



A Cluster-Based Under-Sampling Algorithm for Class-Imbalanced Data

A. Guzmán-Ponce^{1,2(✉)}, R. M. Valdovinos¹, and J. S. Sánchez² 

¹ Facultad de Ingeniería, Universidad Autónoma del Estado de México,
Toluca, Mexico

aguzmanp643@alumno.uaemex.mx, rvaldovinosr@uaemex.mx

² Institute of New Imaging Technologies,
Department of Computer Languages and Systems,
Universitat Jaume I, Castelló de la Plana, Spain
sanchez@uji.es

Abstract. The resampling methods are among the most popular strategies to face the class imbalance problem. The objective of these methods is to compensate the imbalanced class distribution by over-sampling the minority class and/or under-sampling the majority class. In this paper, a new under-sampling method based on the DBSCAN clustering algorithm is introduced. The main idea is to remove the majority class instances that are identified as noise by DBSCAN. The proposed method is empirically compared to well-known state-of-the-art under-sampling algorithms over 25 benchmarking databases and the experimental results demonstrate the effectiveness of the new method in terms of sensitivity, specificity, and geometric mean of individual accuracies.

Keywords: Class imbalance · DBSCAN · Under-sampling · Noise filtering

1 Introduction

The class imbalance problem has traditionally been viewed as a challenging scenario in Data Mining and Machine Learning [13]. A two-class data set is said to be imbalanced when one of the classes is strongly under-represented (minority or positive class) in comparison with the other class (majority or negative class) [12]. Several studies have shown that the class-imbalanced data problem may give rise to negative effects on the performance of most standard classifiers because they are often strongly biased towards the classification of the majority class and results in poor performance on the minority class [7], which is paradoxically the one of most interest in many real-world applications.

Much research has been conducted to tackle the class imbalance problem by developing different techniques at both algorithmic and data levels [11–13]. The methods at the algorithmic level modify the existing learning models for biasing the discrimination process towards the minority class. On the other hand,

the data level solutions consist of a preprocessing step to resample the original data set, either by over-sampling the minority class and/or under-sampling the majority class, until the classes are approximately equally represented. In general, the resampling strategies have been the most widely used techniques because they are independent of the classifier and can be easily implemented for any problem [1,9].

The simplest resampling methods are random over-sampling and random under-sampling (RUS). The former balances the class distribution through the random replication of minority class instances; although this method appears to be quite effective without adding new information into the data set, it may increase the likelihood of overfitting because it makes exact copies of the minority class examples. On the other hand, random under-sampling aims at balancing the data set through the random removal of instances that belong to the over-sized class; despite important information can be lost when instances are discarded at random, it has empirically been shown to be one of the most effective resampling methods.

Despite the apparent superiority of over-sampling [8,19], this paper concentrates on under-sampling because it seems to be a better choice for problems represented by large-scale databases [23], which is a situation of great interest with the advent of Big Data. Despite under-sampling may result in throwing away potentially useful information about the majority class and perturbing the a priori probability of the data set, it has to be taken in mind that over-sampling worsens the computational burden of most classifiers, may increase the likelihood of overfitting and introduce noise that could result in a loss of performance [3]. Taking all these questions into account, we introduce an under-sampling method based on the DBSCAN clustering algorithm [6], which will be applied only to the instances of the majority class. The proposed method is then empirically compared to other well-known under-sampling techniques over a collection of imbalanced two-class data sets taken from the KEEL repository. From a practical point of view, the major contributions of this paper are: (1) an adaptation of the DBSCAN algorithm to obtain a compact and representative cluster of the majority class, and (2) a new cluster-based under-sampling method to face the class imbalance problem.

Several cluster-based approaches have already been proposed to reduce the size of the majority class. For instance, Yen and Lee proposed the use of the K -means clustering technique to downsizing the majority class [26]. Lin et al. introduced an under-sampling strategy also based on K -means, which sets a number of clusters equal to the number of minority class instances [16]. Inspired by the approach proposed by Yen and Lee, Ofek et al. developed a new under-sampling algorithm that clusters only the minority class instances [17]. Tsai et al. introduced a novel under-sampling technique that combines clustering analysis (using the affinity propagation algorithm) and instance selection (the authors use three different algorithms) over the majority class [23]. Verma et al. proposed a hybrid approach based on the DBSCAN algorithm that integrates over-sampling

and under-sampling [24]. Onan under-samples the majority class instances by employing a consensus cluster-based scheme [18].

Hereafter the paper is organized as follows. Section 2 provides a brief review of the under-sampling methods used in this study. Next, the new algorithm based on DBSCAN clustering is introduced in Sect. 3. Section 4 describes the experimental set-up, whereas Sect. 5 discusses the experimental results. Finally, Sect. 6 remarks the main conclusions and outlines some directions for further research.

2 Under-Sampling Methods

To remedy the shortcomings associated with the lazy RUS strategy, many other under-sampling methods have been developed with the aim of incorporating a more intelligent selection of the majority class instances that will be removed from the data set.

For instance, the Hart’s condensing (CNN) algorithm [10] has been used as an under-sampling technique by applying the concept of consistent subset to eliminate the majority class instances that are sufficiently far away from the decision boundary because these are considered to be irrelevant for learning. Analogously, the Tomek links (TL) [22] have already been employed to remove the majority class instances that lie in the proximity of the decision boundary since, if two examples form a Tomek link, then either one of these can be seen as noise or both instances are borderline [5].

In contrast to CNN and Tomek links, the Wilson’s editing (ENN) technique [25] emphasizes more data cleaning than data reduction to under-sample the majority class. In this case, the ENN algorithm identifies and removes majority class instances that are considered to be noisy according to the k -nearest neighbor (k -NN) classifier. Tomek proposed two extensions of the ENN algorithm with the repeated Wilson’s editing (RENN) and All- k -NN (ALL k) methods [21]. The RENN algorithm extends the Wilson’s editing approach by repeating multiple the deletion process until no more instances can be removed. The ALL k method similar to RENN except for the fact that the value of k (i.e., the number of neighbors) is increased after each iteration.

The neighborhood cleaning (NCL) rule developed by Laurikkala [15] consists of two stages. Firstly, the ENN algorithm is used to remove majority class instances. Then, a second step is performed to find the neighbors of each minority class instance and those that belong to the majority class are removed. According to the experiments carried by the author, the NCL method processes noisy instances carefully but, it is strongly biased in favor of the minority class and leads to poor specificity and overall accuracy.

It has to be pointed out that one critical disadvantage of all the ENN-based algorithms is that they fail when a large amount of noise is present in the majority class because in such cases, noise results incorrectly classified by other noisy instances.

3 The Proposed Under-Sampling Method Based on DBSCAN

The DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm is a popular unsupervised learning algorithm that assumes that the clusters correspond to dense regions in space separated by regions of lower density [6], where density is defined as a minimum number of points within a certain distance of each other [14]. Thus the basic idea of DBSCAN is that for each point of a cluster the neighborhood of a given radius ϵ has to contain at least a minimum number of points ($minPts$). This technique presents some interesting properties since it can discover clusters of any arbitrary shape, works well even with noisy data, and is also efficient for large data sets [20].

Algorithm 1. DBSCAN

Require: Data set $D = \{p_1, p_2, \dots, p_n\}$, ϵ , $minPts$

```

1:  $C \leftarrow \emptyset$ 
2: for each unvisited instance  $p_i$  in  $D$  do
3:   Mark  $p_i$  as visited
4:    $nbhdP \leftarrow \text{Neighbors}(\epsilon, p_i)$ 
5:   if  $sizeof(nbhdP) < minPts$  then
6:     Mark  $p_i$  as noise
7:   else
8:      $C = \text{next cluster}$ 
9:      $C \leftarrow p_i$ 
10:    for each instance  $p'$  in  $nbhdP$  do
11:      if  $p'$  is not visited then
12:        Mark  $p'$  as visited
13:         $nbhdP' \leftarrow \text{Neighbors}(\epsilon, p')$ 
14:        if  $sizeof(nbhdP') \geq minPts$  then
15:           $nbhdP = nbhdP \text{ joined with } nbhdP'$ 
16:        end if
17:      end if
18:      if  $p'$  is not yet member of any cluster then
19:         $C \leftarrow p'$ 
20:      end if
21:    end for
22:  end if
23: end for
  
```

Unlike the original DBSCAN algorithm that is applied to the whole data set, the modification proposed in this paper will concentrate on obtaining a compact and representative cluster of the majority class. Besides, the new under-sampling method will take advantage of the fact that DBSCAN can detect noisy data (Step 6 in Algorithm 1), thus removing those majority class instances that have been identified as noise. On the other hand, the input parameters ϵ and $minPts$ in the proposed algorithm will be estimated using a modification of the original equations proposed by Smiti and Elouedi [20]:

$$\epsilon = \sqrt{\frac{\sum_{i=1}^{n^-} \text{dist}(m, p_i^-)}{n^-}} \quad (1)$$

where m is the middle vector of the majority class, p_i^- represents a majority class instance, dist corresponds to the Euclidean distance between p_i and m , and n^- denotes the number of majority class instances.

$$\text{minPts} = \frac{\pi \times \epsilon^2}{\text{TotalVolume}} \times n^+ \quad (2)$$

where $\text{TotalVolume} = \frac{4}{3} \times \pi \times \epsilon^3$ and n^+ is the number of minority class instances. Note that we have supposed the spherical form to shape the majority class. Therefore, minPts is based on a ratio of change between the area of a circle and the sphere volume, due to DBSCAN moves around density-reachable instances according to the Euclidean distance.

The under-sampling algorithm based on DBSCAN starts by splitting the data set into two subsets C^- and C^+ with the majority and minority class instances, respectively. Then it takes an arbitrary instance p_i^- from the majority class C^- and searches all instances density-reachable from p_i^- with respect to ϵ and minPts . If there are not minPts in the ϵ -neighborhood of p_i^- , then this is labeled as noise and removed. If an instance is found to be part of a cluster, all ϵ -neighborhood instances are also part of that cluster. If the instance p_i^- is a border point, no points are density-reachable from p_i^- and the algorithm visits the next majority class instance to discover whether it is noise or it belongs to a further cluster [27]. This process will be repeated on the majority class until ϵ and minPts do not change. The algorithm for under-sampling is described as follows:

Algorithm 2. Under-sampling based on DBSCAN

Require: Data set $D = \{p_1, p_2, \dots, p_n\}$

- 1: $D' \leftarrow \emptyset$
 - 2: Split D into two subsets: $C^- = \{p_1^-, p_2^-, \dots, p_n^-\}$ with the majority class instances and $C^+ = \{p_1^+, p_2^+, \dots, p_n^+\}$ with the minority class instances
 - 3: **repeat**
 - 4: Estimate ϵ and minPts
 - 5: **for** each unvisited instance p_i^- in C^- **do**
 - 6: Select an arbitrary instance p_i^-
 - 7: Mark p_i^- as visited
 - 8: Find all instances density-reachable from p_i^- according to ϵ and minPts
 - 9: **if** p_i^- does not have minPts in its ϵ -neighborhood **then**
 - 10: Remove p_i^- from C^-
 - 11: **end if**
 - 12: **end for**
 - 13: **until** ϵ and minPts do not change
 - 14: $D' = C^- \cup C^+$
-

As can be seen, in Algorithm 2, DBSCAN is used to take advantage of removing instances that cannot be density-reachable. Given the size of the majority class, the proposal removes most of the instances from this class that are identified as noise.

4 Experimental Set-Up

With the aim of evaluating the performance of the DBSCAN under-sampling method and comparing it with that of some state-of-the-art algorithms, we carried out a series of experiments on a large collection of imbalanced two-class

Table 1. Some characteristics of the data sets (sorted by the IR values)

	Data set	Class distribution	#Instances	IR
1	yeast-2_vs_4	51 – 463	514	9.08
2	yeast-0-5-6-7-9_vs_4	51 – 477	528	9.35
3	vowel0	90 – 898	988	9.98
4	glass-0-1-6_vs_2	17 – 175	192	10.29
5	glass2	17 – 197	214	11.59
6	shuttle-c0-vs-c4	123 – 1706	1829	13.87
7	yeast-1_vs_7	30 – 429	459	14.30
8	glass4	13 – 201	214	15.47
9	ecoli4	20 – 316	336	15.80
10	page-blocks-1-3_vs_4	28 – 444	472	15.86
11	glass-0-1-6_vs_5	9 – 175	184	19.44
12	shuttle-c2-vs-c4	6 – 123	129	20.50
13	yeast-1-4-5-8_vs_7	30 – 663	693	22.10
14	glass5	9 – 205	214	22.78
15	yeast-2_vs_8	20 – 462	482	23.10
16	flare-F	43 – 1023	1066	23.79
17	yeast4	51 – 1433	1484	28.10
18	yeast-1-2-8-9_vs_7	30 – 917	947	30.57
19	yeast5	44 – 1440	1484	32.73
20	ecoli-0-1-3-7_vs_2-6	7 – 274	281	39.14
21	abalone-17_vs_7-8-9-10	58 – 2280	2338	39.31
22	yeast6	35 – 1449	1484	41.40
23	shuttle-2_vs_5	49 – 3267	3316	66.67
24	kdd-buffer_oflow_vs_back	30 – 2203	2233	73.43
25	poker-8-9_vs_5	25 – 2050	2075	82.00

data sets taken from the KEEL Data Set Repository (<https://sci2s.ugr.es/keel/imbalanced.php#subA>). The main characteristics of the data sets are summarized in Table 1, including the a prior class distribution, the total number of instances, and the imbalance ratio (IR), that is, the ratio of the majority class size to the minority class size.

The 10-fold cross-validation method was adopted for the experimental design: each original data set was randomly divided into 10 stratified parts of size $n/10$, where n denotes the total number of instances in the data set. For each fold, nine blocks were pooled as the training set, and the remaining portion was used as an independent test set.

The k -NN classifier (with $k = 5$) was applied to the sets preprocessed by the under-sampling method here proposed and the strategies described in Sect. 2 (CNN, ENN, RUS, RENN, ALL k , NCL, TL), and also to each original training set, which can be deemed as a baseline for comparison. Then the results from classifying the test samples were averaged across the 10 runs using three standard performance evaluation measures calculated from a 2×2 confusion matrix, where each entry contains the number of correct/incorrect classifications: the sensitivity or true-positive rate (TPR), the specificity or true-negative rate (TNR), and the geometric mean of sensitivity and specificity (Gmean) computed as follows:

$$Gmean = \sqrt{TPR \cdot TNR} \quad (3)$$

where TPR and TNR are defined as the percentage of positive (minority class) and negative (majority class) instances correctly classified, respectively.

5 Experiment Results

For each data set, Table 2 reports the geometric mean (averaged across the 10 runs) of the k -NN classifier using the under-sampled training sets, along with the Friedman’s average rank for each algorithm at a significance level of $\alpha = 0.05$. Values in boldface highlight the best performing method in each data set. As can be seen, the new method outperformed the other under-sampling algorithms in 10 out of 25 databases, which represents a 40% of the cases. The reason why DBSCAN performed the best on these databases is that it keeps the most informative instances. Besides the performance of DBSCAN was quite similar to that of the best algorithms in four databases (yeast-2_vs_4, shuttle-c0-vs-c4, page-blocks-1-3_vs_4, and ecoli-0-1-3-7_vs_2-6). Conversely, this method achieved very poor results for some other databases (shuttle-c2-vs-c4, yeast-1-4-5-8_vs_7, yeast-1-2-8-9_vs_7, and abalone-17_vs_7-8-9-10), whose characteristics should be further analyzed to gain some insight.

As the Friedman test only can detect significant differences over the whole set of comparisons, the Nemenyi’s post hoc test was also computed to report any significant differences between all pairs of methods [4]. This test states that performances of two or more algorithms are significantly different if their average ranks are at least as great as their critical difference (CD) with a given level of significance:

$$CD = q_\alpha \sqrt{\frac{L(L+1)}{6N}} \quad (4)$$

where N denotes the number of databases, L is the number of algorithms, and q_α is a critical value based on the Studentized range statistic divided by $\sqrt{2}$.

Table 3 summarizes the Nemenyi's statistic for a significance level of $\alpha = 0.05$. The positive values in boldface indicate that the method in the row significantly outperforms the method in the column, whereas the negative values in boldface mean that the method in the column is significantly better than the method in the row. Despite RENN, All k and NCL appear to be the best performing techniques for the databases used in these experiments, the statistical significance study shows that no method was significantly better than the DBSCAN under-sampling algorithm (no value in the DBSCAN row is greater than the critical difference). Additionally, the cluster-based method proposed here performed better than CNN, RUS and TL.

Table 2. Geometric mean results

Data Set	Original	DBSCAN	CNN	ENN	RUS	RENN	ALL k	NCL	TL
yeast-2_vs_4	81.4	83.7	78.8	83.9	78.8	84.0	83.9	84.0	82.5
yeast-0-5-6-7-9_vs_4	52.0	60.4	56.4	69.8	54.5	74.1	75.4	77.5	68.2
vowel0	98.3	99.9	40.8	98.3	94.9	98.3	98.3	98.8	98.3
glass-0-1-6_vs_2	0.0	47.5	45.9	41.9	41.9	42.0	45.9	33.9	0.0
glass2	0.0	53.2	49.9	41.9	42.0	41.7	42.0	34.1	0.0
shuttle-c0-vs-c4	99.6	99.6	100	99.6	99.6	99.6	99.6	99.6	99.6
yeast-1_vs_7	44.7	31.6	56.4	44.6	41.9	51.6	40.8	57.5	36.5
glass4	54.8	87.7	48.7	72.6	72.2	73.4	73.0	72.6	61.2
ecoli4	86.3	86.3	94.9	89.4	97.5	89.4	89.4	92.2	86.3
page-blocks-1-3_vs_4	96.3	96.3	90.3	95.8	84.5	96.4	96.4	98.1	96.3
glass-0-1-6_vs_5	87.4	88.2	67.4	94.3	81.6	94.3	94.3	94.3	80.7
shuttle-c2-vs-c4	0.0	0.0	0.0	0.0	83.3	0.0	0.0	0.0	0.0
yeast-1-4-5-8_vs_7	0.0	0.0	29.8	18.2	52.9	31.6	18.2	31.5	0.0
glass5	74.0	88.2	60.3	81.4	88.2	81.4	74.5	81.6	74.0
yeast-2_vs_8	74.1	74.2	68.4	74.2	52.4	74.2	74.2	74.2	74.2
flare-F	30.4	72.9	43.5	64.4	62.6	68.1	54.9	62.6	34.0
yeast4	44.1	39.5	44.0	67.1	41.9	70.0	68.6	65.5	48.4
yeast-1-2-8-9_vs_7	18.2	18.2	35.9	18.2	66.7	25.8	18.3	18.2	18.3
yeast5	83.7	87.6	70.1	94.0	70.1	95.3	94.1	93.9	86.3
ecoli-0-1-3-7_vs_2-6	84.2	84.2	78.7	84.4	84.5	84.4	84.4	84.4	84.2
abalone-17_vs_7-8-9-10	26.3	18.6	34.3	34.7	74.1	34.7	32.2	39.4	26.2
yeast6	69.5	75.3	71.1	79.2	72.7	82.8	82.8	80.9	75.4
shuttle-2_vs_5	100	100	87.7	100	96.9	100	100	100	100
kdd-buffer_overflow_vs_back	98.3	98.3	0.0	98.3	81.4	98.3	98.3	98.3	98.3
poker-8-9_vs_5	0.0	52.1	0.0	0.0	42.0	0.0	0.0	0.0	0.0
Av. rank	6.64	4.80	6.28	4.46	5.14	3.46	4.12	3.84	6.26

Table 3. Summary of the Nemenyi’s test with $\alpha = 0.05$

	Original	DBSCAN	CNN	ENN	RUS	RENN	ALLk	NCL	TL
Original		−1.84	−0.36	−2.18	−1.50	−3.18	−2.52	−2.80	−0.38
DBSCAN			1.48	−0.34	0.34	−1.34	−0.68	−0.96	1.46
CNN				−1.82	−1.14	−2.82	−2.16	−2.44	−0.02
ENN					0.68	−1.00	−0.34	−0.62	1.80
RUS						−1.68	−1.02	−1.30	1.12
RENN							0.66	0.38	2.80
ALLk								−0.28	2.14
NCL									2.42
TL									
Critical difference (CD)									2.40

In class-imbalanced data problems, it is especially important to evaluate the performance on each individual class because the minority class is usually the one of major interest and with the highest misclassification costs [2].

Table 4. True-positive rates

Data Set	Original	DBSCAN	CNN	ENN	RUS	RENN	Allk	NCL	TL
yeast-2_vs_4	0.667	0.706	0.706	0.706	0.706	0.706	0.706	0.706	0.686
yeast-0-5-6-7-9_vs_4	0.275	0.373	0.431	0.490	0.431	0.549	0.569	0.608	0.471
vowel0	0.967	1.000	1.000	0.967	1.000	0.967	0.967	0.978	0.967
glass-0-1-6_vs_2	0.000	0.235	0.235	0.176	0.176	0.176	0.235	0.118	0.000
glass2	0.000	0.294	0.353	0.176	0.176	0.176	0.176	0.118	0.000
shuttle-c0-vs-c4	0.992	0.992	1.000	0.992	0.992	0.992	0.992	0.992	0.992
yeast-1_vs_7	0.200	0.100	0.367	0.200	0.176	0.267	0.167	0.333	0.133
glass4	0.308	0.769	0.769	0.538	0.846	0.538	0.538	0.538	0.385
ecoli4	0.750	0.750	0.950	0.800	1.000	0.800	0.800	0.850	0.750
page-blocks-1-3_vs_4	0.929	0.929	1.000	0.929	1.000	0.929	0.929	0.964	0.929
glass-0-1-6_vs_5	0.778	0.778	1.000	0.889	1.000	0.889	0.889	0.889	0.667
shuttle-c2-vs-c4	0.000	0.000	1.000	0.000	0.833	0.000	0.000	0.000	0.000
yeast-1-4-5-8_vs_7	0.000	0.000	0.100	0.033	0.700	0.100	0.033	0.100	0.000
glass5	0.556	0.778	1.000	0.667	1.000	0.667	0.556	0.667	0.556
yeast-2_vs_8	0.550	0.550	0.500	0.550	0.550	0.550	0.550	0.550	0.550
flare-F	0.093	0.535	0.209	0.419	0.395	0.465	0.302	0.395	0.116
yeast4	0.196	0.157	0.235	0.451	0.176	0.490	0.471	0.431	0.235
yeast-1-2-8-9_vs_7	0.033	0.033	0.133	0.033	0.667	0.067	0.033	0.033	0.033
yeast5	0.705	0.773	0.818	0.886	0.818	0.909	0.886	0.886	0.750
ecoli-0-1-3-7_vs_2-6	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714
abalone-17_vs_7-8-9-10	0.069	0.034	0.138	0.121	0.707	0.121	0.103	0.155	0.069
yeast6	0.486	0.571	0.600	0.629	0.771	0.686	0.686	0.657	0.571
shuttle-2_vs_5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
kdd-buffer_oflow_vs_back	0.967	0.967	1.000	0.967	0.667	0.967	0.967	0.967	0.967
poker-8-9_vs_5	0.000	0.280	0.000	0.000	0.176	0.000	0.000	0.000	0.000

Correspondingly, Tables 4 and 5 report the true-positive and true-negative rates, respectively. As in the previous results, again the best values in each database are shown in boldface. One should analyze the results of both tables together in order to understand the effect of each under-sampling algorithm on the performance of the classifier.

One can observe that using the original data set with no preprocessing achieved true-negative rates similar to those of the under-sampling algorithms, but at the cost of misclassifying many minority class instances as indicated by the generally low true-positive rates. Focusing on the results of the DBSCAN-based algorithm, it is worth noting that the true-positive rates are among the best ones for many databases, while the true-negative rates are always 1 or very close to 1. Nonetheless, there are still some cases (shuttle-c2-vs-c4, yeast-1-4-5-8-vs.7, yeast-1-2-8-9-vs.7, abalone-17-vs.7-8-9-10 and glass2) in which the true-positive rate was not high enough and therefore, a deeper analysis would be necessary for a better understanding of the conditions under which the under-sampling algorithm proposed here performs well.

Table 5. True-negative rates

[illegible]

6 Conclusions and Future Work

In this paper, the traditional DBSCAN clustering algorithm has been modified for under-sampling the majority class in imbalanced two-class data sets. One important feature of the proposed method is that it takes advantage of the capability of DBSCAN to discover and remove noise. Thus our under-sampling algorithm allows to eliminate the majority class instances that have been identified as noise by the basic procedure of DBSCAN. In addition, compared to other algorithms such as RUS, the new under-sampling technique can effectively avoid the important information loss of majority class. As a by-product, the proposed method reduces the size of the data sets, which is a common advantage of the under-sampling algorithms over the over-sampling strategy and results especially interesting in the current era of Big Data.

The experiments carried out over 25 two-class data sets with moderate to high class imbalance ratio have demonstrated the effectiveness of the new under-sampling method when compared with other popular state-of-the-art algorithms. In the empirical analysis, we have evaluated the classification performance of the under-sampling methods by using the true-positive rate, the true-negative rate and the geometric mean, and then we have checked for statistically significant differences through the Friedman's average ranks and the Nemenyi's post hoc test.

Although the results of this preliminary study are promising, we believe that a more exhaustive analysis is still necessary to gain deeper insight into which intrinsic data characteristics can affect the performance of the proposed under-sampling method. On the other hand, the experiments of this paper have concentrated on the k -NN classifier, but other prediction models such as decision trees and neural networks should be also tested to draw more general and robust conclusions. Finally, another avenue for further research is the generalization of our algorithm to multi-class and/or multi-label imbalanced databases.

Acknowledgment. This work was partially supported by the Universitat Jaume I under grant [UJI-B2018-49], the 5046/2020CIC UAEM project and the Mexican Science and Technology Council (CONACYT) under scholarship [702275].

References

1. Cao, P., Zhao, D., Zaiane, O.: Hybrid probabilistic sampling with random subspace for imbalanced data learning. *Intell. Data Anal.* **18**(6), 1089–1108 (2014)
2. Chawla, N.V., Cieslak, D.A., Hall, L.O., Joshi, A.: Automatically countering imbalance and its empirical relationship to cost. *Data Mining Knowl. Disc.* **17**(2), 225–252 (2008)
3. Dal Pozzolo, A., Caelen, O., Bontempi, G.: When is undersampling effective in unbalanced classification tasks? In: Appice, A., Rodrigues, P.P., Santos Costa, V., Soares, C., Gama, J., Jorge, A. (eds.) *ECML PKDD 2015. LNCS (LNAI)*, vol. 9284, pp. 200–215. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23528-8_13

4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006)
5. Devi, D., Biswas, S., Purkayastha, B.: Redundancy-driven modified Tomek-link based undersampling: a solution to class imbalance. *Pattern Recognit. Lett.* **93**, 3–12 (2017)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231. AAAI Press, Portland (1996)
7. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *Int. J. Approximate Reason.* **50**(3), 561–577 (2009)
8. García, V., Sánchez, J.S., Marqués, A.I., Florencia, R., Rivera, G.: Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. *Exp. Syst. Appl.* **158**, 113026 (2019). <https://doi.org/10.1016/j.eswa.2019.113026>
9. García, V., Sánchez, J.S., Mollineda, R.A.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowl.-Based Syst.* **25**(1), 13–21 (2012)
10. Hart, P.: The condensed nearest neighbor rule. *IEEE Trans. Inf. Theory* **14**(3), 515–516 (1968)
11. He, H., Ma, Y.: *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley - IEEE Press, Piscataway (2013)
12. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intell. Data Anal.* **6**(5), 429–449 (2002)
13. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.* **5**(4), 221–232 (2016)
14. Kumar, K.A., Rangan, C.P.: Privacy preserving DBSCAN algorithm for clustering. In: Alhajj, R., Gao, H., Li, J., Li, X., Zaiane, O.R. (eds.) *Advanced Data Mining and Applications*, pp. 57–68. Springer, Heidelberg (2007). <https://doi.org/10.1007/11811305>
15. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) *AIME 2001. LNCS (LNAI)*, vol. 2101, pp. 63–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48229-6_9
16. Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S.: Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **409–410**, 17–26 (2017)
17. Ofek, N., Rokach, L., Stern, R., Shabtai, A.: Fast-CBUS: a fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing* **243**, 88–102 (2017)
18. Onan, A.: Consensus clustering-based undersampling approach to imbalanced learning. *Sci. Programm.* **2019**, 5901087 (2019). Article ID 5901087
19. Prati, R.C., Batista, G.E., Silva, D.F.: Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowl. Inf. Syst.* **45**, 247–270 (2015). <https://doi.org/10.1007/s10115-014-0794-3>
20. Smiti, A., Elouedi, Z.: DBSCAN-GM: an improved clustering method based on Gaussian Means and DBSCAN techniques. In: *Proceedings of IEEE 16th International Conference on Intelligent Engineering Systems*, Lisbon, Portugal, pp. 573–578 (2012)
21. Tomek, I.: An experiment with the edited nearest-neighbor rule. *IEEE Trans. Syst. Man Cybern. SMC* **6**(6), 448–452 (1976)

22. Tomek, I.: Two modifications of CNN. *IEEE Trans. Syst. Man Cybern. SMC* **6**(6), 769–772 (1976)
23. Tsai, C.F., Lin, W.C., Hu, Y.H., Yao, G.T.: Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Inf. Sci.* **477**, 47–54 (2019)
24. Verma, M.K., Xaxa, D.K., Verma, S.: DBCS: density based cluster sampling for solving imbalanced classification problem. In: *Proceedings of International conference of Electronics, Communication and Aerospace Technology*, Coimbatore, India, vol. 1, pp. 156–161 (2017)
25. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern. SMC* **2**(3), 408–421 (1972)
26. Yen, S.J., Lee, Y.S.: Cluster-based under-sampling approaches for imbalanced data distributions. *Exp. Syst. Appl.* **36**(3, Part 1), 5718–5727 (2009)
27. Yue, S.H., Li, P., Guo, J.D., Zhou, S.Q.: Using greedy algorithm: DBSCAN revisited II. *J. Zhejiang Univ. - SCIENCE A* 5(11), 1405–1412 (2004). <https://doi.org/10.1631/jzus.2004.1405>