between the two.

5. Conclusions and future work

We presented a novel oversampling technique, the CCR algorithm, designed to deal with the imbalanced data classification task, which employs two core ideas. Firstly, to clear the decision border by pushing away majority examples located too closely to minority ones. Secondly, to oversample selectively, with a higher number of synthetic data points generated around unsafe samples. During the experimental evaluation we empirically proved that the proposed algorithm is well suited for tackling the imbalanced data problem. The CCR algorithm achieved best performance in combination with the CART decision tree. Additionally, it scored the best recall for the majority of the tested classifiers. In most cases high recall was, however, accompanied by precision lower than that of the reference methods. This trade-off turned out to be beneficial with regard to the value of the combined metrics, since in several cases CCR also achieved the best F-measure, G-mean and AUC.

Despite its good performance on the benchmarks considered, the CCR algorithm in its current form has some limitations. It was not designed to deal with categorical data. In the conducted experimental study, no such datasets were considered and, therefore, the algorithm's performance in such cases remains unknown. It is possible that to properly deal with categorical datasets the algorithm would have to be modified accordingly. Secondly, since the method is distance-based, it performs best when the features take values in similar ranges. This, however, is easily mitigated by proper preprocessing. Finally, in the proposed form the algorithm is suitable only for a two-class classification problem. To be usable in a multi-class task, the problem has to be decomposed into several binary tasks.

Furthermore, in this paper we did not focus on the computational complexity of the presented algorithm. The execution time of CCR on the benchmarks considered was comparable with that of the reference methods. However, an more thorough analysis would be required to assess the algorithm's behavior on larger datasets. To make learning feasible on such data, adjustments to the algorithm would be required. For instance, in the presented form the algorithm can be easily parallelized.

Finally, several simplifications were made in the proposed implementation of the algorithm that do not full capture its intended behavior. Using spheres as the regions out of which majority samples are pushed out is computationally inexpensive. However, it does not take into account the exact position of the said samples within the sphere. More sophisticated shapes might be required to accurately capture the nature of complicated distributions. Similarly, generating synthetic samples randomly within the sphere is a naive approach. Using more sophisticated regions for sampling could potentially

Table 3. Average rankings obtained by applying the Friedman procedure: the highest average ranking for every classifier (Cl.) and metric combination in boldface. Shaffer's post-hoc procedure was used to determine the statistical significance of the results. Methods that achieved significantly different results (with $p = 0.05$) than CCR are denoted in subscript, with the minus sign for methods that achieved better results and the plus sign for those that achieved worse results.

Cl.	Metric	Base	SMOTE	ADASYN	Bord	SMOTE+TL	SMOTE+ENN	NCL	CCR
CART	Accuracy	**3.0156 –**	4.8438 –	5.2031	3.6250 –	4.5156 –	3.5469 –	4.3750 –	6.8750
	Precision	3.0938 –	4.5312 –	5.5625	3.7500 –	4.4375 –	**2.9375 –**	5.0000	6.6875
	Recall	7.1875 +	4.8750 +	2.8906	5.1094 +	4.6719 +	5.5156 +	3.8594 +	**1.8906**
	F-measure	4.9688	4.7188	4.1875	4.4062	4.4688	4.5000	**3.4062 –**	5.3438
	G-mean	6.8750 +	4.8125 +	3.4062	5.0938 +	4.5000 +	5.3125 +	3.8125	**2.1875**
	AUC	6.6875 +	5.0469 +	3.4062	4.9844 +	4.5625 +	5.0938 +	3.7500	**2.4688**
CART+Bag	Accuracy	**2.7344 –**	4.4062 –	5.1406	4.1875 –	4.3125 –	4.5000 –	3.8438 –	6.8750
	Precision	**2.2500 –**	4.7344 –	5.6250	4.2812 –	4.1875 –	3.7031 –	4.3125 –	6.9062
	Recall	7.3281 +	4.4844 +	2.7969	5.2656 +	4.1094 +	5.3750 +	4.9844 +	**1.6562**
	F-measure	5.4062	4.2031	3.9688	4.4062	**3.8750**	4.6406	4.2500	5.2500
	G-mean	6.9375 +	4.3281 +	2.9688	5.2500 +	3.8750	5.3906 +	5.0625 +	**2.1875**
	AUC	6.8438 +	4.4375 +	3.0000	5.1094 +	3.9375	5.3906 +	4.9375 +	**2.3438**
k-NN	Accuracy	**1.9062 –**	4.9688 –	5.2812	4.3750 –	4.9844 –	4.5000 –	3.0312 –	6.9531
	Precision	**1.9688 –**	5.0625	5.5938	4.7188 –	4.8438 –	4.1875 –	2.7500 –	6.8750
	Recall	7.9062 +	3.2969	3.6562	4.0000	3.5938	4.5156 +	6.5469 +	**2.4844**
	F-measure	4.8594	4.0625	5.2812	**3.5625 –**	4.0000	4.2500	4.1719	5.8125
	G-mean	7.4531 +	**3.0625**	4.4375	3.8125	3.1562	4.0938	6.4219 +	3.5625
	AUC	7.4375 +	**3.2500**	4.3750	3.4375	3.4219	4.0156	6.3750 +	3.6875
k-NN+Bag	Accuracy	**1.9219 –**	5.5312	5.4375	4.7500	4.9375	3.9062 –	3.1094 –	6.4062
	Precision	**1.9219 –**	5.5312	5.8438	5.0312	4.8125	3.8125 –	2.7031 –	6.3438
	Recall	7.7344 +	3.2969	3.6719	3.8750	3.4219	4.4844	6.7188 +	**2.7969**
	F-measure	5.0469	4.2188	5.3125	3.8438	3.7812	**3.7500**	4.5781	5.4688
	G-mean	7.3281 +	3.4375	4.2500	3.9062	**3.2188**	3.7500	6.7031 +	3.4062
	AUC	7.3438 +	3.5938	4.2500	3.7344	3.4531	3.6719	6.6875 +	**3.2656**
SVM	Accuracy	**2.2969 –**	4.6875 –	5.9531	5.1875 –	4.6875 –	3.6562 –	2.4062 –	7.1250
	Precision	**2.8594 –**	4.2344 –	5.7188	5.3125	4.4531 –	3.1562 –	3.4531 –	6.8125
	Recall	7.8906 +	4.0938 +	2.4531	3.7344	4.2344 +	4.8594 +	6.7031 +	**2.0312**
	F-measure	6.0625	**3.4844 –**	4.5312	4.2188	3.8281	3.5625	4.8438	5.4688
	G-mean	7.6875 +	**3.2969**	3.3125	4.0312	3.4531	3.9062	6.5000 +	3.8125
	AUC	7.4062 +	**3.3594**	3.5625	3.9062	3.6719	3.9062	6.2812 +	3.9062
NB	Accuracy	4.1562	4.0469	7.3281 +	4.5156	4.5625	**3.2812**	4.4688	3.6406
	Precision	3.8750	3.8750	6.5000 +	5.1562 +	4.7344	3.6719	4.5938	**3.0625**
	Recall	5.0625	4.5781	**2.5312 –**	4.4844	4.4531	4.7344	4.7031	5.4531
	F-measure	4.7812	4.3438	6.0000 +	4.5625	4.7188	3.9688	4.5938	**3.0312**
	G-mean	5.1875	4.0000	6.0938 +	4.4688	4.2500	3.6875	4.9375	**3.3750**
	AUC	4.9375	4.2500	5.5000 +	4.9062	4.6250	4.0625	4.6875	**3.0312**

be beneficial, especially when considering the precision of the algorithm. Alternatively, a guided sampling strategy could be employed, in which information about the local neighborhood would be used.

Acknowledgment

This work was supported by the Polish National Science Center under the grant no. UMO-2015/19/B/ST6/01597 and the PLGrid Infrastructure, as well as the statutory fund of the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Science and Technology.

References

Aggarwal, C.C., Hinneburg, A. and Keim, D.A. (2001). On the surprising behavior of distance metrics in high dimensional space, *International Conference on Database Theory, London, UK*, pp. 420–434.

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L. and Herrera, F. (2010). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework, *Journal of Multiple-Valued Logic and Soft Computing* **17**(2–3): 255–287.

Barua, S., Islam, M.M., Yao, X. and Murase, K. (2014). MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning, *IEEE Transactions on Knowledge and Data Engineering* **26**(2): 405–425.

Batista, G.E., Prati, R.C. and Monard, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data, *ACM SIGKDD Explorations Newsletter* **6**(1): 20–29.

Bunkhumpornpat, C. and Sinapiromsaran, K. (2015). CORE: Core-based synthetic minority over-sampling and borderline majority under-sampling technique, *Inter-*

national Journal of Data Mining and Bioinformatics **12**(1): 44–58.

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. (2009). Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, *Pacific-Asia Conference on Knowledge Discovery and Data Mining, Bangkok, Thailand*, pp. 475–482.

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

Chawla, N.V., Lazarevic, A., Hall, L.O. and Bowyer, K.W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting, *European Conference on Principles of Data Mining and Knowledge Discovery, Cavtat/Dubrovnik, Croatia*, pp. 107–119.

Dubey, R., Zhou, J., Wang, Y., Thompson, P.M. and Ye, J. (2014). Analysis of sampling techniques for imbalanced data: An $n = 648$ ADNI study, *NeuroImage* **87**: 220–241.

Estabrooks, A., Jo, T. and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets, *Computational Intelligence* **20**(1): 18–36.

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

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

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, *IEEE Transactions on Systems, Man, and Cybernetics C: Applications and Reviews* **42**(4): 463–484.

Galar, M., Fernández, A., Barrenechea, E. and Herrera, F. (2013). EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, *Pattern Recognition* **46**(12): 3460–3471.

García, S. and Herrera, F. (2009). Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy, *Evolutionary Computation* **17**(3): 275–306.

García, V., Sánchez, J. and Mollineda, R. (2007). An empirical study of the behavior of classifiers on imbalanced and overlapped data sets, *Iberoamerican Congress on Pattern Recognition, Valparaiso, Chile*, pp. 397–406.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, *International Conference on Intelligent Computing, Hefei, China*, pp. 878–887.

Hao, M., Wang, Y. and Bryant, S.H. (2014). An efficient algorithm coupled with synthetic minority over-sampling technique to classify imbalanced PubChem BioAssay data, *Analytica Chimica Acta* **806**: 117–127.

He, H., Bai, Y., Garcia, E.A. and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning, *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China*, pp. 1322–1328.

He, H. and Garcia, E.A. (2009). Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering* **21**(9): 1263–1284.

Hoens, T.R., Polikar, R. and Chawla, N.V. (2012). Learning from streaming data with concept drift and imbalance: An overview, *Progress in Artificial Intelligence* **1**(1): 89–101.

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

Khreich, W., Granger, E., Miri, A. and Sabourin, R. (2010). Iterative Boolean combination of classifiers in the ROC space: An application to anomaly detection with HMMs, *Pattern Recognition* **43**(8): 2732–2752.

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

Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution, *Conference on Artificial Intelligence in Medicine in Europe, Cascais, Portugal*, pp. 63–66.

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

Liu, X.-Y., Wu, J. and Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning, *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics* **39**(2): 539–550.

Liu, Y.-H. and Chen, Y.-T. (2005). Total margin based adaptive fuzzy support vector machines for multiview face recognition, *2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA,* Vol. 2, pp. 1704–1711.

López, V., Fernández, A., García, S., Palade, V. and Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, *Information Sciences* **250**: 113–141.

Maciejewski, T. and Stefanowski, J. (2011). Local neighbourhood extension of SMOTE for mining imbalanced data, *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Paris, France,* pp. 104–111.

Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A. and Tourassi, G.D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, *Neural Networks* **21**(2): 427–436.

Napierała, K. and Stefanowski, J. (2012). Identification of different types of minority class examples in imbalanced data, *International Conference on Hybrid Artificial Intelligence Systems, Salamanca, Spain*, pp. 139–150.

Napierała, K., Stefanowski, J. and Wilk, S. (2010). Learning from imbalanced data in presence of noisy and borderline examples, *International Conference on Rough Sets and Current Trends in Computing, Warsaw, Poland*, pp. 158–167.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**(Oct): 2825–2830.

Prati, R.C., Batista, G. and Monard, M.C. (2004). Class imbalances versus class overlapping: An analysis of a learning system behavior, *Mexican International Conference on Artificial Intelligence, Mexico City, Mexico*, pp. 312–321.

Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Cornelis, C. and Herrera, F. (2012). SMOTE-FRST: A new resampling method using fuzzy rough set theory, *10th International FLINS Conference on Uncertainty Modelling in Knowledge Engineering and Decision Making, Istanbul, Turkey*.

Sáez, J. A., Galar, M., Luengo, J. and Herrera, F. (2013). Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness, *Information Sciences* **247**: 1–20.

Sanz, J.A., Bernardo, D., Herrera, F., Bustince, H. and Hagras, H. (2015). A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data, *IEEE Transactions on Fuzzy Systems* **23**(4): 973–990.

Stefanowski, J. (2016). Dealing with data difficulty factors while learning from imbalanced data, *in* S. Matwin and J. Mielniczuk (Eds.), *Challenges in Computational Statistics and Data Mining*, Springer, Heilderberg, pp. 333–363.

Stefanowski, J. and Wilk, S. (2008). Selective pre-processing of imbalanced data for improving classification performance, *International Conference on Data Warehousing and Knowledge Discovery, Turin, Italy*, pp. 283–292.

Sun, Y., Wong, A.K. and Kamel, M.S. (2009). Classification of imbalanced data: A review, *International Journal of Pattern Recognition and Artificial Intelligence* **23**(04): 687–719.

Tomek, I. (1976). Two modifications of CNN, *IEEE Transactions on Systems, Man, and Cybernetics* **6**(11): 769–772.

Triguero, I., del Río, S., López, V., Bacardit, J., Benítez, J.M. and Herrera, F. (2015). ROSEFW-RF: The winner algorithm for the ECBDL14 big data competition. An extremely imbalanced big data bioinformatics problem, *Knowledge-Based Systems* **87**: 69–79.

Van Hulse, J., Khoshgoftaar, T.M. and Napolitano, A. (2007). Skewed class distributions and mislabeled examples, *7th IEEE International Conference on Data Mining Workshops (ICDMW 2007), Omaha, NE, USA*, pp. 477–482.

Verbiest, N., Ramentol, E., Cornelis, C. and Herrera, F. (2014). Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection, *Applied Soft Computing* **22**: 511–517.

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

Wei, W., Li, J., Cao, L., Ou, Y. and Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data, *World Wide Web* **16**(4): 449–475.

Wilson, D.L. (1972). Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man, and Cybernetics* **2**(3): 408–421.

Yu, H., Ni, J. and Zhao, J. (2013). ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, *Neurocomputing* **101**: 309–318.

Zhang, H. and Li, M. (2014). RWO-sampling: A random walk over-sampling approach to imbalanced data classification, *Information Fusion* **20**: 99–116.

Zhang, Z., Krawczyk, B., García, S., Rosales-Pérez, A. and Herrera, F. (2016). Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data, *Knowledge-Based Systems* **106**: 251–263.

Michał Koziarski received his MSc degree in computer science from the Wrocław University of Science and Technology, Poland, in 2016. Currently, he is a PhD student at the Department of Systems and Computer Networks of the same university. His research interests include computer vision, neural networks and imbalanced data classification.

Michał Woźniak is a professor of computer science at the Department of Systems and Computer Networks, Wrocław University of Science and Technology, Poland. He received his MSc degree in biomedical engineering from the Wrocław University of Technology in 1992, and his PhD and DSc (habilitation) degrees in computer science in 1996 and 2007, respectively, from the same university. In 2015 he was granted a professorial title by the President of Poland. His research focuses on machine learning, compound classification methods, classifier ensemble, data stream mining, imbalance data processing. Prof. Woźniak has been involved in research projects related to the above-mentioned topics, and has been a consultant in several commercial projects for well-known Polish companies and public administration. He has published over 260 papers and three monographs. He is a member of the editorial board of high ranked journals, such as *Information Fusion*, *Applied Soft Computing*, *Engineering Applications of Artificial Intelligence*.

Received: 12 January 2016
Revised: 24 June 2017
Re-revised: 22 August 2017
Accepted: 25 August 2017