Received: 24 July 2022 | Revised: 10 November 2022 | Accepted: 12 December 2022

DOI: 10.1002/cpe.7586

RESEARCH ARTICLE

WILEY

Undersampling of approaching the classification boundary for imbalance problem

Lei Jiang[1] | Peng Yuan[1] | Jing Liao[1] | Qiongbing Zhang[1] | Jianxun Liu[1] | Keqin Li[2]

[1]Key Laboratory of Knowledge Processing and Networked Manufacture, Hunan University of Science and Technology, Xiangtan, China

[2]Department of Computer Science, State University of New York, New Paltz, USA

Correspondence
Lei Jiang, Key Laboratory of Knowledge Processing and Networked Manufacture, Hunan University of Science and Technology, Xiangtan, 411201, China.
Email: jleihn@hnust.edu.cn

Funding information
Hunan Provincial Department of Education Innovation Platform Open Fund Project, China, Grant/Award Number: 20K050; Key Project of Hunan Provincial Education Department, China, Grant/Award Number: 19A172; National Natural Science Foundation of China, Grant/Award Number: 62107013; the Ministry of education of Humanities and Social Science project, China, Grant/Award Number: 17YJAZH032

Summary

Using imbalanced data in classification affect the accuracy. If the classification is based on imbalanced data directly, the results will have large deviations. A common approach to dealing with imbalanced data is to re-structure the raw dataset via undersampling method. The undersampling method usually uses random or clustering approaches to trimming the majority class in the dataset, since some data in the majority class makes not contribute to classification model. In this paper a revised undersampling approach is proposed. First, we perform space compression in the vertical direction of the separating hyperplane. Then, a weighted random sampling hybrid ensemble learning method is carried out to make the sampled objects spread more widely near the separating hyperplane. Experiments with 7 under-sampling methods on 21 imbalanced datasets show that our method has achieved good results.

KEYWORDS

classification, imbalanced data, separation hyperplane, undersampling

1 | INTRODUCTION

With the development of big data era, the data size for classification is increased rapidly. In some dataset, the expanded speed of data catagories are not synchronized. Some catagories of data increase very rapidly, while others grows slowly. This will cause a data imbalance problem in these datasets. For example, in the statistics of bank card credit,[1,2] the number of users with good credit increases much faster than users with poor credit. Traditional machine learning methods perform well with balanced dataset. But for imbalanced dataset, their performance is usually not as effective as expected.[3] The cause of this results is the trained classifier, such as Decision Tree, Bayes Networks, and SVM,[4-6] will be biased towards the majority class,[7-9] and be easily misclassified minority class into the majority class. In practise, this biase maybe have serious consequence, such as mis-approve credit card applications from people with low credit, classify tumor cells as normal cells, and misclassify faulty parts as normal parts[5,10] which cause heavy economic losses or personal safety problems.

In order to reduce the effect of data imbalance, researchers proposed resampling technology to balance data.[11] The resampling technology includes two important methods, namely oversampling and undersampling. The undersampling method balances dataset by partially selecting the majority class dataset. Oversampling method is to artificially generate data to minority class to achieve a balance dataset.[12] Both methods are effective to imbalance classification.

In this paper, we focus mainly on undersampling method. The undersampling method is to reduce the size of majority class through resampling technology to balance the data. One of the most popular undersampling method is random undersampling (RUS).[13,14] RUS balances the data by randomly selecting samples of the majority class. The advantage of RUS is that it can quickly train the classification model, but it may eliminate useful data.[15] Another widely used undersampling method is based on clustering.[16,17] Cluster-based methods preserve the data distribution characteristics

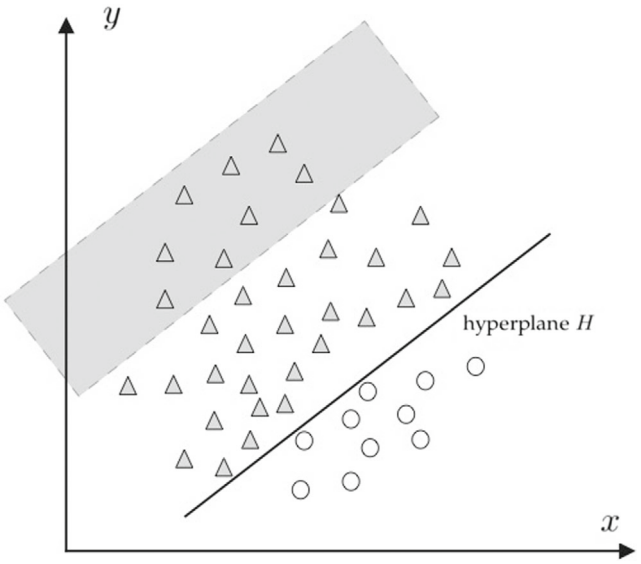

FIGURE 1 Data and its classification hyperplane

while retaining the useful samples.[16,18] The cluster-based method divides the majority class into multiple groups, and then selects representative samples from every group.[19] However, cluster-based method rely on experience to determine the number of group and to select representative samples. In addition, undersampling technique are usually combined with ensemble learning method[20] such as Adaboost and AsymBoost.[21] In the study of undersampling methods, noise filtering is needed in data preprocessing for imbalanced data learning. Noise filtering mainly include *threshold-adjusted filter*,[22] *iterative-partitioning filter*,[23] and *KNN filter*.[10]

We notice that the undersampling method essentially discards part of the data in a majority class through re-sampling. The experiments of Van[22] and S *á* ez[23] proved that not all data are useful for classification. Researchers[24,25] have also found that sampling in border areas can improve the classification effect in the oversampling study. Inspired by these, we studied that how to actively discard samples with small contributions. As the shaded data shown in Figure 1, we have an intuitive that data far away from the separating hyperplane has little contributes to classification learning. Through probability analysis we also found that the sampling space obtained by space compression, as long as it is closer to the classification boundary, the higher the classification accuracy. In this way, the problem is transformed into how to find an undersampling method that makes the sample set approach the classification boundary. Therefore, we propose an method which is used adaptive space compression, weighted sampling and ensemble learning method to make the sampling approach the classification boundary.

The rest of our paper is structured as follows. Section 2 discusses related works. The detailed description of the model is given in Section 3. Section 4 presents extensive experiments to justify the effectiveness of our proposed method. Finally, summary and future works are included in Section 5.

2 | RELATED WORK

In present, researchers are gradually paying more attention to the imbalanced classification problem,[26-28] and propose a large number of solutions.[29,30] The most direct method is to adjust the bias of the classifier to majority classes during classification. However, the degree of bias is difficult to define and describe which makes this method hard to implement.[26,31] Currently, data processing and classification are separated in a large number of studies. That is, they perform balance processing on imbalanced data firstly, and then start the subsequent classification tasks. From the current research outputs, we can see that the research of undersampling methods have achieved remarkable results. The undersampling methods are mainly based on RUS to discard some majority class data according to the distribution of data to re-balance data.[32,33] There is also quite a lot of research on clustering methods which dig out the data distribution and retain the useful samples through clustering.[16,18] However, the clustering methods need to define the cluster number first. Generally, the data distribution description will change with the number of clusters. Obviously, an inaprociate number of cluster would amplify this difference and affect the predictive performance of the classifier, and the new dataset obtained in this way may be very different from the raw dataset.

In recent studies, in order to determine the k value of cluster in undersampling, Tsai[16] designed CBIS to solve the problem by using AP algorithm.[34] The AP algorithm measures the Euclidean distance among all data points to calculate their similarity. Then, the value of k is determined according to responsibility and availability to divide the clusters. This improvement has achieved good results. However, the clustering of AP

algorithm still requires certain experience, and the determination of the number of clusters is still limited. Ng[35] developed a method similar to clustering named Hashing-Based Undersampling (HBUS). HBUS performs hash calculation on the majority class dataset, divide the majority class into multiple subclass via hash value. The re-sampling majority dataset is determined by the mapping relationship between majority class data and the minority class data. However, the way of dividing the space still has an impact on the final sampling selection.

Some researchers recently have introduced big data methods for sample selection which achieved good results. Koziarski[36] used the concept of mutual potential which is proposed by Krawczyk[37] in oversampling to guide the selection of majority class data in undersampling. It ranks according to the classification potential of most observers, and then determines the order of undersampling of majority class data based on this ranking. This method brings a lot of computational overhead. Since most of the existing undersampling methods usually separate data processing from classifiers, Peng[38] suggested to parametrize the data sampler and integrated the optimization of evaluation metrics into the data sampling process. Then the data sampling procedure is formulated as a Markov decision process (MDP), and uses reinforcement learning to train the data sampler.

It can be seen that the latest undersampling methods mentioned above are all biased sampling. The method they chose is either to preserve the structure of the dataset through clustering, or to use some approach to increase the probability of some samples being sampled, such as hash value and mutual potential. In other words, there is no unified theory and method on how to determine high-value samples. At present, this issue is still open, and it is also a promising field.

3 | PROPOSED METHOD

In this section, we propose an undersampling method as shown in Figure 2. First, we preprocess the raw data to remove the noise in the dataset and obtain a preliminary separation hyperplane *H* as classification boundary. Then, we apply an adaptive space compression technique to make sampling space of undersampling closed to *H*. Finally, a combination of weighted sampling and ensemble learning is used to approach the *H*.

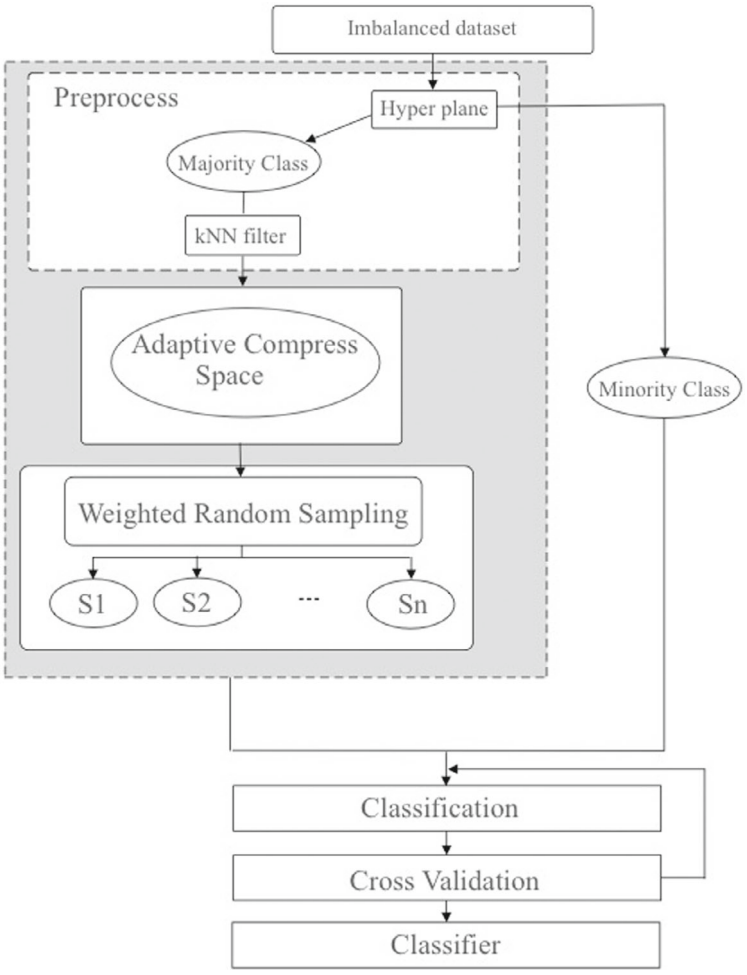

FIGURE 2 The steps of the proposed method

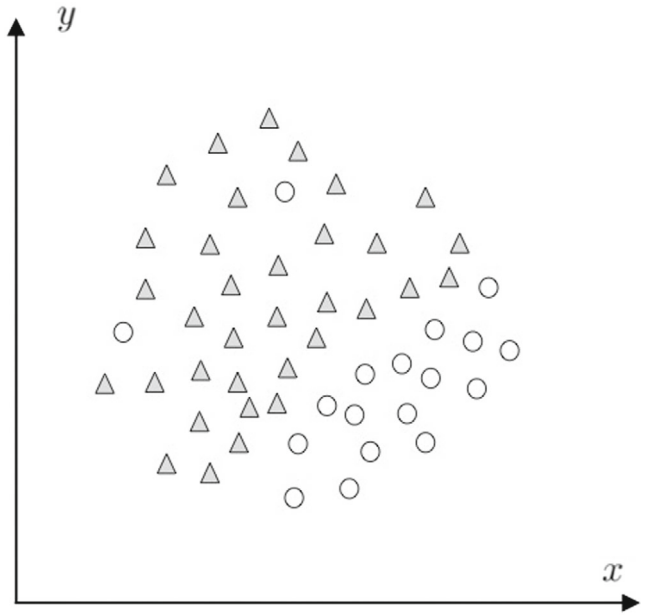

FIGURE 3 The noise of minority class in majority class

3.1 | Preprocessing

Before the re-sampling method, we refer to the work of Kang[10] to preprocess the data. That is, as shown in Figure 3, we use KNN approach to filter the *circle* samples surrounded by *triangle* samples as noise. The new dataset thus obtained will be used to construct a classification model.

The proposed method is to extract data from both the vertical and parallel directions on separating hyperplane. Obviously, the separating hyperplane is the foundation. Therefore, we roughly divide all samples by using linear SVM to obtain the hyperplane. Herein, the linear kernel function is used in SVM and its formula is shown in Equation (1):

$$K(x, x_i) = x \cdot x_i \tag{1}$$

Then, the hyperplane $\hat{H}$ formula is as follows:

$$\hat{H} : wx + b = 0 \tag{2}$$

3.2 | Adaptive space compression technique based on separation hyperplane

3.2.1 | Theoretical analysis of space compression based on separating hyperplane

In this part, we will first conduct probabilistic analysis to prove that it is feasible to find high-contribution samples through space compression. Then the algorithm will be designed.

There is an imbalanced dataset D with two classes, one is D_u, the other is D_b, and $|D_u|/|D_b| > 1.5$. We denote the class label of D by $C_i \in \{u, b\}$ and the sample as $x_i, i \in \{1, \cdots, n\}$.

Consider a set D_{us} whose elements are sampled from the majority class D_u with a probability measure μ in a Hilbert space $\mathcal{H}$. Herein, μ has a bounded support $\mathcal{M}_u$.

Definition 1. Let $B_\epsilon(x) \subset \mathcal{H}$ denote an open ball of a radius ϵ around sample x:

$$B_\epsilon(x) = \{u \in \mathcal{H} : ||x - u|| < \epsilon\} \tag{3}$$

Then, the covering number of $\mathcal{M}_u$, $\mathcal{N}(\epsilon, \mathcal{M}_u)$ is defined as the smallest number of a series of B_ϵ whose union contains $\mathcal{M}_u$:

$$\mathcal{N}(\epsilon, \mathcal{M}_u) = \inf\{k : \exists u_1, \cdots, u_k \in H s.t. \mathcal{M}_u \subset \bigcup_{i=1}^{k} B_\epsilon(u_i)\} \tag{4}$$

Definition 2. The support of the probability measure μ defined on D_u is defined as:

$$\mathcal{M}_u = \{x \in D_u : \forall \epsilon > 0, \mu(B_\epsilon(x)) > 0\}$$

Lemma 1. *Let H be a separating hyperplane for the D. Its training set $\mathcal{A}$ is consisted of D_b and D_{us}. D_{us} contain at least N_u samples which are drawn i.i.d. according to a probability measure μ from $\mathcal{M}_u$. Herein, $N_u \geq \mathcal{N}(\epsilon/2, \mathcal{M}_u)$. Then the probability of classification accuracy from the $\mathcal{A}$ with the linear classifier is lower bounded as*:

$$Pr(\hat{C}(x) = u) \geq 1 - \frac{\mathcal{N}(\epsilon/2, \mathcal{M}_u)}{2N_u} \tag{5}$$

Proof. Let $N_u = n$. For any $x_i \in \mathcal{M}_u$, we have

$$Pr(||x - x_i|| > \epsilon | x_i) = (1 - \mu(B_\epsilon(x_i)))^{n-1}$$

Following Kulkarni and Posner,[39] we take an $\epsilon/2$-covering of $\mathcal{M}_u$ and get a series of balls, $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_{\mathcal{N}(\epsilon/2, \mathcal{M}_u)}$. For $x_k \in \mathcal{M}_u$, $\exists \mathcal{B}_j \subset B_\epsilon(x_k)$. Let $N = \mathcal{N}(\epsilon, \mathcal{M}_u)$. We can define an $\epsilon/2$ -partition as follows.

$\forall i = 1, 2, \cdots, N$, let

$$P_i = \mathcal{B}_i - \bigcup_{k \neq i} \mathcal{B}_k.$$

Then $P_i \subset \mathcal{B}_i$, and

$$\bigcup_{i=1}^{N} P_i = \bigcup_{i=1}^{N} \mathcal{B}_i$$

Furthermore, $P_i \cap P_j = 0$, and

$$\sigma_{i=1}^{N} \mu(P_i) = 1.$$

Then, for $x_k \in \mathcal{M}_u$, $\exists P_i \subset \mathcal{B}_i \subset B_\epsilon(x_k)$ is established. Namely, $p_i = \mu(P_i) \leq \mu(B_\epsilon(x_k))$. Hence

$$Pr(||x - x_i|| > \epsilon | x_i) = (1 - p_i)^{n-1}$$

and

$$Pr(||x - x_i|| > \epsilon) = \Sigma_{i=1}^{N} p_i (1 - p_i)^{n-1}.$$

We use the result in the proof of Theorem 1 in Reference 39, that is

$$Pr(||x - x_i|| > \epsilon) \leq \frac{N}{2n}$$

And following Vural,[40] we get the lower bound of probability of classification accuracy:

$$Pr(||x - x_i|| < \epsilon) \geq 1 - \frac{\mathcal{N}(\epsilon/2, \mathcal{M}_u)}{2N_u}$$

namely,

$$Pr(\hat{C}(x) = u) \geq 1 - \frac{\mathcal{N}(\epsilon/2, \mathcal{M}_u)}{2N_u}$$

■

Theorem 1. *There is an ideal separation hyperplane H on the dataset D, which can classify D_b and D_u perfectly. Let the hyperplane called H′ which is parallel to H and their distance is L. Then B_L is the space enveloped by H and H′. Set D_{us} is sampled i.i.d from B_L. Then the accuracy of the classification model f obtained by training on D_{us} and D_b increases as L decreases.*

Proof. In order to balance with the minority class, we sample a total of $N_u = 1.5 \times |D_b|$ objects to form D_{us} in every undersampling. Then, according to Equation (3), when the value of N_u does not change, $\mathcal{N}(\epsilon/2, \mathcal{M}_u)$ will become smaller as the value of L becomes smaller. Namely,

$$1 - \frac{\mathcal{N}(\epsilon/2, \mathcal{M}_u)}{2N_u}$$

will become larger accordingly. ∎

3.2.2 | Adaptive space compression technique

According to the above theoretical analysis, we argue that the closer to the separation hyperplane to sample, the better the classification model obtained. Therefore, as shown in the Figure 4, we first obtain L_{max} which is the maximum distance between sample of majority class D_u to $\hat{H}$. Then, let $L = L_{max} \times \alpha$, where α is the space compression factor and $\alpha < 1$. In this way, the compressed space B_L is obtained. Finally, random samples are drawn from B_L to form D_{us}. Herein, the compression factor α needs to be determined. Obviously, the compression factor of datasets with different imbalance rates are different. Therefore, we adopt an adaptive method to set α. Since the number of decimation is set to be $1.5 \times |D_b|$, we keep reducing α until the number of samples in the compressed space is between $1.5 \times |D_b|$ and $2 \times |D_b|$. The detailed process is shown in Algorithm 1.

3.3 | Weighted sampling based on separating hyperplane

3.3.1 | Theoretical analysis of sampling in the directions of parallel separating hyperplanes

As shown in Figure 5, if the undersampling is done only one time, there will be few samples involved in the construction of the classification model in the direction of the parallel separating hyperplane. Therefore, it is generally perform undersampling multiple times and ensemble learning is used to increase the direction of the parallel separation hyperplane. But what will happen to this increase? We will conduct the following analysis.

Lemma 2. *For an imbalanced dataset D, its majority class D_u has m samples. Random sampling is performed m times with replacement in D_u, and one sample is drawn in each time. When $m \to \infty$, the probability of sample X_i not being selected is $1/e$. Namely,*

$$\lim_{m \to \infty} Pr(X_i) = \frac{1}{e} \tag{6}$$

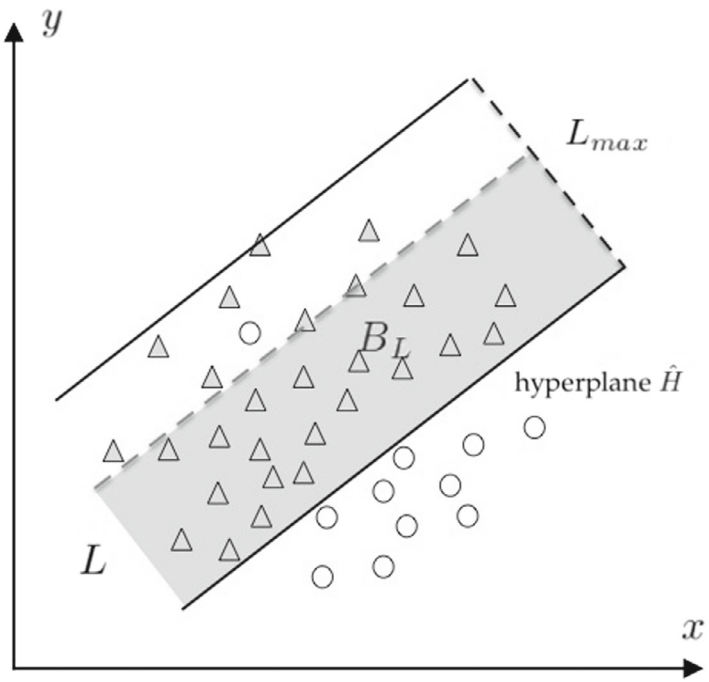

FIGURE 4 Compression space B_L

Algorithm 1. ASC: Adaptive space compression

Input: Training dataset D
Minority class D_b
Hyperplane $\hat{H}$
Multiple factor rDecreasing number d
Output: Compressed space dataset B_L
Set Space compression factor $\alpha = 1, k = 0$
for $i = 1$ to n **do**
Calculate the distance L_i from the sample x_i to the $\hat{H}$.
$L_i \leftarrow \frac{1}{|w|}(wS_i + b)$
Put L_i into distance set *dset*
Gain L_{max}
end for
while $!(r \times |D_b| > k > 1.5 \times |D_b|)$ **do**
$B_L \leftarrow \phi$
for $j = 1$ to n **do**
$k \leftarrow 0$
if $L_j \leq \alpha L_{max}$ **then**
$b_k \leftarrow x_j$
Put b_j into B_L
k++
end if
end for
$\alpha \leftarrow \alpha - d$
end while

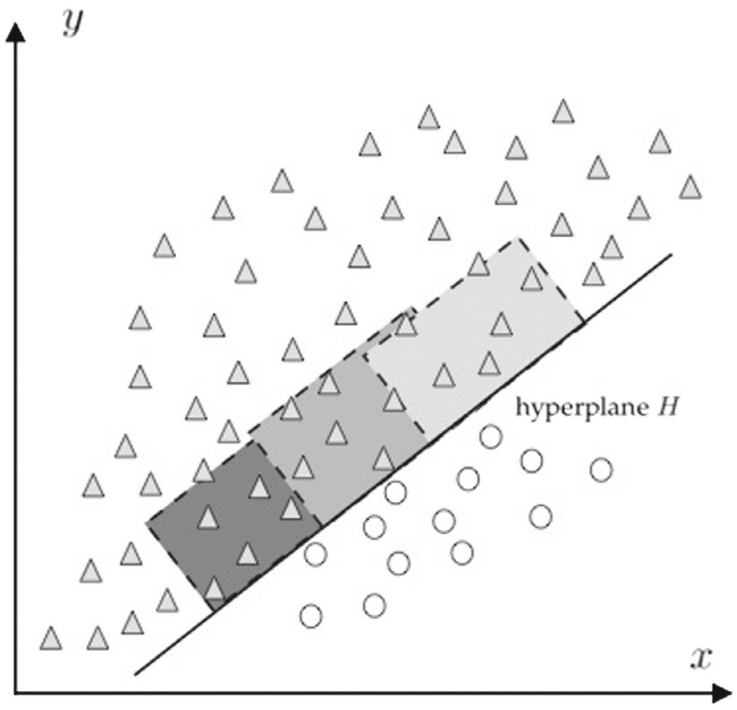

FIGURE 5 Ensemble learning of under-sampling classification

Proof. The probability of a sample being drawn is $1/m$, and the probability of not being drawn is $(1 - 1/m)$. If it is drawn m times, the probability of sample X_i not being drawn is

$$\left(1 - \frac{1}{m}\right)^m$$

When $m \to \infty$,

$$\left(1 - \frac{1}{m}\right)^m = \frac{1}{e}$$

■

From Lemma 2, we can know that if sampling with replacement is used. Even if you draw an infinite number of times, 36.8% of the samples will not be drawn. Obviously, this will greatly affect the effect of classification.

Theorem 2. *For an imbalanced dataset D, its majority class D_u has m samples. t samples are randomly selected in each time without replacement. In this way, m times consecutive extractions are done. When $m \to \infty$, the probability of sample X_i not being selected is*

$$Pr(X_i) = \left(1 - \frac{t}{m}\right)^m \tag{7}$$

Proof. The probability of a sample being drawn is $1/m$, and the probability of not being drawn is

$$\left(1 - \frac{1}{m}\right)\left(1 - \frac{1}{m-1}\right)\cdots\left(1 - \frac{1}{m-(t-2)}\right)\left(1 - \frac{1}{m-(t-1)}\right) = 1 - \frac{t}{m}.$$

Thus, after m times consecutive extractions, the probability of a sample not being drawn is

$$\left(1 - \frac{t}{m}\right)^m.$$

■

It can be known from Lemma 2 and Theorem 2 that when the number of samples of a majority class m is far greater than the number of sampling t, t/m will be very close to $1/m$. That is, about 36.8% of the samples in the parallel direction close to H will not be drawn to construct a classification model. For ensemble learning in this situation, the probability of a sample not being drawn is

$$\left(1 - \frac{t}{m}\right)^a.$$

where a is not an infinite number. It represents the number of classifiers in ensemble learning. That is, the number of undersampling classifications. Obviously,

$$1 - \frac{t}{m} > \left(1 - \frac{t}{m}\right)^a > \left(1 - \frac{t}{m}\right)^m.$$

It can be seen from the above formula that ensemble learning using random sampling is better than single undersampling classification. However, there are still quite a few samples in the direction of the parallel separating hyperplane that have not been used to construct the classification model.

3.3.2 | Weight generation mechanism based on separating hyperplane and its sampling

In order to get a large spreading area of samples near the separation hyperplane, an effective method is requied which will increase the probability of samples close to H and reduce the probability of samples far from H in the B_L space. Therefore, we propose a weight generation mechanism based on separating hyperplanes. Firstly, the sample i obtains a random sampling probability u_i through uniform distribution. Then, u_i is scaled according to the distance between i and the separating hyperplane H to get a new probability k_i. And the k_i is given in Equation (7):

$$\begin{cases} k_i = u_i^{\frac{||w||}{wx_i+b}} \\ u_i = random(0, 1) \end{cases} \tag{8}$$

And the weighting random sampling procedure is shown in Algorithm 2.

Algorithm 2. WRS: Weighting random sampling

Input: Minority class D_b
Compressed space dataset B_L
Output: Sampling results D_{us}
$U \leftarrow \emptyset$
for $i = 1$ to $|B_L|$ **do**
Use Equation (7) to get k_i
Put k_i into U
end for
$K \leftarrow \lfloor 1.5 \times |D_b| \rfloor$
$S \leftarrow$ select the top K in U
for $i = 1$ to K **do**
$j \leftarrow$ index of S_i
Put b_j into D_{us}
end for

Algorithm 3. UACB

Input: imbalanced dataset D
Multiple factor r
Decreasing number d
Output: Classification model f
$D' \leftarrow KNN(D)$
$\hat{H} \leftarrow$ linear $svm(D')$
$B_L \leftarrow \text{ASC}(D', D_b, \hat{H}, r, d)$
for $i = 1$ to n **do**
$D_{us}^{(i)} \leftarrow \text{WRS}(D_b, B_L)$
end for
$f \leftarrow \text{AdaBoost}(D_{us})$

3.4 | Our method: UACB and its time complexity

According to the foregoing description, we named our method UACB (**U**ndersampling of **A**pproaching the **C**lassification **B**oundary). Then the algorithm is shown as follows (Algorithm 3).

In order to have enough objects in B_L to be sampled, we set the number of samples in B_L to be between 1.5- r times of $|D_b|$. The main reason is that B_L is closer to the ideal classification boundary H the better the effect which we get by theoretically analyzed. But, we are using the linear separation hyperplane $\hat{H}$ to simulate the H. The sampling result will be biased towards to $\hat{H}$ and deviated from H. We note that *WRS* is a random sampling with a weight biased towards to $\hat{H}$. Therefore, as long as there are enough objects in B_L to be sampled, some samples biased to H will have chance to be sampled. Then the problem will be alleviated.

Next, we analyze the time complexity of UACB. Firstly, Assuming the time complexity of KNN and linear SVM in preprocessing are both $O(N^2)$. Then, in the vertical of to H, the time complexity of compressing the space to obtain B_L is $O(N)$. And the *Weighting random sampling* is $O(N)$. Lastly, the AdaBoost's time complexity is $O(N^2)$. Therefore, the time complexity of UACB is

$$O(N^2) + O(N^2) + O(N) + O(N) + O(N^2) = O(N^2).$$

4 | EXPERIMENTS AND RESULTS

In this section, we present the details of experiments to test the proposed method UACB. First, we introduce the experimental setup, including datasets, benchmark methods, parameters and metrices. Second, we do the results analysis including comparation with the benchmark methods.

TABLE 1 Dataset statistics

Dataset	Dimension	Examples	Ir
Abalone9-18	8	731	16.4
Ecoli-0_vs_1	7	220	1.86
Ecoli1	7	336	3.36
Car-good	6	1728	24.04
Cleveland-0_vs_4	13	177	12.62
Dermatology-6	34	358	16.9
Glass-0-1-2-3_vs_4-5-6	9	214	3.2
Iris0	4	150	2
Kr-vs-k-zero-one_vs_draw	6	2901	26.63
New-thyroid2	5	215	5.14
Page-blocks0	10	5472	8.79
Segment0	19	2308	6.02
Shuttle-c0-vs-c4	9	1829	13.87
Shuttle-c2-vs-c4	9	129	20.5
Led7digit-0-2-4-5-6-7-8-9_vs_1	7	443	10.97
Vehicle1	18	846	2.9
Winequality-white-3_vs_7	11	900	44
Page-blocks-1-3_vs_4	10	472	15.86
Yeast-1-4-5-8_vs_7	8	693	22.1
Paw02a-600-5-30-BI	2	600	5
Yeast5	8	1484	32.73

4.1 | Experimental setup

Datasets We conduct experiments on 21 imbalanced datasets from **keel**[*]. The relevant information of the datasets is shown in Table 1. We can see that the range of imbalance rate (ir) of datasets is $(1.86, 44)$, the range of examples of datasets is $(129, 5472)$, and the range of dimension of datasets is $(2, 34)$. For each dataset, we used a 5-fold cross validation method to carry out the experiment, and repeat it 10 times. Herein, the average of 10 results of the experiment is used as the final results.

Benchmark methods As mentioned in Section 2, we used 5 latest imbalanced undersampling methods and 2 classic algorithims listed below as our benchmark methods.

1. **RUS**[13] randomly selects the number of samples from majority class as equal as minority class, then obtains a classifier by using AdaBoost.
2. **CBUS**[41] uses the k-means method to find k cluster centers in the majority class, and select samples which nearest to the centroid in its i th cluster from majority class . At last, AdaBoost is used to obtain a classifier.
3. **CBIS**[16] uses the AP algorithm to cluster data of majority class, then uses the IS3[42] method to select samples from each cluster of majority class, combine them with minority class samples to form train set. Then ensembles learning is used to get the classifier.
4. **RBU**[36] uses the Gaussian kernel function to calculate the mutual class relationship between the majority class samples and the minority class objects based on the mutual class potential, and through the diffusion kernel radius to achieve the balance of majority and minority class. Finally the naive bayes is applied to get classification.
5. **HUE**[35] uses IQT[43] algorithm to divide the majority class space into multiple hash subspaces, then calculates the Hamming distance between the subspace and the minority class space, and then weights them to obtain the selected subspace. At the same time, in order not to lose all other majority class information, The subspace is set to be the reference subspace, and most other samples are selected to form the final sample set. Finally the classifier is trained by using ensemble learning.
6. **TU**[38] is a deep learning method. It parameterize the data sampler, optimize and integrate the evaluation metric into the data sampling process, then abstract this process into a Markov process. The classification model is trained via reinforcement learning.

7. **UA-KF**[10] performs noise filtering on minority class data, then randomly undersamples majority data, and finally uses AdaBoost for ensembles learning.

Parameters and Metrices In our proposed method, we set $k = 15$ in the KNN filtering step, and use 5 cart trees as weak classifiers in AdaBoost step.

It is well known that the ratio of the majority class to the minority class exceeds 1.5 as imbalanced data, so we set the ratio of the sampled data to the ratio of the majority class to the minority class to 1.5.

For an imbalance dataset, accuracy of classification will be bias to data of majority class. The result of accuracy is not convincing. In this work, we use two criterion, F-measure and AUC, to evaluate experimental performance.[44,45] The F-measure is calculated according to Equation (14). AUC can be obtained by calculating the area of ROC. ROC is the relationship curve between False Positive Rate and True Positive Rate. The relevant formula is as follows,

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \tag{9}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{10}$$

$$False\ Negative\ Rate = \frac{TN}{TN + FP} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$Fmeasure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

and the meaning of the variables involved in the formula is shown in Table 2.

4.2 | Experiments

4.2.1 | Analysis of KNN filtering noise

We adopted the KNN method to filter the noise of the dataset. In order to determine whether this step is effective, we compared the UACB algorithm with the KNN denoising step and the UACB without of this step. It can be seen from Figure 6 that KNN can improve the algorithm in the all datasets. In order to further distinguish whether the effect of the algorithm is completely caused by denoising, we compared UACB with UA-KF. UA-KF is the recently released algorithm that we can find in authoritative journals that uses KNN denoising and random undersampling. From Figure 6, we can see that whether UACB uses KNN to denoise or not, its results are better than UA-KF. This shows that although KNN denoising improves our algorithm, the improvement of the algorithm result is mainly caused by sampling in the vertical and parallel directions on the separating hyperplane.

4.2.2 | Analysis of the effect of space compression

It is known from Theorem 1 that the closer the separation hyperplane is to the sampling, the better the classification result. In this section, experiments is designed to verify it. We used a series of α values to test the performance of our algorithm in different compressed spaces. Then, it is shown in Tables 3,4, we set the value of α to 1, 0.75, 0.5, 0.25, 0.2. Relying on these settings, we can find from Tables 3,4 that the classification results of each dataset gradually become better with the decrease of the space compression factor α. That is, the feasibility of the space compression proposed by us is confirmed from both the experimental results and the probability analysis.

TABLE 2 Confusion matrix

	Predicted positive	**Predicted negative**
Actual positive	True Positive(TP)	False negative(FN)
Actual negative	False Positive(FP)	True negative(TN)

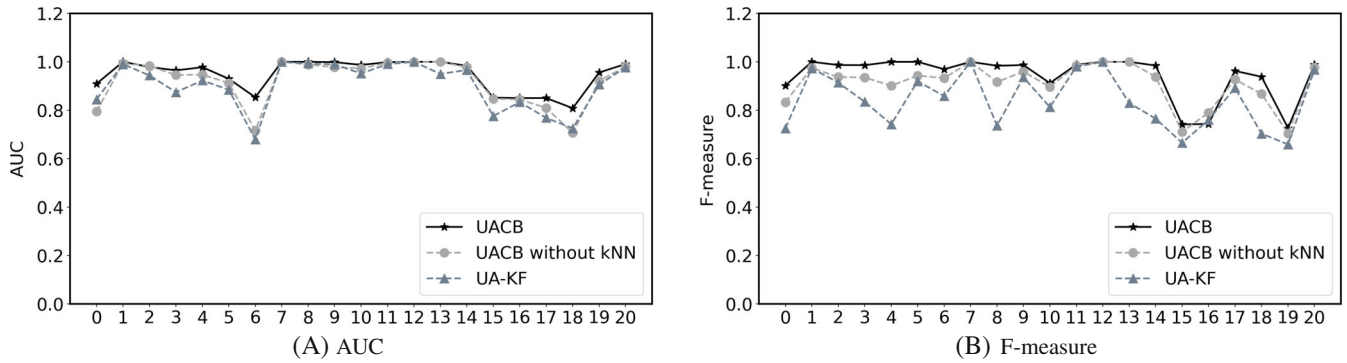

FIGURE 6 Comparison with KNN filtering

TABLE 3 AUC of different ratio, "—" indicates that sufficient majority class samples cannot be obtained under the corresponding ratio

Dataset	1	0.75	0.5	0.25	0.2
Abalone9-18	0.9091	0.9276	0.9357	—	—
Ecoli-0_vs_1	1	1	1	1	1
Ecoli1	0.9787	0.9803	0.9811	0.9828	0.9881
Car_good	0.9916	0.9984	—	—	—
Cleveland-0_vs_4	0.95	0.975	1	1	1
Dermatology-6	1	1	1	1	1
Glass-0-1-2-3_vs_4-5-6	0.9884	0.9969	0.9984	0.9992	0.9969
Iris0	1	1	1	1	—
Kr-vs-k-zero-one_vs_draw	0.991	0.9943	0.9935	0.9975	0.9961
New-thyroid2	0.9987	0.9968	0.9984	0.9989	0.9989
Page-blocks0	0.9872	0.9855	0.9846	0.9842	0.9838
Segment0	0.9986	0.9971	0.9984	0.9989	0.9985
Shuttle-c0-vs-c4	1	1	1	—	—
Shuttle-c2-vs-c4	1	—	—	—	—
Led7digit-0-2-4-5-6-7-8-9_vs_1	1	1	0.9933	1	1
Vehicle1	0.8519	0.8582	0.8643	0.8732	0.8483
Winequality-white-3_vs_7	0.8501	0.8375	—	—	—
Page-blocks-1-3_vs_4	0.9833	0.9942	0.9952	0.9928	0.9846
Yeast-1-4-5-8_vs_7	0.8083	—	—	—	—
Paw02a-600-5-30-BI	0.9131	—	—	—	—
Yeast5	0.9909	0.9916	0.9937	—	—

4.2.3 | Displaying the distribution of weighted random sampling based on separation hyperplane

In this section, we will visualize the sample distribution from the weighted random sampling based on the separating hyperplane and analyze its effect. Here, the combination of the second and sixth feature in the 9-dimensional dataset εshuttle-c0-vs-c4ε is used to visualize the effect of weighted random sampling (Figure 7). The minority class in the figure is marked with triangles, and the majority class after undersampling are marked with circles.

TABLE 4 F-measure of different ratio, "—" indicates that sufficient majority class samples cannot be obtained under the corresponding ratio

Dataset	1	0.75	0.5	0.25	0.2
Abalone9-18	0.7851	0.8738	0.9018	—	—
Ecoli-0_vs_1	1	1	1	1	1
Ecoli1	0.9742	0.9631	0.9741	0.9806	0.9867
Car_good	0.9806	0.9862	—	—	—
Cleveland-0_vs_4	0.9333	0.9333	1	1	1
Dermatology-6	1	1	1	1	1
Glass-0-1-2-3_vs_4-5-6	0.9512	0.9603	0.9639	0.9789	0.9694
Iris0	1	1	1	1	—
Kr-vs-k-zero-one_vs_draw	0.9716	0.9818	0.9766	0.9862	0.9837
New-thyroid2	0.9867	0.9867	0.9846	0.9933	0.9867
Page-blocks0	0.9225	0.9181	0.9151	0.9164	0.9121
Segment0	0.9862	0.9924	0.9892	0.9879	0.9883
Shuttle-c0-vs-c4	1	1	1	—	—
Shuttle-c2-vs-c4	1	—	—	—	—
Led7digit-0-2-4-5-6-7-8-9_vs_1	0.9692	0.9846	0.9814	0.9846	0.9846
Vehicle1	0.7106	0.7236	0.7413	0.7568	0.7413
Winequality-white-3_vs_7	0.7933	0.7433	—	—	—
Page-blocks-1-3_vs_4	0.9664	0.9628	0.9646	0.9746	0.9625
Yeast-1-4-5-8_vs_7	0.7266	—	—	—	—
Paw02a-600-5-30-BI	0.9381	—	—	—	—
Yeast5	0.9882	0.9777	0.9882	—	—

Firstly, from Figure 7A,B, we can see that all sampling results after 8 times boost in ensemble learning are basically consistent with the distribution that has not been undersampled after space compression. Consider the situation of other datasets, we finally set the number of ensemble learning to 10 times.

Secondly, Figure 7C,D verify that the probability of repeated sampling of samples near the boundary increases when weighted random sampling is used. At the same time, the boundary formed by them is quite clear. Obviously, the model effect obtained in this way will be better.

Finally, since we set the weights based on the separating hyperplane $\hat{H}$ and $\hat{H}$ is probably not the best, the majority class after undersampling may have some bias. From Figure 7 we can see that although those $\hat{H}$-based objects will be sampled multiple times in the ensemble learning, other objects still have a chance to be sampled. This will disturb it and improve the accuracy of the final result.

4.2.4 | Experiments analysis

Comparison with classic algorithms We chose two classic algorithms of RUS (based on random extraction) and CBUS (based on clustering) for comparison. Under the AUC indicator, it can be seen from Table 5 that the UACB method is ahead of RUS in the other 20 datasets except *winequality-white-3_vs_7*. Among them, 8 datasets are far ahead (the difference exceeds 3). Simultaneously, we can see that UACB is better than CBUS in all 21 datasets, and 14 datasets are significantly ahead. From Table 6, we can also conclude that UACB is still superior to RUS and CBUS under the F-measure evaluation.

Comparison with latest algorithms We used the latest undersampling method published in 3 authoritative journals and 1 top conference to compare with UACB. As shown in Tables 5,6. It can be seen from the experimental results that UACB is superior to other imbalanced sampling methods, which fully demonstrates the effectiveness of UACB. AUC average of UACB were 2.00%, 5.17%, 2.74%, 3.68% better than CBIS, RBU, HUE and TU respectively. And F-measure average of UACB are 4.37%, 11.65%, 5.90%, 5.98% better than these algorithms respectively.

CBIS uses AP algorithm for clustering and uses IS3 algorithm to select samples. HUE divides the majority class space into multiple hash subspaces, then calculates the Hamming distance between the subspace and minority class space, and then weights them to obtain the selected

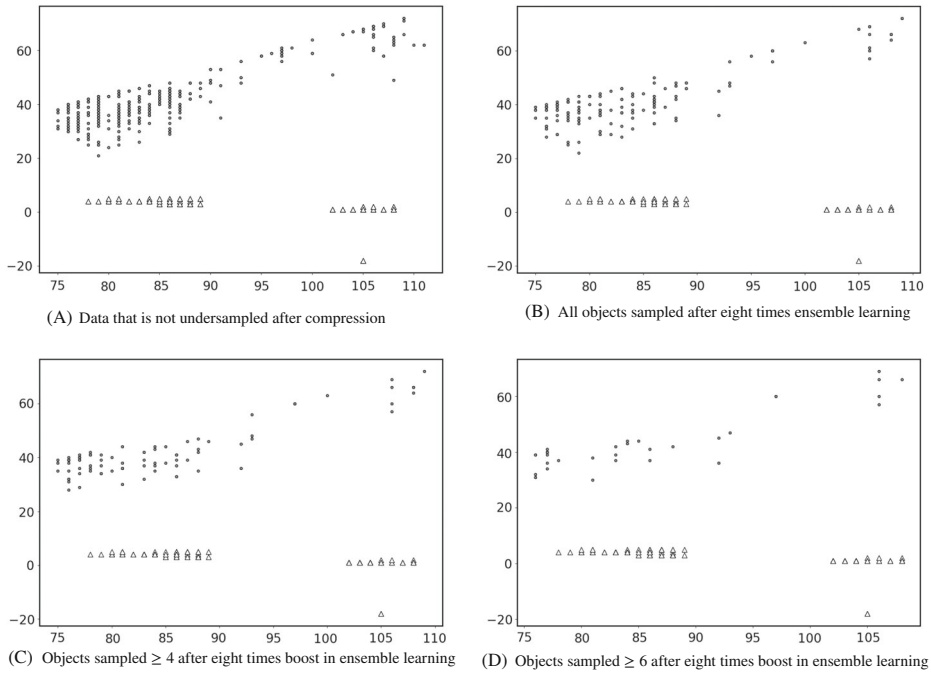

FIGURE 7 Data distribution of undersampling ensemble learning after space compression of *shuttle-c0-vs-c4*

TABLE 5 AUC of comparison methods

Dataset	UACB	RUS	CBUS	CBIS	RBU	HUE	TU	UA-KF
Abalone9-18	**0.9357**	0.8369	0.6598	0.8892	0.8084	0.8312	0.8949	0.8446
Ecoli-0_vs_1	**1**	0.9932	0.9895	0.9823	0.9528	0.9945	0.9938	0.9902
Ecoli1	**0.9881**	0.9375	0.9214	0.9586	0.9038	0.9325	0.9544	0.9441
Car-good	**0.9984**	0.9703	0.9923	0.9817	0.9053	0.9847	0.9978	0.9053
Cleveland-0_vs_4	**1**	0.9333	0.9935	**1**	0.8739	**1**	0.9000	0.8739
Dermatology-6	**1**	0.9575	0.9750	0.9875	0.9985	0.9473	0.9975	0.9985
Glass-0-1-2-3_vs_4-5-6	**0.9969**	0.9785	0.9138	0.9547	0.9914	09359	0.9585	0.9914
Iris0	1	1	1	1	1	1	1	1
Kr-vs-k-zero-one_vs_draw	**0.9961**	0.9668	0.9945	0.9831	0.8781	0.9912	0.9791	0.8781
New-thyroid2	**0.9989**	0.9962	0.9857	0.9567	0.9437	0.9956	0.9926	0.9896
Page-blocks0	**0.9838**	0.9573	0.9665	0.9780	0.9401	0.9672	0.9668	0.9518
Segment0	**0.9985**	0.9929	0.9885	0.9968	0.9968	0.9928	0.9973	0.9903
Shuttle-c0-vs-c4	1	1	1	1	1	1	1	1
Shuttle-c2-vs-c4	1	1	0.95	1	1	1	1	0.95
Led7digit-0-2-4-5-6-7-8-9_vs_1	**1**	0.9841	0.9431	0.9556	0.9862	0.9512	0.9473	0.9862
Vehicle1	**0.8483**	0.7613	0.7601	0.8214	0.8006	0.8031	0.6783	0.7757
Winequality-white-3_vs_7	0.8375	**0.8750**	0.8212	0.8464	0.8243	0.7815	0.6035	0.8334
Page-blocks-1-3_vs_4	0.9846	0.9733	0.9562	0.9633	0.9927	0.9366	**0.9966**	0.9927
Paw02a-600-5-30-BI	**0.9131**	0.8385	0.8162	0.8955	0.8116	0.8725	0.8927	0.8116
Yeast-1-4-5-8_vs_7	**0.8083**	0.7055	0.7672	0.7334	0.6274	0.8043	0.7731	0.725
Yeast5	**0.9937**	0.9873	0.9819	0.9781	0.9613	0.9834	0.9873	0.9771
Average	**0.9658**	0.9355	0.9227	0.9458	0.9141	0.9384	0.9290	0.9243

TABLE 6 F-measure of comparison methods

Dataset	UACB	RUS	CBUS	CBIS	RBU	HUE	TU	UA-KF
Abalone9-18	**0.9018**	0.775	0.6082	0.7032	0.7033	0.7623	0.7226	0.7244
Ecoli-0_vs_1	**1**	0.9736	0.9874	0.9213	0.9351	0.9937	0.9934	0.9733
Ecoli1	**0.9867**	0.8439	0.8769	0.8423	0.8001	0.7564	0.9349	0.9144
Car-good	**0.9862**	0.9396	0.9851	0.9506	0.8345	0.9725	0.9664	0.8345
Cleveland-0_vs_4	**1**	0.8099	0.9718	0.9714	0.7421	0.9600	0.8914	0.7421
Dermatology-6	**1**	0.9382	0.9536	0.9627	0.9199	0.9218	0.9778	0.9199
Glass-0-1-2-3_vs_4-5-6	**0.9694**	0.9318	0.8426	0.9442	0.8582	0.8708	0.9135	0.8582
Iris0	1	1	1	1	1	1	1	1
Kr-vs-k-zero-one_vs_draw	**0.9837**	0.8884	0.9532	0.9242	0.7359	0.9528	0.9258	0.7359
New-thyroid2	0.9867	0.9866	0.9867	0.9372	0.8928	**0.9904**	0.9864	0.9359
Page-blocks0	0.9121	0.8879	0.8791	**0.9258**	0.7568	0.7641	0.9026	0.8127
Segment0	0.9883	0.9845	0.9756	0.9774	0.9774	**0.9919**	0.9857	0.9802
Shuttle-c0-vs-c4	1	1	1	1	0.9957	1	1	1
Shuttle-c2-vs-c4	1	1	0.9332	1	1	0.9324	1	0.83
Led7digit-0-2-4-5-6-7-8-9_vs_1	**0.9846**	0.9346	0.8602	0.8625	0.7652	0.8863	0.9061	0.7652
Vehicle1	0.7413	0.7075	0.6637	**0.7543**	0.6552	0.6923	0.5903	0.6651
Winequality-white-3_vs_7	0.7433	**0.8099**	0.7276	0.7859	0.6874	0.7032	0.5576	0.76
Page-blocks-1-3_vs_4	**0.9625**	0.9511	0.9173	0.9442	0.8918	0.9512	0.9484	0.8918
Paw02a-600-5-30-BI	**0.9381**	0.7796	0.7488	0.8842	0.7029	0.7926	0.8513	0.7029
Yeast-1-4-5-8_vs_7	**0.7266**	0.6104	0.6204	0.6553	0.5848	0.6903	0.6534	0.6588
Yeast5	**0.9882**	0.966	0.9567	0.9346	0.9137	0.9751	0.8359	0.9671
Average	**0.9428**	0.8914	0.8785	0.8991	0.8263	0.8838	0.8830	0.8415

subspace. In order not to lose all other majority class information, select other majority samples to form the final majority sample set. In other word, both CBIS and HUE retain the structure of majority class. In this way, samples that are far away from the separating hyperplane that characterize the majority class structure will be selected. But, from Theorem 1, we know that samples far away from the separation hyperplane contribute little to the final construction of the classification model. UACB does not have such a problem, so its results are better than CBIS and HUE. RBU is based on the mutual class potential. It calculates the mutual class relationship between majority class sample and minority class data through the Gaussian kernel function. However, the samples of the minority class are fixed and the number is small, so the range of samples selected from the majority class is determined. Therefore, the training set samples formed in this way lose their diversity. That is to say, the sampling of the majority class relying on the characteristics of the minority class. It will lose the characteristics of the majority class, which will affect the final classification results. TU is a deep learning method. Its results obtained by using the network architecture and parameters provided by the method proponent are worse than UACB. We believe that it can use better network structure and parameters to get better results. But there is no good way to guide how to find them.

Based on the above analysis of the results, we can clearly see that for most datasets, guiding sampling of majority class with sapce compresson and weighted random sampling will improve the AUC and F-measure results in classification task. At the same time, it can be seen from Tables 5 and 6 that we can actively discard some small contribution data samples when we process data with the imbalanced undersampling method.

5 | CONCLUSION

The imbalance problem exists in applications in many fields and affects the results of classification. It is very promising to study it. This work proposes a new undersampling method named UACB to deal with imbalance problems. The proposed method is mainly to perform space compression in the vertical direction of the separation hyperplane, and then use weighted sampling combined with ensemble learning to make the sampled objects spread more widely near the separating hyperplane. We conducted experiments on 21 datasets and compared the proposed method with

7 competitive methods. The experimental results indicate that the proposed method outperforms other alternatives in most cases in terms of AUC and F-measure. Through experiments, we have two understandings. Firstly, that denoising plays a role in the imbalance classification problem, but the effect in UACB is mainly obtained by space compression and ensemble learning. Secondly, we believe that the success of UACB lies in that the sampling near the classification boundary makes the classification model approach the ideal state through the boosting of the many weak classifier. Therefore, an effective research direction in the future is how to find an effective method to make the model more close to the ideal state in the imbalanced classification problem.

ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Education of Humanities and Social Science project, China (Grant no.17YJAZH032). And the work was supported in part by the National Natural Science Foundation of China (Grant 62107013). And the work was supported part by Hunan Provincial Department of Education Innovation Platform Open Fund Project, China (Grant no.20K050). And the work was supported part by the Key Project of Hunan Provincial Education Department, China (Grant no.19A172).

CONFLICT OF INTEREST

No potential conflict of interest was reported by the authors.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in datasets from keel at https://sci2s.ugr.es/keel/imbalanced.php.

ENDNOTE

*https://sci2s.ugr.es/keel/imbalanced.php

ORCID

Lei Jiang https://orcid.org/0000-0002-5654-7748
Jianxun Liu https://orcid.org/0000-0003-0722-152X

REFERENCES

1. Niu K, Zhang Z, Liu Y, Li R. Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. *Inf Sci*. 2020;536:120-134.
2. Mahajan S, Nayyar A, Raina A, Singh SJ, Vashishtha A, Pandit AK. A Gaussian process-based approach toward credit risk modeling using stationary activations. *Concurr Comput Pract Exp*. 2022;34(5):e6692.
3. Cao L, Shen H. CSS: Handling imbalanced data by improved clustering with stratified sampling. *Concurr Comput Pract Exp*. 2022;34(2):e6071.
4. Cano A, Zafra A, Ventura S. Weighted data gravitation classification for standard and imbalanced data. *IEEE Trans Cybern*. 2013;43(6):1672-1687.
5. Codetta-Raiteri D, Portinale L. Dynamic Bayesian networks for fault detection, identification, and recovery in autonomous spacecraft. *IEEE Trans Syst Man Cybern Syst*. 2014;45(1):13-24.
6. Tang Y, Zhang YQ, Chawla NV, Krasser S. SVMs modeling for highly imbalanced classification. *IEEE Trans Syst Man Cybern Part B (Cybern)*. 2008;39(1):281-288.
7. Zhang X, Hu BG. A new strategy of cost-free learning in the class imbalance problem. *IEEE Trans Knowl Data Eng*. 2014;26(12):2872-2885.
8. Feng Y, Zhou M, Tong X. Imbalanced classification: a paradigm-based review. *Stat Anal Data Min ASA Data Sci J*. 2021;14(5):383-406.
9. Tarekegn AN, Giacobini M, Michalak K. A review of methods for imbalanced multi-label classification. *Pattern Recognit*. 2021;118:107965.
10. Kang Q, Chen X, Li S, Zhou M. A noise-filtered under-sampling scheme for imbalanced classification. *IEEE Trans Cybern*. 2016;47(12):4263-4274.
11. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern Part C (Appl Rev)*. 2011;42(4):463-484.
12. Khorshidi HA, Aickelin U. Constructing classifiers for imbalanced data using diversity optimisation. *Inf Sci*. 2021;565:1-16.
13. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng*. 2009;21(9):1263-1284.
14. Mishra S. Handling imbalanced data: SMOTE versus random undersampling. *Int. Res. J. Eng. Technol*. 2017;4(8):317-320.
15. Douzas G, Bacao F, Last F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf Sci*. 2018;465:1-20.
16. Tsai CF, Lin WC, Hu YH, Yao GT. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Inf Sci*. 2019;477:47-54.
17. Le HL, Landa-Silva D, Galar M, Garcia S, Triguero I. EUSC: a clustering-based surrogate model to accelerate evolutionary undersampling in imbalanced classification. *Appl Soft Comput*. 2021;101:107033.
18. Lin WC, Tsai CF, Hu YH, Jhang JS. Clustering-based undersampling in class-imbalanced data. *Inf Sci*. 2017;409:17-26.
19. Ng WW, Hu J, Yeung DS, Yin S, Roli F. Diversified sensitivity-based undersampling for imbalance classification problems. *IEEE Trans Cybern*. 2014;45(11):2402-2412.
20. Dhar S, Cherkassky V. Development and evaluation of cost-sensitive universum-SVM. *IEEE Trans Cybern*. 2014;45(4):806-818.
21. Elkan C. The foundations of cost-sensitive learning. Paper presented at: 17 of International joint conference on artificial intelligence, Lawrence Erlbaum Associates Ltd; 2001:973-978.
22. Van Hulse J, Khoshgoftaar TM, Napolitano A. A novel noise filtering algorithm for imbalanced data. Paper presented at: 2010 Ninth International Conference on Machine Learning and Applications, IEEE; 2010:9-14.
23. Sáez JA, Luengo J, Stefanowski J, Herrera F. SMOTE–IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inf Sci*. 2015;291:184-203.

24. Han H, Wang WY, Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Paper presented at: International Conference on Intelligent Computing, Springer; 2005:878-887.
25. Nguyen HM, Cooper EW, Kamei K. Borderline over-sampling for imbalanced data classification. Paper presented at: 2009 of Proceedings: Fifth International Workshop on Computational Intelligence & Applications, IEEE SMC Hiroshima Chapter; 2009:24-29.
26. Krawczyk B. Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell*. 2016;5(4):221-232.
27. Wang K, An J, Yu Z, Yin X, Ma C. Kernel local outlier factor-based fuzzy support vector machine for imbalanced classification. *Concurr Comput Pract Exp*. 2021;33(13):e6235. doi:10.1002/cpe.6235
28. Ding H, Chen L, Dong L, Fu Z, Cui X. Imbalanced data classification: a KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener Comput Syst*. 2022;131:240-254.
29. Vuttipittayamongkol P, Elyan E, Petrovski A. On the class overlap problem in imbalanced data classification. *Knowl Based Syst*. 2021;212:106631.
30. Korkmaz S, Şahman MA, Cinar AC, Kaya E. Boosting the oversampling methods based on differential evolution strategies for imbalanced learning. *Appl Soft Comput*. 2021;112:107787.
31. Branco P, Torgo L, Ribeiro RP. A survey of predictive modeling on imbalanced domains. *ACM Comput Surv (CSUR)*. 2016;49(2):1-50.
32. Ludvigsen J. *Handling Imbalanced Data Classification with Variational Autoencoding and Random Under-Sampling Boosting*. www.diva-portal.org; 2020.
33. Nguyen TV, Huy TQ, Nguyen VD, Thu NT, Anh GQ, Tan TD. Hybrid random under-sampling approach in mri compressed sensing. *Intelligent Computing in Engineering*. Springer; 2020:943-950.
34. Wang K, Zhang J, Li D, Zhang X, Guo T. Adaptive affinity propagation clustering. *arXiv preprint arXiv:0805.1096*; 2008.
35. Ng W, Xu S, Zhang J, Tian X, Rong T, Kwong S. Hashing-based undersampling ensemble for imbalanced pattern classification problems. *IEEE Trans Cybern*. 2022;52(2):1269-1279.
36. Koziarski M. Radial-based undersampling for imbalanced data classification. *Pattern Recognit*. 2020;102:107262.
37. Krawczyk B, Koziarski M, Woźniak M. Radial-based oversampling for multiclass imbalanced data classification. *IEEE Trans Neural Netw Learn Syst*. 2019;31(8):2818-2831.
38. Peng M, Zhang Q, Xing X, et al. Trainable undersampling for class-imbalance learning. Paper presented at: 33 of Proceedings of the AAAI Conference on Artificial Intelligence; 2019:4707-4714.
39. Kulkarni SR, Posner SE. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Trans Inf Theory*. 1995;41(4):1028-1039.
40. Vural E, Guillemot C. A study of the classification of low-dimensional data with supervised manifold learning. *J. Mach. Learn. Res.* 2017;18(1):5741-5795.
41. Zhang YP, Zhang LN, Wang YC. Cluster-based majority under-sampling approaches for class imbalance learning. Paper presented at: 2010 2nd IEEE International Conference on Information and Financial Engineering, IEEE; 2010:400-404.
42. Aha DW, Kibler DF, Albert MK. Instance-based learning algorithms. *Mach Learn*. 1991;6(1):37-66.
43. Gong Y, Lazebnik S, Gordo A, Perronnin F. Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans Pattern Anal Mach Intell*. 2012;35(12):2916-2929.
44. Bradley A. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit*. 1997;30(7):1145-1159.
45. Powers DM. Evaluation: from precision, recall, and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*; 2011.

How to cite this article: Jiang L, Yuan P, Liao J, Zhang Q, Liu J, Li K. Undersampling of approaching the classification boundary for imbalance problem. *Concurrency Computat Pract Exper*. 2023;35(6):e7586. doi: 10.1002/cpe.7586