Decision Boundary Computation-based Over-sampling for Imbalance Learning

Yi Sun, Lijun Cai, Junlin Xu, Bo Liao, and Wen Zhu

***Abstract*—Over-sampling is a very effective method to solve the imbalanced problem by generating new synthetic samples for the minority class. But rare over-sampling methods focus on the borderline between classes and only use the linear-interpolation between boundary samples to fill the decision boundary, so not take full use of information in the decision boundary at all. To fill this gap, one novel method named Decision Boundary Computation-based Oversampling is proposed. Firsts, the novel method treats surrounding areas of both boundary majority and minority samples as the decision boundary. Then, compute it's area belonging to majority class and treat the remained one as the area belonging to the minority class. Thus, the novel method greatly enhances the full use of boundary information and implicitly complements the nature insufficiency of information of minority class at the same time. Finally, new synthetic samples are generated in the partition of decision boundary of minority class. Extensive experiments indicate the good performance of proposed method when compared with other state-of-art methods.**

***Index Terms*—Imbalance learning, decision boundary, area partition, over-sampling.**

I. Introduction

CLASS imbalanced problem, served as one of the most challenging problems in data mining [1] and machine learning [2], appears in many real-world applications like image classification [3], [4], credit fraud detection [5], stream data mining [6], face recognition [7] and so on. Instead of the multi-class classification [8] [9], we focus on the binary classification of imbalanced problem that one class with smaller number of samples is called as the minority class and another one as the majority class. Generally in one imbalanced problem, the classifier tends to bias towards the recognition of majority samples. For example, given 10 minority samples and 90 majority samples, the classifier can achieve 90% accuracy when classifying all samples as the majority class. While many real-world applications care more about the recognition of rare minority samples, especially for some secure domains. So, learning from imbalanced data is a long-standing and significant challenge for machine learning [10].

To deal with the imbalanced problem, several techniques have been reported and proved to be efficient that are mainly involving the algorithm-level strategy [11]–[13] and the data-level strategy [14]–[16]. First for the algorithm-level strategy, the cost-sensitive learning [17], [18] and the ensemble learning [19], [20] are two commonly used techniques to cope with the imbalanced problem. Besides, the algorithm-level strategy also includes some other techniques like hyperplane shift [21], kernel perturbation [22] and multiobjective optimization [23]. Then, for the data-level strategy, the minority over-sampling [24], [25] and majority under-sampling [26], [27] are two commonly used techniques to cope with the imbalanced problem. The minority over-sampling technique balances the number of samples between classes by generating new synthetic samples for the minority class [28], [29]. Inversely, the under-sampling technique generally removes majority samples [30] or noisy minority samples [27] which may lead to loss of informative samples [31]. Thus, in this paper, we focus on the over-sampling technique for it's characteristic that does not miss any original information.

The over-sampling technique mainly includes the linear-interpolation [32]–[34] and non-linear or structure-preserving interpolation methods [35]–[37]. For example, synthetic minority over-sampling technique(SMOTE) [32] generates one new synthetic sample by the linear interpolation between the target minority sample and it's random one of k-nearest minority neighbours. On the basis of SMOTE, the linear-interpolation method also involves in the borderline minority over-sampling [38], hard-to-learn minority over-sampling [33], [39] and kernel over-sampling [14], [25] techniques. Contrary to those linear-interpolation methods, structure-preserving interpolation method first estimates the corresponding structure of minority class, then generates new samples to maintain or preserve this estimated structure. For example, in methods of [35] and [36], they use the covariance of minority class to generate new synthetic samples.

However, only several over-sampling methods [33], [38], [39] focus on the borderline between classes. And those borderline-related over-sampling methods only use the linear-interpolation between boundary samples to fill the decision boundary, so not fully take use of information in the decision boundary at all. To fill this gap, we, therefore, propose one Decision Boundary Computation-based Over-sampling (DBO) method to fill this gap. From the intuitive observation, the design boundary not only includes individual samples, but also their surrounding areas. Thus, to take full use of boundary information, we first compute the decision boundary on the basis of the boundary majority and minority samples and their surrounding areas; and compute the partition belonging to the majority class on the basis of the boundary majority

This work was supported by the National Key R&D Program of China (No.2020YFB2104400).

Y. Sun, L. Cai and J. Xu are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China. (e-mail: id.yisun@gmail.com; ljcai@hnu.edu.cn; 18273118685@163.com).

B. Liao and W. Zhu are with the Key Laboratory of Computational Science and Application of Hainan Province, Hainan Normal University, Haikou 571158, China. (e-mail: dragonbw@163.com; syzhuwen@163.com).

Corresponding authors:Lijun Cai and Junlin Xu.

samples and their surrounding areas. Then, we obtain the partition belonging to the minority class by subtracting the partition of majority class from the decision boundary. For convenience, we call the decision boundary as the decision boundary area, the partition belonging to the minority class as the boundary minority area and the partition belonging to the majority class as the boundary majority area. Finally, we generate new synthetic samples in the boundary minority area to cope with the imbalanced problem.

Contributions are summarized as:

1) We innovatively attempt to compute the decision boundary area between classes and divide it into different boundary areas corresponding to different classes, providing a new view for many classification tasks.
2) We propose one novel minority over-sampling method that generates synthetic samples in the boundary minority area for class imbalance problem.
3) We take full use of information on the decision boundary by simultaneously considering boundary individual samples and their surrounding areas.
4) The subtraction of boundary majority area can well avoid synthetic samples deeply rooting into the majority area and make our over-sampling method robust to several outliers at the same time.

The rest of paper is organized as follows. Sections II reviews related literature. Section III presents the decision boundary computation-based over-sampling (DBO) method. Experimental results and discussion are respectively prepared in Section IV and V. In Sections VI, the conclusion is included.

II. Related Work and motivation

A. Related Work

First for linear-interpolation methods, Han et al. [38] propose B-SMOTE1 and B-SMOTE2 to only generate synthetic samples for borderline minority samples. Where one minority sample is considered as the borderline one when the number of neighbouring majority is larger than the minority. For example, denoting the number of majority samples in m nearest neighbours as m', one minority sample is determined as the borderline one when $m/2 \leq m' < m$. For those borderline minority samples, B-SMOTE1 searches k nearest neighbours from the minority class for linear-interpolation to generate new synthetic samples; specially, B-SMOTE2 searches k nearest neighbours from both classes for linear-interpolation. To make subtler and better distinctions between different borderline minority samples, He et al. [33] propose ADASYN to assign different borderline minority with different weights according to the ratio m'/m. Where the higher weight means higher level of difficulty in learning, and calling those borderline minority samples as hard-to-learn minority samples. Slightly different, Barua et al. [39] propose MWMOTE to assign different borderline minority with different weights according to their Euclidean distance from the nearest majority samples. Specially, MWMOTE does not use k nearest neighbours, but clusters for linear-interpolation. For example, for one hard-to-learn minority sample, MWMOTE searches one random minority sample from the same cluster for linear-interpolation.

And for non-linear or structure-preserving interpolation methods, Sharma et al. [40] propose SWIM to generate synthetic samples with the same Mahalanobis distance from the majority class mean. Besides, Cao et al. [35] propose INOS to generate synthetic samples in the whole data space with the corresponding covariance structure of minority class and subsequently cleans the synthetic data that nearer to majority samples. And Xie et al. [41] propose GDO to generate synthetic samples to follow a Gaussian distribution model.

As seen in Fig. 1. (a), B-SMOTE1, ADASYN and MWMOTE may generate one new sample S1 in the line between point A and B; B-SMOTE2 generate one new sample S2 in the line between point A and D. As seen in Fig. 1. (b), SWIM may generate new samples S1 and S2 in the curve with the same Mahalanobis distance from the majority class mean. As seen in Fig. 1. (c), INOS may generate new samples S1 and S2 in the global minority area with corresponding covariance structure of minority class and nearer to the minority class at the same time. As seen in Fig. 1. (d), GDO may generate new samples S1 and S2 to follow a Gaussian distribution model of point A.

Obviously, linear-interpolation methods only use individual minority and majority samples for linear-interpolation and not consider their surrounding areas at all. And non-linear or structure-preserving interpolation methods do not consider the borderline, and ignore surrounding areas of boundary majority samples.

B. Motivation

From the intuitive observation, boundary individual samples and their surrounding areas together constitute the decision boundary area as seen in Fig. 2. (c) in $\mathbb{R}^2$. Where the green area means the boundary majority area and the blue area means the boundary minority area, and the green and blue area together constitute the decision boundary area. So we ask whether generating synthetic samples in the boundary minority area can help much for the classification of imbalanced data. However, two problems make it difficult to directly compute the boundary minority area. On the one hand, rare number of boundary minority samples lead to the missing information in the boundary minority area. On the other hand, complex distributions of data make it impossible to directly compute the integral or continuous boundary minority area.

To solve the first problem, we borrow information from boundary majority samples. Owing to enough boundary majority samples, we first combine them with rare boundary minority samples to compute the decision boundary area, then only use them to compute the boundary majority area; finally, we can obtain the boundary minority area by subtracting the boundary majority area from the decision boundary area. To solve the second problem, we do not directly compute the integral boundary area. In other words, we first compute a series of local boundary areas, then integrate those local boundary areas together to approximately represent the integral boundary area. Details of the proposed method are described below.

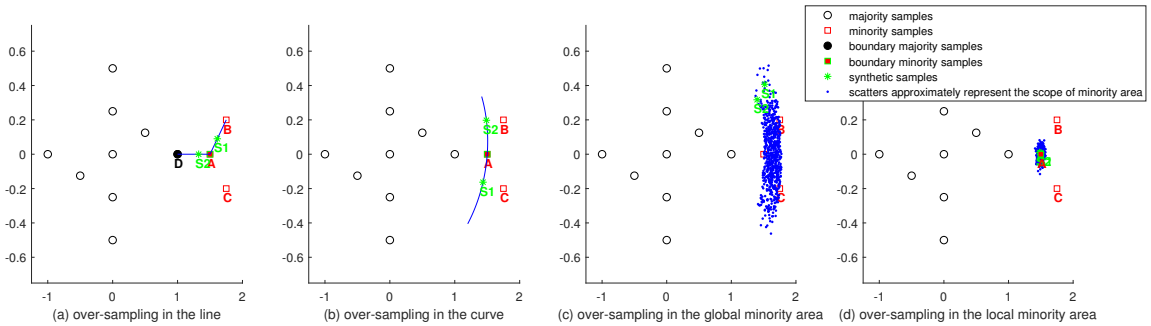

Fig. 1. Strategies of current over-sampling works. (a) over-sampling in the line, (b) over-sampling in the curve, (c) over-sampling in the global minority area, (d) over-sampling in the local minority area. In (c) and (d), the complete area is difficult to draw, thus we use scatters to approximately represent the scope of corresponding minority area.

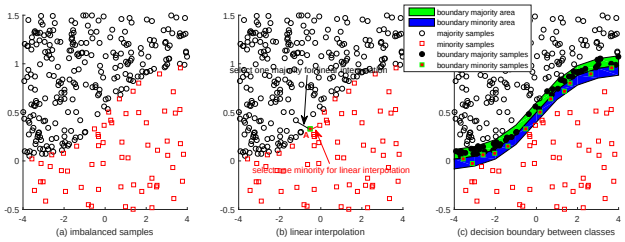

Fig. 2. Motivation of proposed method. (a) original imbalanced samples; (b) linear interpolation; (c) decision boundary between classes.

C. Preliminary knowledge of area computation

In this section, we introduce how to compute the corresponding local area when give a group of samples. For example, when given one group of samples $\{x_1, x_2, x_3, ..., x_m\}$, we compute it's corresponding area as the set:

$$S = \{x|\ (x-\bar{x})^T Q^{-1}(x-\bar{x}) \leq 1\} \tag{1}$$

where Q is one symmetric and positive definite matrix; and $\bar{x}$ is the center of this group:

$$\bar{x} = \frac{1}{m} \times \sum_{j=1}^{m} x_j \tag{2}$$

Obviously, this area is just one ellipsoid where Q defines how far it extends in each direction from $\bar{x}$. To facilitate understanding, we compute:

$$Q^{-1} = (\alpha * U)^{-1} \tag{3}$$

where U is the covariance matrix of this group; α is one predefined length for this covariance matrix U. And the inverse matrix of covariance matrix can be obtained by the eigen decomposition of the covariance matrix U:

$$U^{-1} = (VEV^T)^{-1} = V^T E^{-1} V \tag{4}$$

where E is one diagonal matrix with diagonal elements as $(\lambda_1, \lambda_2, ..., \lambda_n)$ (supposing no zero eigen value existed).

As seen in Fig.3. (a), the black circle denotes one group of samples. Fig.3. (b) plots the area of this group when assigning

$$\alpha = \alpha_A = (x_A - \bar{x})^T U^{-1}(x_A - \bar{x}) \tag{5}$$

where point A is one sample in this group, O is the center of this group; x_A is one 2-D coordinate of vector for point A. And Fig.3. (c) plots the area of this group when assigning $\alpha = 1.5 \times \alpha_A$. Obviously, A is one point on the surface of this area when assigning $\alpha = \alpha_A$; A is one interior point when assigning $\alpha > \alpha_A$ (for example $\alpha = 1.5 \times \alpha_A$). To describe conveniently, we mean α_A as the length of A (on covariance matrix U).

Thus, Eq. 1 can be transformed as:

$$S = \{x|\ (x-\bar{x})^T U^{-1}(x-\bar{x}) \leq \alpha\} \tag{6}$$

To this end, we can compute the area of one group of samples when assigning it's corresponding covariance matrix with one length α. In other words, the certain area depends on the selection of the corresponding group of samples and the assignment of length α.

III. Decision Boundary Computation-Based Over-Sampling

Intuitively, in the binary classification task, one side of the decision boundary area belongs to the majority class, and anther side belongs to the minority class. Of course, directly estimating the global decision boundary area is difficult. Thus, from the perspective of the local area, we estimate one ellipsoid that covering a local area belong to the decision boundary, then divide it into two partitions. Similarly, directly dividing this ellipsoid is difficult, thus we first estimate another one ellipsoid that covers the corresponding local area belong to majority class, then treat the intersection of two ellipsoids belong to the majority class and the remained area belong to the minority class. For convenience, we also call the ellipsoid that covers the local area belong to the decision boundary as the local decision boundary area or the ellipsoid of decision boundary area; and the ellipsoid that covers the local area belong to the majority class as the local boundary majority area or the ellipsoid of boundary majority area. As seen in Fig.4, the general procedure chart is plotted. Obviously, after subtracting the ellipsoid of local boundary majority area from the ellipsoid of local decision boundary area, the remained area is served as the local boundary minority area. In the end, we can integrate all those local boundary minority areas together to approximately represent the integral boundary minority

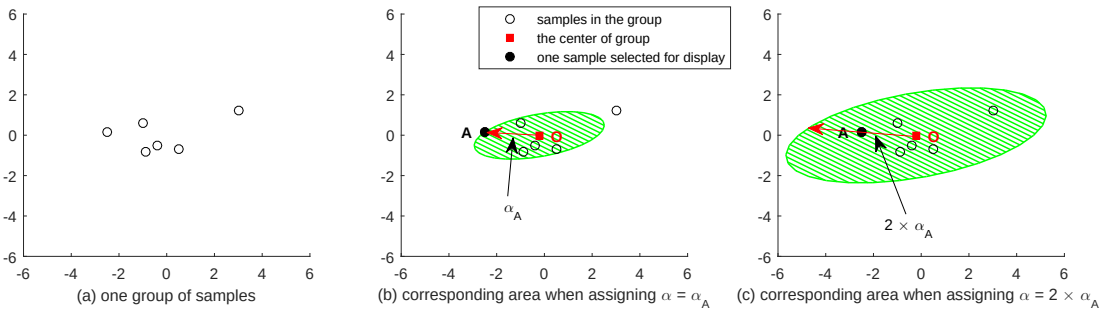

Fig. 3. Area computation. (a) a group of samples. (b) corresponding area when assigning $\alpha = \alpha_A$; (c) corresponding area when assigning $\alpha = 2 \times \alpha_A$; the red arrow denotes the predefined length α for the covariance matrix of the group.

area.

To this end, the proposed method mainly involves into three steps respectively as the computation of the ellipsoid of decision boundary area, the computation of the ellipsoid of boundary majority area and the generation of synthetic samples in the boundary minority area.

A. decision boundary area

1) a group of samples: In this subsection, we first compute boundary minority samples, then select the group of samples for each boundary minority sample. As seen in Fig.2, boundary minority samples are nearer to the majority class than other non-boundary minority samples. Thus, for each majority samples, we first compute it's nearest minority one and add it to

$$B_1 = \{w_1, w_2, ..., w_i, ..., w_n\} \tag{7}$$

where n is the number of boundary minority samples and w_i is the i-th boundary minority sample in B_1. Then, for each boundary minority sample w in B_1, compute it's k nearest majority samples:

$$B_2 = \{z_1, z_2, .., z_k\} \tag{8}$$

where B_2 includes k nearest majority samples for w.

Finally, select the group of samples as.

$$G = \{w, h_1, h_2, ..., h_k\} \tag{9}$$

$$h_j = \frac{z_j + w}{2} \tag{10}$$

Obviously, except w, G_i just includes mean points between w and it's k nearest majority samples.

Suppose $h_j = (1 - ratio) \times w + ratio \times z_j$, as seen in Fig.5, the reason why we choose mean points instead of k nearest majority samples is plotted and discussed. Obviously as seen in Fig.5. (d), if selecting k nearest majority samples, the local decision boundary area tends to cover the whole local boundary majority area and the remained boundary minority area will deeply root into the majority area. Details and deep explanation are seen in the next subsection.

2) corresponding length: In this subsection, we assign one length to the corresponding covariance matrix of preferentially selected group. Firsts, compute the center and covariance matrix of group G as $\bar{x}^{DB}$ and U^{DB}. Then, compute the length of w:

$$\alpha_1 = (w - \bar{x}^{DB})^T (U^{DB})^{-1} (w - \bar{x}^{DB}) \tag{11}$$

To cover both w and it's surrounding area with enough size, we compute the corresponding length of the local decision boundary area:

$$\alpha_2 = 2 * \alpha_1 \tag{12}$$

Since the group of samples and corresponding length are respectively selected and assigned, we compute the local decision boundary area:

$$S^{DB} = \{x |\ (x - \bar{x}^{DB})^T (U^{DB})^{-1} (x - \bar{x}^{DB}) \leq \alpha_2\} \tag{13}$$

where $(U_i^{DB})^{-1}$ is the inverse matrix which can be obtained by the eigen decomposition in Eq. 4.

As seen in Fig.4. (b) and (d), the local decision boundary area does not deeply root in the majority area and is of enough size at the same time. In detail, for the first goal, as seen in Fig.5. (b), we zoom out the local decision boundary area by setting $ratio = 0.5$. Obviously in Fig.5. (a), smaller $ratio$ makes the local decision boundary area smaller like $ratio = 0.25$. As seen in Fig.5. (c) or (d), larger $ratio$ makes the local decision boundary area larger like $ratio = 0.75$ or 1. For the second goal, as seen in Fig.4. (b) and Fig.5. (b), we zoom in the local decision boundary area by setting $\alpha = 2 \times \alpha_B$ (where point B is the target boundary minority sample) to cover both point B and it's near local minority area. In general, we first double down this area, then double up this area to simultaneously meet above two goals. After reviewing the whole method, those two zooming operations can be further understood.

B. Boundary majority area

Similarly, we first select a group of samples, then assign one length to the corresponding covariance matrix. Different from the local decision boundary area, for each boundary minority sample w, we directly select B_2 in Eq. 8 as the group.

Firsts, compute the center and covariance matrix of the

group B_2 as $\bar{x}^{MAJ}$ and U^{MAJ}. Then, compute the lengths of w and z_1 (z_1 is the nearest majority to w):

$$\alpha_3 = (w - \bar{x}^{MAJ})^T (U^{MAJ})^{-1} (w - \bar{x}^{MAJ}) \tag{14}$$

$$\alpha_4 = (z_1 - \bar{x}^{MAJ})^T (U^{MAJ})^{-1} (z_1 - \bar{x}^{MAJ}) \tag{15}$$

As seen in Fig. 6. (a-c), we respectively set α_5 as α_4, α_3 and $0.5*(\alpha_4+\alpha_3)$. Obviously, when setting $\alpha_5 = 0.5*(\alpha_4+\alpha_3)$, almost the half of the ellipsoid of local decision boundary area belongs to the majority class that intuitively cutting the corresponding ellipsoid into nearly two halves. As seen in Fig. 6. (d-e), we respectively set α_5 as $\alpha_4 + 0.5 * (\alpha_3 - \alpha_4)$ (or $0.5 * (\alpha_4 + \alpha_3)$) and $\alpha_4 + 0.5 * |\alpha_3 - \alpha_4|$. Obviously, when setting $\alpha_5 = \alpha_4 + 0.5 * |\alpha_3 - \alpha_4|$, the remained minority area is empty that robust to several outliers or over-lapped ones. Thus, we compute the corresponding length of the local boundary majority area as:

$$\alpha_5 = \alpha_4 + 0.5 \times |\alpha_3 - \alpha_4| \tag{16}$$

Since the group of samples and corresponding length are respectively selected and assigned, we compute the local boundary majority area as:

$$S^{MAJ} = \{x |\ (x - \bar{x}^{MAJ})^T (U^{MAJ})^{-1} (x - \bar{x}^{MAJ}) \leq \alpha_5\} \tag{17}$$

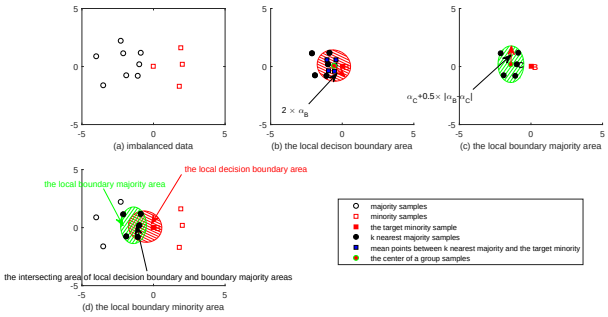

Fig. 4. The computation of the local boundary minority area. (a) imbalanced data, (b) the ellipsoid of local decision boundary area, (c) the ellipsoid of local boundary majority area, (d) the local boundary minority area. Since the ellipsoid of local decision boundary area and the ellipsoid of local boundary majority area are computed, we obtain the local minority area by subtracting the ellipsoid of local boundary majority area.

C. Generation of synthetic samples

In this section, we first give the local boundary minority area, then give the final integral boundary areas and finally generate new synthetic samples. Firstly, since the local decision boundary and boundary majority areas are computed, we obtain the local boundary minority area as:

$$S^{MIN} = S^{DB} - S^{DB} \cap S^{MAJ} \tag{18}$$

As seen in Fig.4. (d), we subtract the intersecting area from the local decision boundary area and the remained area denotes the local boundary minority area.

Then, we respectively integrate those local boundary areas

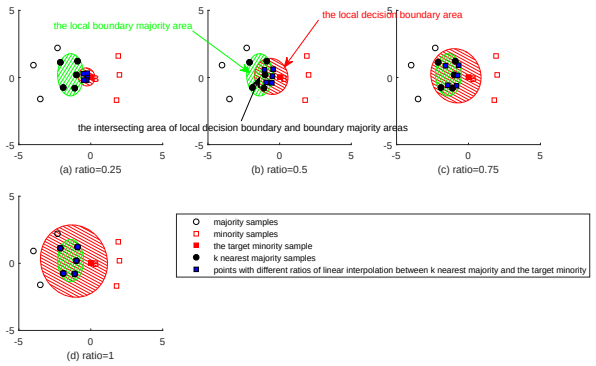

Fig. 5. Different selections of the group of samples from the local decision boundary area. (a) $ratio = 0.25$, (b) $ratio = 0.5$, (c) $ratio = 0.75$, (d) $ratio = 1$.

together to approximately represent the corresponding integral boundary areas.

$$S_{IDB} = \bigcup_{i=1}^{n} S_i^{DB} \tag{19}$$

$$S_{IMAJ} = \bigcup_{i=1}^{n} S_i^{Maj} \tag{20}$$

$$S_{IMIN} = \bigcup_{i=1}^{n} S_i^{MIN} \tag{21}$$

where n denotes the number of boundary minority samples in B_1 in Eq. 7; S_{IDB} is the integral decision boundary area, S_{IMAJ} is the integral boundary majority area and S_{IMIN} is the integral boundary minority area.

Finally, we generate new synthetic samples in the boundary minority area. Of course, direct generation in the boundary minority area is impossible. As seen in Eq. 18, we only own the information of the local decision boundary area and boundary majority area. Thus, we first generate the synthetic sample in the local decision boundary area, then judge whether it falls in the boundary majority area.

In detail, we first use the Gaussian distribution ($G(\mu = 1, \sigma = 1)$) to generate one random value, then use it to obtain a random length in [0,1] as:

$$l = (1 - |G(\mu, \sigma)|) \tag{22}$$

Next, randomly generate one normalized direction satisfying

$$||d|| = 1 \tag{23}$$

Compute the temporary sample:

$$t = V^{DB} (E^{DB})^{1/2} (\alpha_2 * l * d) + \bar{x}^{DB} \tag{24}$$

where V^{DB} and E^{DB} are components in the eigen decomposition of covariance matrix U^{DB} in Eq.4.

In the last step, judge whether t falls in S^{Maj} by Eq.17. If not, record t as the new synthetic sample; if is, re-generate one temporary sample t again.

In experience, we restrict the time of re-generation as 100 for the possibility that the local boundary majority area would cover the whole local decision boundary area. The algorithm of DBO is seen in Algorithm S1.

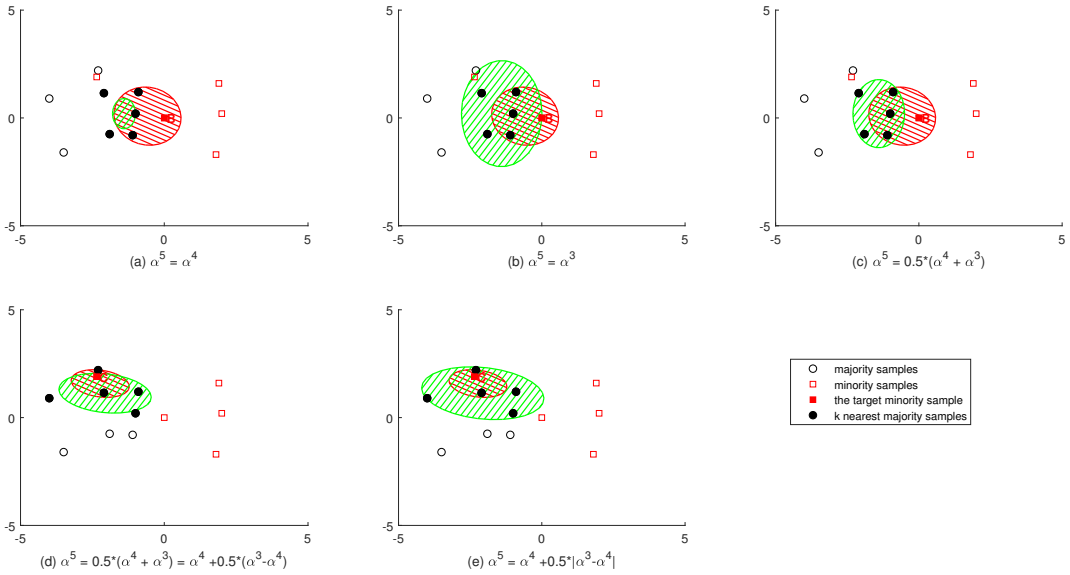

Fig. 6. Different selections of α_5. (a) $\alpha^5 = \alpha^4$; (b) $\alpha^5 = \alpha^3$; (c) $\alpha^5 = 0.5 * (\alpha^4 + \alpha^3)$; (d) $\alpha^5 = 0.5 * (\alpha^4 + \alpha^3) = \alpha^4 + 0.5 * (\alpha^3 - \alpha^4)$; (e) $\alpha^5 = \alpha^4 + 0.5 * |\alpha^3 - \alpha^4|$.

IV. Experimental results

In this section, we pick three borderline-related methods as B-SMOTE2 [38], ADASYN [33] and MWMOTE [39], and other three state-of-the-art methods as INOS [35], SWIM [40] and GDO [41], for comparisons. Firsts, we generate synthetic samples for those methods in 2D emulational datasets for visualization. Then, we test all methods on real-world benchmark datasets that collected from the UCI machine learning repository [42] and [43]; and carry statistical hypothesis tests for those methods. Finally, we analyse those over-sampling methods.

A. Synthetic data in 2D space

As seen in Fig. 7, we generate synthetic samples for above picked methods in three 2-D datasets, respectively as Circle dataset, Triangle dataset and lappedCircle dataset. Specially, for Triangle dataset and lappedCircle dataset, we add them with several outliers. In a row of pictures, first plot the original data, then plot synthetic samples of each methods. Where black denotes majority samples and red denotes minority samples. For each method, generate $n_{maj} - n_{min}$ synthetic samples, where n_{maj} is the number of majority samples and n_{min} is the number of minority samples in the original data.

Obviously, our method DBO is robust to some outliers and almost generates all synthetic samples in the decision boundary that to form a hollow structure as seen in Fig. 7. On the one hand, this implies that DBO takes full use of information in the decision boundary. On the other hand, DBO well divides the decision boundary area into the boundary majority and minority areas.

B. Comparison on real-world benchmark datasets

In this section, we evaluate the performance of each method on real-world benchmark datasets from the UCI repository [42] and [43] as seen in Table S1. Before the experiment, all datasets are preprocessed by the standardized z-scores. We generate $n_{maj} - n_{min}$ synthetic samples for each over-sampling methods. And select NN (Neural Network), SVM (Support Vector Machine) and AdaBoostM1 (Method: AdaBoostM, NLearn: 10, Learners: decision tree) as the base classifiers. And we apply a twofold SKFCV (stratified k-fold cross validation, and setting k=2) for 30 times, that resulting in 60 runs on each dataset; then record the corresponding mean and standard deviation of 60 runs as the results.

To evaluate the classification performance, accuracy is currently used to evaluate the classification performance. But it does not apply to imbalanced data at all. The reason is that imbalanced classification cares more about the minority class. Thus, we select g-mean as the measurement to evaluate the classification performance. Besides, also select precision and recall for the comparison between different methods.

$$\begin{aligned} precision &= \frac{TP}{TP + FP} \\ recall &= \frac{TP}{TP + FN} \\ g - mean &= \sqrt{\frac{TP \times TN}{(TP + FN) \times (TN + FP)}} \end{aligned} \tag{25}$$

where TP, TN, FN and FP are respectively as the number of true positives, true negatives, false negatives and false positives.

As shown in Table S2, S3 and I, the recall, precision and g-mean of each method on the classifier NN are respectively displayed; besides, the recall, precision and g-mean on SVM

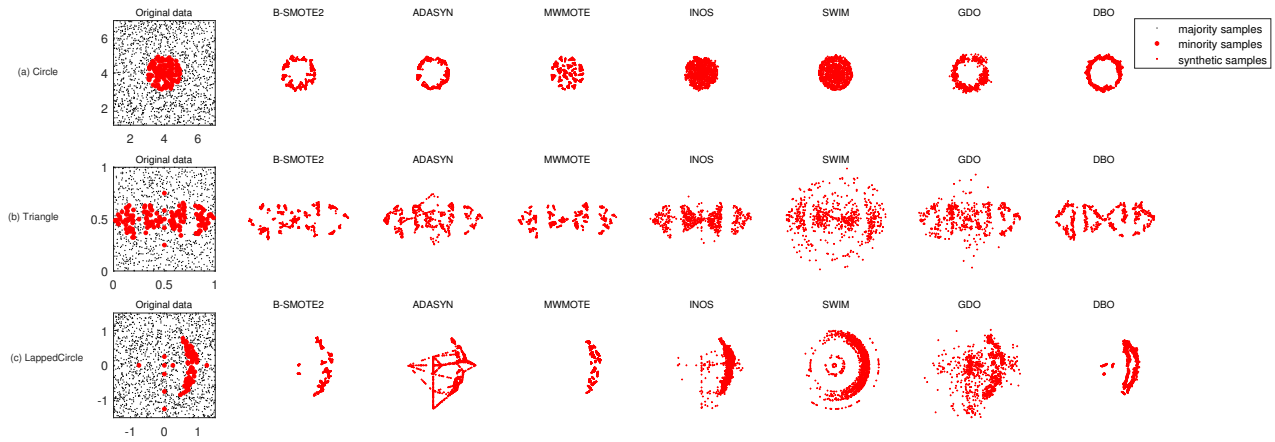

Fig. 7. Synthetic data in 2-D space with three emulational datasets including (a) Circle, (b) Triangle, (c) LappedCircle. For each row, the original data distribution is first plotted where red denotes the minority and black denotes the majority; then synthetic data by each over-sampling method are subsequently plotted.

are shown in Table S6, S7 and S8, and the recall, precision and g-mean on AdaBoostM1 are shown in Table S11, S12 and S13. As shown in Fig. S1, S2 and S3, their best-rank counts are respectively plotted. Where Ori means the method whose classifier is directly trained on imbalanced dataset. Different from the Ori method, other methods simultaneously send imbalanced dataset and synthetic samples to the classifier. And in each table cell, except the mean and standard deviation of 60 runs, the rank among all methods is recorded in a bracket (1 denotes the best rank).

As shown in Table S4, S9 and S14, mean ranks of precision, recall and g-mean on NN, SVM and AdaBoostM1 are computed for the further comparison. Besides, the Friedman test, as one of non-parametric statistical test, is applied to judge whether the significant difference exists among all methods. For example, as seen in Table S4, the actual value among all methods on g-mean is 92.18 that is larger than the table look-op value 14.07 (n=8-1, $\alpha = 0.05$); we reject the original hypothesis; thus there exists the significant difference among all methods on g-mean. Moreover, the Bonferroni-Dunn test, as one of post-hoc test, is applied to judge whether the significant difference exists between our method DBO and any one of other methods when DBO achieves the best mean rank. For example, as seen in Table S4, the gap of mean rank on g-mean between B-SMOTE2 and DBO is 2.24 (4.47-2.23) that is larger than the critical value 1.70; thus there exists the significant difference between B-SMOTE2 and DBO; in other words, DBO performs better than B-SMOTE2; and we denote a dagger symbol after the mean rank of B-SMOTE2 as 4.47†.

As shown in Table S5, S10 and S15, the Wilcoxon paired signed-rank test, as one of non-parametric statistical hypothesis test, is applied for pairwise comparisons between DBO and one of other methods that the significant difference does not exist after using the Bonferroni-Dunn test. In the Wilcoxon paired signed-rank test, the significant difference exists when the corresponding p-value is smaller than 0.05.

C. Performance analysis

In this subsection, we analyse the performance between different methods according to their mean ranks in Table S4 and S9. Roughly, DBO, SWIM, GDO, ADASYN and B-SMOTE2 obtain the good recall but worse precision; INOS, MWMOTE and Ori obtain the worse recall but better precision. It is straightforward that the former generates more synthetic samples in the decision boundary than the latter. In detail, DBO just generates synthetic samples in the decision boundary. SWIM generates synthetic samples along the borderline for boundary minority samples. GDO and ADASYN assign higher weights to boundary minority samples so generate more synthetic samples for them. B-SMOTE2 only generates synthetic samples for boundary minority samples. On the contrary, INOS generates synthetic samples in the integral minority area. And MWMOTE similarly assigns higher weights to boundary minority samples, but use clusters to generate synthetic samples that not tends to fall in the decision boundary.

From the perspective of g-mean, better g-mean means the good recognition rate of both minority and majority classes. As seen in Table S4, S9 and S14, DBO achieves the best mean ranks on g-mean. This implies DBO can well-meet the trade-off of classification between the minority and majority classes.

Specially for precision, the results of DBO of all three classifiers are not promising. The reason is that DBO just generates synthetic samples for the minority class in the decision boundary that make the classifier bias toward the minority class in turn. Thus, more boundary samples are identified as minority class, in other words, more majority samples are wrongly identified as minority class, that increasing the false positives (FP) so make the not promising average precision.

TABLE I
NN: AVERAGE G-MEAN

Dataset	*Ori*	*B-SMOTE2*	*ADASYN*	*MWMOTE*	*INOS*	*SWIM*	*GDO*	*DBO*
Cancer wpbc ret	0.4232±0.2073(8)	**0.6355±0.0622(1)**	0.6278±0.0922(3)	0.6308±0.0829(2)	0.6164±0.0701(4)	0.6028±0.0949(7)	0.6080±0.0804(6)	0.6094±0.1151(5)
Diabetes absent	0.6901±0.0517(8)	0.7337±0.0304(2)	0.7329±0.0296(3)	0.7309±0.0255(6)	0.7311±0.0249(5)	**0.7356±0.0270(1)**	0.7269±0.0362(7)	0.7328±0.0202(4)
Housing MEDV>35	0.6963±0.1937(8)	0.8357±0.0529(6)	0.8605±0.0510(5)	0.8342±0.0588(7)	0.8610±0.0508(4)	0.8701±0.0287(2)	0.8687±0.0431(3)	**0.8728±0.0346(1)**
Iris versicolor	0.9189±0.1256(8)	0.9290±0.1261(7)	0.9453±0.0243(3)	0.9432±0.0219(5)	0.9469±0.0271(2)	0.9315±0.0528(6)	0.9449±0.0241(4)	**0.9484±0.0268(1)**
Iris virginica	0.9381±0.0345(7)	0.9461±0.0393(5)	0.9483±0.0302(3)	0.9465±0.0339(4)	0.9493±0.0338(2)	0.9009±0.0495(8)	0.9439±0.0456(6)	**0.9574±0.0241(1)**
Spectf 0	0.6018±0.1992(8)	0.7612±0.0391(5)	0.7648±0.0406(3)	0.7666±0.0430(2)	0.7553±0.0462(6)	0.7361±0.0501(7)	0.7620±0.0462(4)	**0.7737±0.0359(1)**
Thyriod hyperfunction	0.0260±0.0484(8)	0.6916±0.0519(6)	0.7395±0.0585(3)	0.6932±0.0578(5)	**0.8094±0.0569(1)**	0.6010±0.1239(7)	0.7788±0.0680(2)	0.7375±0.0764 (4)
Vowel 4	0.6394±0.2515(8)	0.8744±0.0728(5)	0.9036±0.0545(3)	0.8724±0.1090(6)	0.8954±0.0673(4)	0.8594±0.0519(7)	0.9037±0.0505(2)	**0.9092±0.0406(1)**
Vowel 5	0.3358±0.3140(8)	0.8207±0.0660(7)	0.8539±0.0590(3)	0.8244±0.0768(6)	0.8444±0.0616(4)	0.8349±0.0693(5)	**0.8622±0.0540(1)**	0.8607±0.0602(2)
BreastTissue1	0.7033±0.3198(8)	0.8465±0.1253(7)	0.8777±0.0619(2)	0.8624±0.0650(5)	0.8775±0.0506(3)	**0.8896±0.0636(1)**	0.8552±0.1219(6)	0.8734±0.0565(4)
BreastTissue3	0.0695±0.1451(8)	0.5734±0.1617(6)	0.5963±0.1426(4)	0.6184±0.1036(3)	0.5854±0.1587(5)	0.5544±0.1972(7)	**0.6313±0.0995(1)**	0.6207±0.1226(2)
Ecoli2	0.8271±0.0471(8)	0.8756±0.0230(3)	0.8734±0.0255(4)	0.8688±0.0348(6)	0.8709±0.0329(5)	0.8679±0.0274(7)	0.8760±0.0269(2)	**0.8781±0.0181(1)**
Glass2	0.5227±0.1874(8)	0.6088±0.1053(5)	0.6232±0.0826(3)	0.6170±0.1350(4)	**0.6369±0.0869(1)**	0.5943±0.1067(7)	0.5980±0.1127(6)	0.6263±0.1265(2)
Glass3	0.0829±0.1716(8)	0.5219±0.1779(7)	0.6206±0.1715(5)	0.5815±0.1584(6)	0.6352±0.1202(4)	0.6381±0.1425(3)	0.6427±0.1393(2)	**0.6444±0.1334(1)**
ImageSegmentation7	0.9954±0.0022(7)	0.9961±0.0026(4)	0.9962±0.0025(3)	0.9956±0.0025(6)	**0.9972±0.0024(1)**	0.9957±0.0023(5)	0.9870±0.0112(8)	0.9965±0.0023(2)
LibrasMovement2	0.4636±0.3595(8)	0.8554±0.1184(7)	0.8809±0.0988(4)	0.8679±0.0918(6)	0.8797±0.0953(5)	0.8882±0.0575(2)	0.8830±0.0868(3)	**0.9029±0.0692(1)**
LibrasMovement14	0.7796±0.2289(8)	0.9025±0.0774(3)	0.9009±0.0673(4)	0.9002±0.0718(5)	0.8967±0.0709(6)	0.8946±0.0440(7)	0.9045±0.0672(2)	**0.9337±0.0564(1)**
Pageblocks2	0.7979±0.0743(8)	**0.9506±0.0082(1)**	0.9478±0.0091(2)	0.9395±0.0141 (4)	0.9380±0.0192(5)	0.8382±0.0701(7)	0.9336±0.0213(6)	0.9426±0.0222(3)
Pageblocks5	0.5227±0.1704(8)	0.8970±0.0195(5)	0.9040±0.0233(2)	0.8873±0.0298(6)	0.9029±0.0208(3)	0.6903±0.1284(7)	0.9017±0.0168(4)	**0.9126±0.0150(1)**
StatlogVehicleSilhouettes2	0.5758±0.1667(8)	0.7612±0.0472(4)	0.7671±0.0408(3)	**0.7693±0.0314(1)**	0.7602±0.0460(5)	0.7386±0.0458(7)	0.7561±0.0419(6)	0.7690±0.0437(2)
WallFollowingRobotNavigation4	0.8723±0.0840(8)	0.9379±0.0135(5)	0.9348±0.0151(6)	0.9335±0.0166(7)	0.9463±0.0115(3)	0.9426±0.0118(4)	0.9474±0.0205(2)	**0.9510±0.0089(1)**
Yeast6	0.6287±0.1934(8)	0.9416±0.0427(5)	0.9360±0.0374(6)	0.9282±0.0455(7)	0.9433±0.0296(4)	0.9464±0.0102(3)	**0.9618±0.0157(1)**	0.9599±0.0157(2)
DMEAntiVirus	0.9701±0.0267(6)	0.9817±0.0139(2)	0.9769±0.0179(3)	0.9722±0.0201(5)	0.9451±0.0312(8)	0.9597±0.0566(7)	**0.9819±0.0156(1)**	0.9736±0.0440(4)
GLRCWL1	0.6566±0.2443(7)	0.7751±0.1149(4)	0.7787±0.1145(3)	0.7585±0.0974(5)	0.7517±0.1054(6)	0.5933±0.1631(8)	0.7795±0.1413(2)	**0.7906±0.0956(1)**
GLRCNBI2	0.3328±0.2387(8)	0.5369±0.1920(5)	0.5389±0.1763(4)	0.5631±0.1583(2)	0.5465±0.1571(3)	0.5129±0.1996(7)	0.5313±0.1625(6)	**0.5915±0.1526(1)**
ParkinsonsDC	0.6581±0.1571(8)	0.7542±0.1021(6)	0.7692±0.0291(2)	0.7672±0.0294(3)	**0.7726±0.0256(1)**	0.7111±0.0449(7)	0.7644±0.0286(4)	0.7579±0.0284(5)
Colon 1	0.6313±0.2037(8)	0.6941±0.1831(5)	0.6914±0.2262(6)	0.7454±0.1522(2)	**0.7470±0.1460(1)**	0.6769±0.1883(7)	0.7254±0.1810(4)	0.7379±0.1556(3)
Leukemia 1	0.9106±0.1125(5)	0.9326±0.0802(2)	**0.9428±0.0543(1)**	0.9282±0.1389 (3)	0.7769±0.0914(7)	0.7292±0.2157(8)	0.9220±0.1519(4)	0.8685±0.1082(6)
DrivFace1	0.8417±0.1070(8)	0.9120±0.0627(3)	0.8977±0.0779(6)	0.9026±0.0711(4)	0.8978±0.0772(5)	0.9341±0.0502(2)	0.8838±0.0733(7)	**0.9368±0.0508(1)**
ARBT8	0.0000±0.0000(7.5)	**0.5299±0.1947(1)**	0.3442±0.2024(4)	0.0000±0.0000(7.5)	0.4982±0.1126(2)	0.2879±0.2789(5)	0.0809±0.1592(6)	0.4976±0.1761(3)

A stratified k-fold cross validation (k=2 in experience) is used for 30 times that 60 (2×30) runs are conducted. Thus for each table cell, the mean and standard deviation of corresponding performance on 60 runs are first recorded, then it's rank among all methods is followed in one bracket. The best rank for each row is highlighted as bold.

V. CHARACTERISTICS OF DBO

A. Ablation study between linear interpolation and surrounding area

To explain why considering the surrounding area is important, we conduct the ablation study between linear interpolation and surrounding area. Where 'Linear interpolation' means the interpolation between the target boundary minority and it's k nearest majority; and 'surrounding area' means the interpolation in the surrounding area (or the local boundary minority area) of the target boundary minority. As seen in Fig. S4, 'surrounding area' obtains the good performance on most datasets. Thus, considering the surrounding area is important.

B. Robust to outliers

As seen in Fig. S5, three scaling size of graphs for each outlier are plotted in a col. First for the bottom and middle one, the local decision boundary area is covered by the local boundary majority area so lead to the empty local boundary minority area. Then, for the top one, the local boundary majority area covers a part of the local decision boundary area; especially in this scene, the remained minority area distributes in the non-majority existed region (a certain region that no majority existed).

In general, DBO is robust to some outliers and only generates synthetic samples in very near regions or temporary non-majority existed regions for other outliers.

C. Parameter setting

DBO involves one parameter as k. But to consider the influence of lengths (α) in two ellipsoid structures, we first change Eq. 12 to $\alpha_2 = r_1 * \alpha_1$, and Eq. 16 to $\alpha_5 = \alpha_4 + r_2 * |\alpha_3 - \alpha_4|$; then test two parameters r_1 and r_2.

(1) k: it means k nearest majority samples of the target boundary minority sample. As seen in Fig. S6, the larger value of k means larger sizes of both the local decision boundary area and local boundary minority area. As seen in Fig. S7, no optimal k for all datasets; thus to maintain the local property for DBO, we set $k = 7$.

(2) r_1: larger r_1 means larger length of α_2 that make the larger size of the local decision boundary area. As seen in Fig. S8, for some datasets, the performance of g-mean decreases as r_1 increases; because the local decision boundary area tends to cover much more majority area that resulting in noises or overlapping. But for some other datasets, the performance of g-mean increases as r_1 increases. Maybe the majority area covered by the local decision boundary area is not out of the local boundary majority area, and the local decision boundary area covers more boundary minority area. In experience, we set $r_1 = 2$.

(3) r_2: larger r_2 means larger length of α_5 that make the larger size of the local boundary majority area. As seen in Fig. S9, for many datasets, the performance of g-mean decreases when $r_2 > 0.5$, because the local boundary majority area tends to cover the overall local decision boundary area that resulting in no synthetic sample; in experience, we set $r_2 = 0.5$.

D. time-consuming

As shown in Table S16, the time-consuming of different methods is displayed. In the experiment, we run the code of eigen decomposition on NVIDIA GeForce GTX 1050Ti and remained code on Intel Core i9 CPU. Obviously, DBO costs many seconds on high-dimensional; because, their computation of eigen decomposition is time-consuming. Besides, for some datasets with middle dimension and large number of minority samples, DBO also costs several seconds because of their large number of local decision boundary area to be computed.

VI. CONCLUSION

In this paper, a novel Decision Boundary Computation-based Oversampling (DBO) method is proposed to address

the imbalanced problem by taking full use of information in decision boundary. Firsts, DBO computes the decision boundary area and the boundary majority area; then obtains the corresponding boundary minority area by subtracting the boundary majority area from the decision boundary area. Finally, DBO generates new synthetic samples in the boundary minority area. Thus, DBO not only takes individual samples, but also their surrounding areas into consideration. Moreover, DBO innovatively divides the decision boundary area into the boundary majority and minority areas. And experimental results on real-world datasets show the good performance on recall and g-mean. Especially on recall, DBO can greatly enhance the recognition rate of minority class.

In the future, some works will be attached to improve the robustness towards outliers and good structure representation of the boundary majority area.

REFERENCES

[1] S. Xia, Y. Zheng, G. Wang, P. He, H. Li, and Z. Chen, "Random space division sampling for label-noisy classification or imbalanced classification," *IEEE Trans. Cybern.*, early access, Apr. 28, 2021, doi: 10.1109/TCYB.2021.3070005.

[2] Z. Zhu, Z. Wang, D. Li, Y. Zhu, and W. Du, "Geometric structural ensemble learning for imbalanced problems," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1617–1629, Apr. 2020.

[3] Z. Ji, X. Yu, Y. Yu, Y. Pang, and Z. Zhang, "Semantic-guided class-imbalance learning model for zero-shot image classification," *IEEE Trans. Cybern.*, early access, May. 26, 2021, doi: 10.1109/TCYB.2020.3004641.

[4] Q. Zhu, W. Deng, Z. Zheng, Y. Zhong, Q. Guan, W. Lin, L. Zhang, and D. Li, "A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification," *IEEE Trans. Cybern.*, early access, May. 25, 2021, doi: 10.1109/TCYB.2021.3070577.

[5] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.

[6] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1532–1545, Jun. 2016.

[7] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Deep imbalanced learning for face recognition and attribute prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2781–2794, Nov. 2020.

[8] A. Sen, M. Islam, K. Murase, and X. Yao, "Binarization with boosting and oversampling for multiclass classification," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1078–1091, May 2016.

[9] M. L. Zhang, Y. K. Li, H. Yang, and X. Y. Liu, "Towards class-imbalance aware multi-label learning," *IEEE Trans. Cybern.*, early access, Nov. 18, 2020, doi: 10.1109/TCYB.2020.3027509.

[10] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, May 2019.

[11] J. Hu, H. Yang, M. R. Lyu, I. King, and A. M. So, "Online nonlinear auc maximization for imbalanced data sets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 882–895, Apr. 2018.

[12] C. Huang, C. C. Loy, and X. Tang, "Discriminative sparse neighbor approximation for imbalanced learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1503–1513, May 2018.

[13] Y. Xu, "Maximum margin of twin spheres support vector machine for imbalanced data classification," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1540–1550, 2017.

[14] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, "Classification of imbalanced data by oversampling in kernel space of support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, Sep. 2018.

[15] C. T. Li, T. Y. Liu, Y. Y. Lin, C. N. Fang, Y. K. Wang, G. Wang, N. R. Pal, and C. H. Chuang, "Minority oversampling in kernel adaptive subspaces for class imbalanced datasets," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 950–962, May 2018.

[16] X. Zhang, D. Ma, L. Gan, S.Jiang, and G. Agam, "Cgmos: Certainty guided minority oversampling," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Indianapolis, IN, USA, 2016.

[17] A. Tayal, T. F. Coleman, and Y. Li, "Rankrc: Large-scale nonlinear rare class ranking," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3347–3359, Dec. 2015.

[18] C. L. Castro and A. P. Braga, "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 888–899, Jun. 2013.

[19] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, May 2015.

[20] S. Ren, W. Zhu, B. Liao, Z. Li, P. Wang, K. Li, M. Chen, and Z. Li, "Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning," *Knowledge-Based Systems.*, vol. 163, pp. 705–722, Jan. 2019.

[21] S. Datta and S. Das, "Near-bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs," *Neural Netw.*, vol. 70, pp. 39–52, Oct. 2015.

[22] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted f-measure and kernel scaling for imbalanced data learning," *Inf. Sci.*, vol. 257, p. 331–341, Feb. 2014.

[23] S. Datta and S. Das, "Multiobjective support vector machines: handling class imbalance with pareto optimality," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1602–1608, May 2019.

[24] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[25] M. Pérez-Ortiz, P. A. Gutiérrez, P. Tino, and C. Hervás-Martínez, "Oversampling the minority class in the feature space," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1947–1961, Sep. 2016.

[26] A. Manukyan and E. Ceyhan, "Classification of imbalanced data with a geometric digraph family," *J. Mach. Learn. Res.*, vol. 17, pp. 1–40, Jan. 2016.

[27] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017.

[28] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 805–808.

[29] S. Liu, J. Zhang, Y. Xiang, and W. Zhou, "Fuzzy-based information decomposition for incomplete and imbalanced data learning," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1476–1490, Dec. 2017.

[30] W. W. Y. Ng, J. Hu, D. Yeung, S. Yin, and F. Roli, "Diversified sensitivity-based undersampling for imbalance classification problems," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2402–2412, Nov. 2015.

[31] W. W. Y. Ng, S. Xu, J. Zhang, X. Tian, T. Rong, and S. Kwong, "Hashing-based undersampling ensemble for imbalanced pattern classification problems," *IEEE Trans. Cybern.*, early access, Jun. 29, 2020, doi: 10.1109/TCYB.2020.3000754.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[33] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[34] P. Lim, C. K. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2850–2861, Sep. 2017.

[35] H. Cao, X. L. Li, D. K. Woon, and S. K. Ng, "Integrated oversampling for imbalanced time series classification," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2809–2822, Dec. 2013.

[36] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[37] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, "Amdo: An over-sampling technique for multi-class imbalanced problems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018.

[38] H. Han, W. Wang, and B. Mao, "Borderline-smote: A new oversampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intelligent Computing.*, 2005, pp. 878–887.

[39] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote—majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[40] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, "Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance," in *Proc. IEEE Int. Conf. on Data Mining.*, 2018, pp. 447–456.
[41] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, "Gaussian distribution based oversampling for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 10, 2020, doi: 10.1109/TKDE.2020.2985965.
[42] D. Dua and K. T. Efi, "Uci machine learning repository," University of California, Irvine, School of Information and Computer Sciences, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.
[43] "One-class classifier results." [Online]. Available: http://homepage.tudelft.nl/n9d04/occ/index.html

PLACE PHOTO HERE

Yi Sun is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China.

His research interests include data mining and machine learning.

PLACE PHOTO HERE

Lijun Cai received the Ph.D. degree in computer application technology from Hunan University, Changsha, China.

He is currently a Full Professor of computer science and technology with Hunan University. His research interests include machine learning and image processing.

PLACE PHOTO HERE

JunLin Xu is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China.

His research interests include machine learning and biochemical research method.

PLACE PHOTO HERE

Bo Liao received the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China, in 2004.

From 2004 to 2006, he was a Post-Doctoral Fellow with the University of Chinese Academy of Sciences, Beijing, China. He is currently a Full Professor with Hainan Normal University. He has authored over 100 papers in international conferences and journals. His research interests include image processing, bioinformatics, and big data processing.

PLACE PHOTO HERE

Wen Zhu received the M.S. degree in computer science and technology from Hunan University, Changsha, China, in 2010.

She is currently a Lecturer with Hainan Normal University. Her research interests include image processing and analysis and bioinformatics.