Decision Boundary Computation-based Over-sampling for Imbalance Learning

Yi Sun [1], Lijun Cai [2], and JunLin Xu [2]

[1]Hunan University
[2]Affiliation not available

October 30, 2023

Decision Boundary Computation-based Over-sampling for Imbalance Learning

Yi Sun, Lijun Cai, and JunLin Xu

***Abstract*—Imbalanced problem, one significant challenge in data mining, occurs when the number of samples in one class (minority) is obviously smaller than the other one (majority). Over-sampling methods that generating new synthetic samples for the minority class have been proven to be effective. But rare over-sampling methods focus on the decision boundary between classes and none of them are proposed to directly compute the certain area of decision boundary for imbalanced problem. Thus, one novel method named Decision Boundary Computation-based Oversampling is proposed to fill this gap. The novel method employs the intuitive observation, that both boundary samples and their surrounding areas corporately constitute the decision boundary, to compute the partition belonging to the minority class by subtracting the partition of majority class from their corporate one. Which greatly enhancing the full use of boundary information brought by both boundary individuals and their near areas, and implicitly complement the nature information insufficiency 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 imbalance, served as one of the most challenging problem in data mining and machine learning, appears in many real-world applications like credit fraud detection [1], stream data mining[2], face recognition [3] and so on. In one binary classification, one class of the smaller number of samples is called as the minority class and samples of this class are called as minority samples, and another one as the majority class and majority samples. 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, especial for some secure domains. So learning from imbalanced data is a long-standing and significant challenge for machine learning [4].

To deal with the imbalanced problem, several techniques have been reported and proven to be efficient that mainly involving the algorithm-level strategy [5], [6] and the data-level strategy [7], [8], [9]. First for the algorithm-level strategy, the cost-sensitive learning [10], [11] and the ensemble learning [12], [13] are two extensively used techniques to cope with the imbalanced problem. Besides, the algorithm-level strategy also includes some other techniques like hyperplane shift [14], kernel perturbation [15] and multiobjective optimization[16]. Then, the data-level strategy mainly include the minority over-sampling [17], [18] and majority under-sampling [21], [22] techniques. The minority over-sampling technique balances the ratio between classes by generating new synthetic samples for the minority class [19], [20]. Inversely, the under-sampling technique decreases the number of majority samples for the balanced ratio. In this paper, we focus on the over-sampling technique for its characteristic that not missing any original information. For example, the under-sampling technique may lose some important information on original data after decreasing the number of majority samples.

From the perspective of interpolation of synthetic samples, the over-sampling technique mainly includes the linear-interpolation [23], [24], [25] and non-linear or structure-preserving interpolation methods [26], [27], [28]. For example, synthetic minority over-sampling technique(SMOTE) [23] generates one new synthetic sample by the linear interpolation between the target minority sample and its randon one of k-nearest neighbours of minority. On the basis of SMOTE, the linear-interpolation method also involves into the borderline minority over-sampling [29], hard-to-learn minority over-sampling [24], [30] and kernel over-sampling [7], [18] techniques. Contrary to those linear-interpolation methods, structure-preserving interpolation method first estimates the corresponding structure of minority class and then generates new samples to maintain or preserve this estimated structure. For example in [26] and [27], they use the covariance of minority class to generate new synthetic samples.

However, only techniques like B-SMOTE1 and B-SMOTE2 in [29], ADASYN in [24] and MWMOTE in[30] involve the decision boundary between classes. In detail, B-SMOTE1 and B-SMOTE2 [29] only generate synthetic samples for minority samples that near to the borderline (called them borderline minority samples for convenience); ADASYN [24] and MWMOTE [30] only generate synthetic samples for hard-to-learn minority samples that with different weights for selection, which making more subtler and fine distinctions between borderline minority samples. Although these techniques select borderline minority samples for generating synthetic samples, only rough information in decision boundary are used. In other words, they only use the linear-interpolation between selected samples to generate synthetic samples and not consider their surrounding areas in decision boundary at

All authors 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).

Corresponding authors:Lijun Cai.

all. Besides, some of other techniques do not care about the decision boundary but potentially or slightly generated several synthetic samples in the decision boundary. For example, like INOS in [26], it preferentially generates synthetic samples in the whole data space with corresponding covariance structure and subsequently cleans synthetic data that nearer to majority samples. Thus, it finally obtains several synthetic samples in the decision boundary but earns much from the date cleaning technique. Simultaneously, INOS borrows a small percentage of synthetic samples from ADASYN to protect the key original minority samples. Moreover, like SWIM in [31], it generates synthetic samples for each minority sample with the same Mahalanobis distance from the majority class mean. Thus, it generates several synthetic samples in the decision boundary for borderline minority samples but may include several overlapped synthetic samples at the same time.

In this sense, we, therefore, propose one Decision Boundary Computation-based Over-sampling (DBO) method to fill this gap. From the intuitive observation, the area in design boundary not only includes individual samples but also their surrounding areas. To take full use of boundary information, we first compute the area of decision boundary on the basis of boundary majority and minority samples and their surrounding areas; and compute the partition belonging to the majority class on the basis of boundary majority samples and their surrounding areas. Then, we obtain the the partition belonging to the minority class by subtracting the partition of majority class from the area of decision boundary. For convenience, we call the area of 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 which may give a theoretical reference for many classification tasks.
2) We propose one novel minority over-sampling method that generating synthetic samples in the boundary minority area for class imbalance problem.
3) We take full use of information in 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 being robust to several outliers at the same time.

The rest of paper is organized as follows. Sections II reviews several 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

Han et al. [29] proposes 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 its majority nearest neighbours is larger than the number of its minority ones. For example, denoting the number of majority samples among 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 minority samples for linear-interpolation to generate new synthetic samples; specially, B-SMOTE2 searches k nearest neighbours from both majority and minority samples for linear-interpolation. To make more subtler and fine distinctions between different borderline minority samples, He et al.[24] proposes ADASYN to assign minority samples with different weights by the ratio of the number of majority samples in m nearest neighbours. Where the higher weight means the higher level of difficulty in learning and more synthetic samples are generated. Slight difference from ADASYN, MWMOTE first identifies hard-to-learn informative minority samples, and then assigns them different weights according to their Euclidean distance from nearest majority samples [30]. Specially, MWMOTE does not use k nearest neighbours for linear-interpolation but use clusters. For example, for one hard-to-learn minority sample, MWMOTE searches one random minority samples that with the same cluster as the hard-to-learn one for linear-interpolation. Thus, all synthetic samples by MWMOTE lie in clusters of minority class.

Among those methods, for the target borderline minority one, B-SMOTE1, ADASYN and MWMOTE only search one from minority samples for linear-interpolation; and B-SMOTE2 searches one from both majority and and minority samples for linear-interpolation. For the first type of methods, B-SMOTE1 and ADASYN search a rand one from k nearest minority neighbours for linear-interpolation; and MWMOTE search a rand one from the same cluster for linear-interpolation. Thus, B-SMOTE2 generates more synthetic samples in the design boundary but may generate some overlapped synthetic samples at the same time. ADASYN generates more synthetic samples for those minority samples with more majority samples surrounded, and MWMOTE generates more synthetic samples for those with nearer distance to majority samples and make those synthetic samples never erroneously falling into the majority class region.

Obviously, above methods only use individual minority and majority samples for linear-interpolation and not consider their surrounding areas at all.

B. Motivation

From the intuitive observation, boundary individual samples and their surrounding areas together constitute the decision boundary area as seen in Fig. 1. (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 existed make it difficult to directly compute the boundary minority area. In the one hand, rare number of boundary minority samples leads to the missing information on minority class in the boundary minority area. In 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 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 (for all decision boundary, boundary majority and minority area); we first compute a serious of local boundary areas; we then integrate those local boundary areas together to approximately represent the integral boundary area. Details of proposed method are described below.

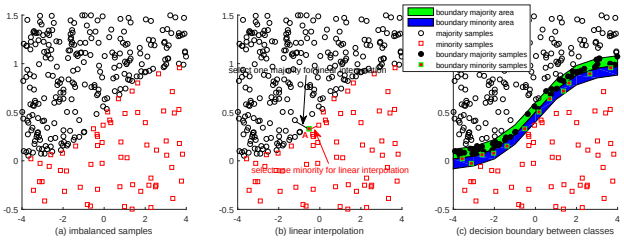

Fig. 1. Motivation of proposed method. (a) original imbalanced samples; black circle means majority samples, red square means minority samples; obviously, the number of majority samples is larger than the number of minority one, and the minority class misses many information in the data space especial for the decision boundary; inversely, the majority class owns more information in the decision boundary. (b) linear interpolation; the solid red square denotes one point A that selected for display of linear interpolation, the red line segment means the linear interpolation between A and one of its neighbouring minority sample and the black line segment means the linear interpolation between A and one of its neighbouring majority sample; for example, like B-SMOTE1, ADASYN and MWMOTE, they would like to generate one synthetic sample in the line segment between A and one other minority sample (denoted as the red line segment); like B-SMOTE2, it would like to generate one synthetic sample in the line segment between A and one other majority or minority sample (denoted as the black or red line segment), but may generate one overlapped synthetic sample in the black line segment. (c) decision boundary between classes; solid black circle means boundary majority samples, solid red square means boundary minority samples, the green area means the boundary majority area and the blue area means the boundary minority area; obviously, boundary majority samples own more information in the decision boundary than the boundary minority one; if we can compute the green and blue areas, it can be beneficial both for the oversampling and classification of imbalanced data.

C. Preliminary knowledge of area computation

In this section, we introduce how to compute 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 its 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.2. (a) the black circle denotes one group of samples. Fig.2. (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 coordinates vector for point A. And Fig.2. (c) plots the area of this group when assigning $\alpha = 1.5 \times \alpha_A$. Obviously, A is one point in 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 its corresponding covariance matrix with one length α. In other words, the certain area depends on the selection of corresponding group of samples and the assignment of length α.

III. DECISION BOUNDARY COMPUTATION-BASED OVER-SAMPLING

The proposed method mainly involves into three steps respectively as the computation of decision boundary area, the computation of boundary majority area and the generation of synthetic samples. For both first two steps, we preferentially select a group of samples and posteriorly assign one length to corresponding covariance matrix. As seen in Fig.3, the general procedure chart of proposed method is plotted for the computation of local boundary minority area. After subtracting the boundary majority area from the decision boundary area, the remained area are obtained as the boundary minority area in which synthetic samples are generated.

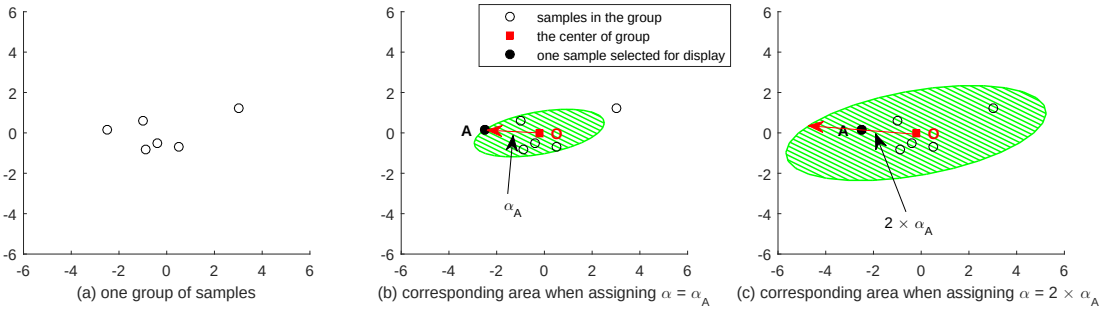

Fig. 2. area computation. (a) a group of samples. (b) corresponding area when assigning $\alpha = \alpha_A$; the solid black circle denotes one samples selected for display, the solid red square denotes the center of this group and the red arrow denotes the predefined length α for the covariance matrix of group; obviously, this area is one ellipse in $\mathbb{R}^2$; there, $\alpha = \alpha_A$ that computed in Eq. 5, thus point A is on the surface of ellipse. (c) corresponding area when assigning $\alpha = 2 \times \alpha_A$; the red arrow denotes the predefined length α for the covariance matrix of group; there, $\alpha = 2 \times \alpha_A > \alpha_A$, thus point A and its surrounding area are simultaneously covered. Totally, for the computation of one local area in our method, only a group of samples and one suitable length α are needed.

A. decision boundary area

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

$$BMIN = \{bmin_1, bmin_2, ..., bmin_i, ..., bmin_n\} \quad (7)$$

where n is the number of boundary minority samples and $bmin_i$ is the i-th boundary minority sample in $BMIN$. Then, for each boundary minority sample in $BMIN$, k nearest majority samples are computed:

$$BMAJ = \{\{Bmaj_1\}, \{Bmaj_2\}, .., \{Bmaj_n\}\} \quad (8)$$

$$Bmaj_i = \{bmaj_1^i, bmaj_2^i, .., bmaj_k^i\} \quad (9)$$

where $Bmaj_i$ includes k nearest majority samples for $bmin_i$.

Finally, for each boundary minority sample, we select the group of samples.

$$groupDecB = \{\{gdb_1\}, \{gdb_2\}, ..., \{gdb_n\}\} \quad (10)$$

$$gdb_i = \{bmin_i, mean_1^i, mean_2^i, ..., mean_k^i\} \quad (11)$$

$$mean_j^i = \frac{bmaj_j^i + bmin_i}{2} \quad (12)$$

where gdb_i includes the target boundary minority sample and mean points between it and its k nearest majority samples.

As seen in Fig.4, the reason why we choose mean points instead of k nearest majority samples is plotted and discussed. Obviously as seen in Fig.4. (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 corresponding covariance matrix of preferentially selected group. To cover both the target boundary minority sample and its surrounding area with enough range size, we double up this area with the length of target boundary minority sample:

$$\alpha_i^{DB} = 2 * \alpha_i^{bmin} \quad (13)$$

$$\alpha_i^{bmin} = (bmin_i - \bar{x}_i^{DB})^T (U_i^{DB})^{-1} (bmin_i - \bar{x}_i^{DB}) \quad (14)$$

where α_i^{DB} is the length to assign to corresponding covariance matrix, $bmin_i$ denotes the i-th minority sample in BMIN; $\bar{x}_i^{DB}$ and U_i^{DB} are respectively as the center and covariance matrix of group ($gdb_i = \{bmin_i, mean_1^i, mean_2^i, ..., mean_k^i\}$); α_i^{bmin} is the length of target boundary minority sample $bmin_i$. Since the group of samples and corresponding length are selected and assigned, the local decision boundary area is computed as:

$$S_i^{DB} = \{x |\ (x - \bar{x}_i^{DB})^T (U_i^{DB})^{-1} (x - \bar{x}_i^{DB}) \leq \alpha_i^{DB}\} \quad (15)$$

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

As seen in Fig.3. (b) and (d), the local decision boundary area does not deeply root into the majority area and is of enough size at the same time. In detail, for the first goal, as seen in Fig.4. (b), we zoom out the local decision boundary area by setting $ratio = 0.5$. Obviously in Fig.4. (a), smaller $ratio$ makes the local decision boundary area smaller like $ratio = 0.25$ that not enough area is covered. As seen in Fig.4. (c) or (d), larger $ratio$ makes the local decision boundary area larger like $ratio = 0.75$ or 1 that over-much area are covered. For the second goal, as seen in Fig.3. (b) and Fig.4. (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 its near local minority area. In general, we first double down this area and 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 and then assign one length to corresponding covariance matrix. Different from the local decision boundary area, we directly select all k nearest majority samples in Eq. 9 into the group (as $Bmaj_i = \{bmaj_1^i, bmaj_2^i, .., bmaj_k^i\}$). To cover both local boundary majority samples and their surrounding areas and not involve into the local boundary minority area at the same time, we refer to lengths of target boundary minority sample and its nearest majority one and assign the nearly mean length of those referred two to corresponding covariance matrix:

$$\alpha_i^{MAJ} = \alpha_i^{nnmaj} + 0.5 \times |\alpha_i^{min} - \alpha_i^{nnmaj}| \quad (16)$$

$$\alpha_i^{nnmaj} = (bmaj_{nn} - \bar{x}_i^{MAJ})^T (S_i^{MAJ})^{-1} (bmaj_{nn} - \bar{x}_i^{MAJ}) \quad (17)$$

$$\alpha_i^{min} = (bmin_i - \bar{x}_i^{MAJ})^T (S_i^{MAJ})^{-1} (bmin_i - \bar{x}_i^{MAJ}) \quad (18)$$

where $bmaj_{nn}$ is the nearest one in k nearest majority samples for the boundary minority sample $bmin_i$, α_i^{nnmaj} is the length of $bmaj_{nn}$; $\bar{x}_i^{MAJ}$ and S_i^{MAJ} are respectively as the center and covariance matrix of group ($Bmaj_i = \{bmaj_1^i, bmaj_2^i, .., bmaj_k^i\}$). Besides, the use of absolute value in Eq. 16 is prepared for outliers of minority class that will be seen in Section V-A.

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

$$S_i^{MAJ} = \{x |\ (x - \bar{x}_i^{MAJ})^T S_i^{-1} (x - \bar{x}_i^{MAJ}) \leq \alpha_i^{MAJ}\} \quad (19)$$

As seen in Fig.3. (c), we refer to lengths of B and C (the nearest majority to B) and assign nearly mean length of those two to corresponding α. Obviously, the local boundary majority area tends to cover much of decision boundary area when assigning $\alpha = \alpha_B$; similarly, it tends to cover little of the decision boundary area when assigning $\alpha = \alpha_C$. To make the trade-off between those two scenes, we assign $\alpha = \alpha_C + 0.5 \times |\alpha_B - \alpha_C|$ owing to that almost the half of local decision boundary area belongs to the minority class and another half belongs to the majority class.

C. Generation of synthetic samples

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

$$S_i^{MIN} = S_i^{DB} - S_i^{DB} \cap S_i^{Maj} \quad (20)$$

As seen in Fig.3. (d), we subtract the intersecting area from the local decision boundary area and the remained area denotes the local boundary minority area. To this end, we respectively integrate those local boundary areas together to approximately estimate corresponding integral boundary areas.

$$S_{DBA} = \bigcup_{i=1}^{n} S_i^{DB} \quad (21)$$

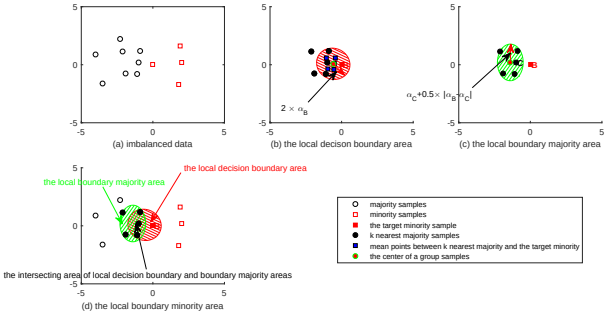

Fig. 3. The computation of the local boundary minority area. (a) imbalanced data; the red square denotes minority samples and the black circle denotes majority samples. (b) the local decision boundary area; the solid red square denotes the target boundary minority sample B for the display of computation of local decision boundary area, the solid black circle denote k nearest majority samples for B (there k=5, and calling them as k nearest boundary majority samples), the solid blue square denotes mean points between B and its k nearest boundary majority samples, the solid red circle denotes the center of those mean points and B, the red arrow denotes the predefined length α for the covariance matrix of group and the red slash area denotes the local decision boundary area; first for the selection of a group of samples, those k mean points and one point B are selected; then for the selection of suitable length α, we set $\alpha = 2 \times \alpha_B$; thus, the local decision boundary area covers the minority samples B and its its surrounding area, and part of boundary majority samples and their surrounding areas. (c) the local boundary majority area; the solid black circle denotes k nearest boundary majority samples for B, the solid red circle denotes the center of those k nearest boundary majority samples, the red arrow denotes the predefined length α for the covariance matrix of group and the green slash area denotes the local boundary majority area; first for the selection of a group of samples, those k nearest boundary majority samples are selected; then for the selection of suitable length α, we set $\alpha = \alpha_C + 0.5 \times |\alpha_B - \alpha_C|$ (notice: α_B in (c) is not equal to α_B in (b), because of different group of samples selected so leading to different covariance matrices or ellipses in $\mathbb{R}^2$); thus, the local boundary majority area covers part of boundary majority samples and their surrounding areas; sometimes, all boundary majority samples are covered, sometimes not, because we care more about whether the majority area near to the target boundary minority sample B is covered; of course, a larger α may help to cover all boundary majority samples, but also tend to cover much local boundary area that belonging to the minority class. (d) the local boundary minority area; since the local decision boundary area and the local boundary majority area are computed, we obtain the local minority area by subtracting the local boundary majority area; obviously, the subtracting part is the intersecting area of the local decision boundary area and the local boundary majority area, and the remained area in local decision boundary area is obtained as the local boundary minority area.

$$S_{BMajA} = \bigcup_{i=1}^{n} S_i^{Maj} \quad (22)$$

$$S_{BMINA} = \bigcup_{i=1}^{n} S_i^{MIN} \quad (23)$$

where n denotes the number of boundary minority samples in $BMIN$ in Eq. 7; S_{DBA} is the integral decision boundary area, S_{BMajA} is the integral boundary majority area and S_{BMINA} is the integral boundary minority area.

Then for imbalanced problem, we generate new synthetic samples in the boundary minority area in Eq. 23. Of course, direct generation in the boundary minority area is impossible. As seen in Eq. 20, we only own the information of local decision boundary area and boundary majority area. Thus, we first generate the synthetic sample in one randomly selected local decision boundary area and then judge whether it falls in the boundary majority area; only condition-satisfied one is

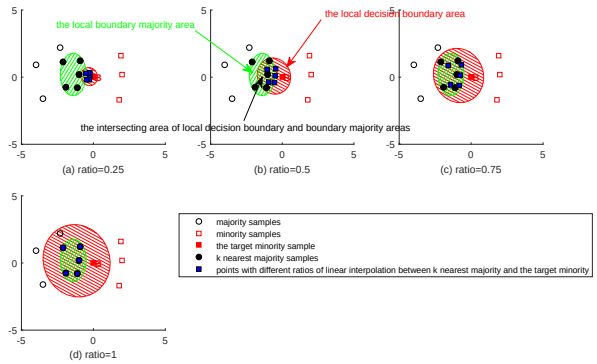

Fig. 4. Different selections of the group of samples for the local decision boundary area; we only carry the liner selection between one target boundary minority sample and its k nearest boundary majority samples (one liner selected sample: $(1-ratio)\times bmin + ratio \times bmaj_j$ where $bmin$ and $bmaj_j$ are respectively coordinate vectors of target boundary minority sample and the j-th one in k nearest boundary majority samples); and use different ratios to denote the location of liner selected samples; for example, liner selected samples are mean points between the target boundary minority sample and its k nearest boundary majority samples when setting $ratio = 0.5$; the red slash area denotes the decision boundary area and the green slash area denotes the boundary majority area for all(a)-(d). (a) $ratio = 0.25$, (b) $ratio = 0.5$, (c) $ratio = 0.75$, (d) $ratio = 1$. Obviously, as the $ratio$ increases, other liner selected points are nearer to corresponding k nearest boundary majority samples (when $ratio = 1$, those liner selected points are just k nearest boundary majority samples); so the local decision boundary area covers more areas as the $ratio$ increases; moreover, the local decision boundary area would cover the whole local boundary majority area with a larger $ratio$ (like $ratio = 0.75$ or 1); of course, smaller $ratio$ (like $ratio = 0.25$) will make the local decision boundary area be of smaller size. In general, for the computation of local decision boundary area, we first zoom out this area (which includes both samples and their surrounding areas) by setting $ratio = 0.5$ first and then zoom in this area by setting $\alpha = 2 \times \alpha_B$. Two zooming operations make the local decision boundary area be of enough size and not root into the majority area too much at the same time, especially avoid the scene that the local decision boundary area covers the whole local boundary majority area.

recorded as the new synthetic sample for the minority class.

In detail, for the generation of one synthetic sample in the local decision boundary area, we first use the rand Gaussian distribution ($G(\mu = 1, \sigma = 1)$) generator to generate one rand value and then obtain the rand length len (in [0,1]) as:

$$len = (1 - |G(\mu, \sigma)|) * \alpha_i^{DB} \tag{24}$$

where $G(\mu, \sigma)$ is one rand value by the Gaussian generator. Next, one normalized direction $direc$ is randomly generated satisfying

$$||direc|| = 1 \tag{25}$$

The new synthetic sample is computed as:

$$New_{temp} = V_i^{DB}(E_i^{DB})^{1/2}(len * direc) + \bar{x}_i^{DB} \tag{26}$$

where V_i^{DB} and (E_i^{DB}) are components in the eigen decomposition of covariance matrix U_i^{DB} in Eq.4. In the next step, we judge whether New_{temp} falling in S_i^{Maj} by Eq.19. If not falling in, record New_{temp} as the new synthetic sample for the minority class; if falling in, re-generate one new sample New_{temp} again. To consider the possibility that the local boundary majority area covers the whole local decision boundary area, we restrict the time of re-generation as 100 in experience. The algorithm of DBO is seen in Algorithm 1.

Algorithm 1 DBO

Input: Training set: $T = \{T_{maj}, T_{min}\}$; number of samples in majority and minority class: n_{maj} and n_{min}.
Output: Synthetic samples S_{new}
Construct the boundary minority set $BMIN$;
Construct the boundary majority set $BMAJ$;
for i=1 to n_{min} **do**
  Pick up the i-th boundary minority sample $bmin_i$ from $BMIN$;
  Pick up corresponding boundary majority samples $Bmaj_i$ from $BMAJ$;
  Estimate the local boundary area S_i^{DB};
  Estimate the local boundary majority area S_i^{Maj};
  Obtain the local boundary minority area $S_i^{MIN} = S_i^{DB} - S_i^{DB} \cap S_i^{Maj}$;
end for
Compute the number of new synthetic samples: $N = n_{maj} - n_{min}$
n=0;
while $n < N$ **do**
  Randomly select one local boundary minority area S_i^{MIN};
  Randomly generate one new data New_{temp} in S_i^{DB};
  Judge whether $New_{temp} \notin S_i^{Maj}$;
  If not, randomly re-generate again (repetition maximum: 100).
  If is, add the new synthetic sample to S_{new}, n=n+1;
end while
return S_{new}

IV. Experimental Results

In this section, we pick three borderline-related methods as B-SMOTE2 [29], ADASYN [24] and MWMOTE [30], and other three state-of-the-art methods as INOS [26], SWIM [31] and GDO [32], for comparisons in this paper. First, 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 UCI machine learning repository [33] and [34]; and carry statistical hypothesis tests for those methods. Finally, we analyse different performances and corresponding characteristics of picked methods.

A. Synthetic data in 2D space

As seen in Fig. 5, we generate synthetic samples for above picked methods in three 2-D datasets. Three 2-D datasets are respectively as Circle dataset, Triangle dataset and lappedCircle dataset. Specially, for Triangle dataset and lappedCircle dataset, we add them with several outliers. Each row corresponding to one dataset. In a row, the original data is first plotted where black denotes majority samples and red denotes minority samples, then synthetic samples of each method are plotted. For each method, $n_{maj} - n_{min}$ synthetic samples are generated, 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 of outliers and almost generates all synthetic samples in the decision boundary that to form a hollow structure as seen in Fig. 5. DBO. In the one hand, this implies that DBO takes full use of information in the decision boundary including both boundary individual samples and their surrounding areas. In the other hand, DBO can well compute the boundary minority area and boundary majority area for different classes.

B. Comparison on real-world benchmark datasets

The performance of each method is evaluated on real-world benchmark datasets from UCI repository[33] and [34]. The basic information on those datasets is seen in Table I. Before

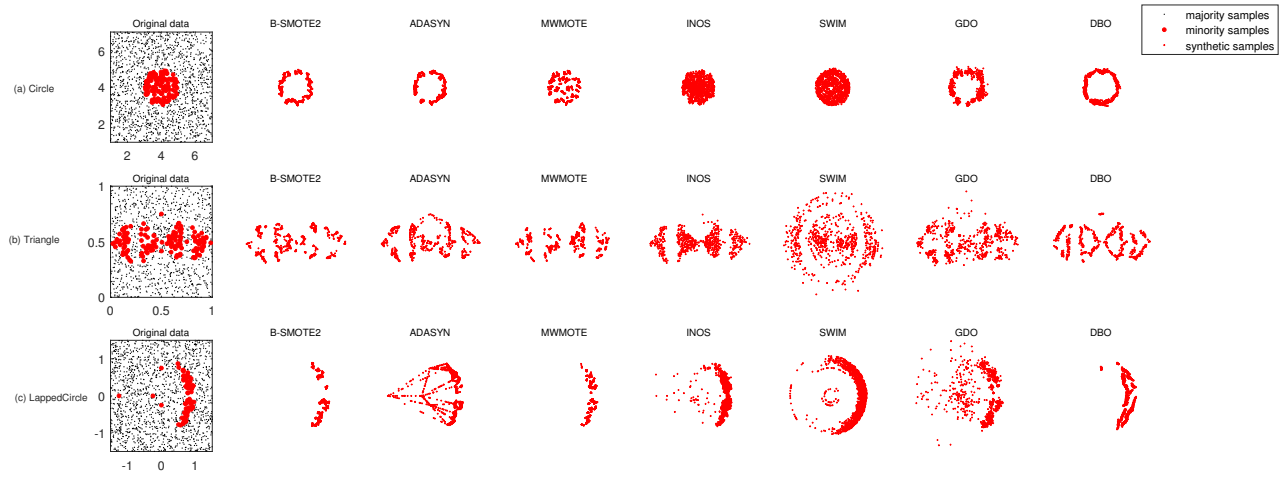

Fig. 5. Synthetic data in 2-D space with three emulational datasets including (a) Circle, (b) Triangle, (c) LappedCircle. For each row, the original data distribution is first plotted where red denotes the minority and black denotes the majority; then synthetic data by each over-sampling method are subsequently plotted. First from the perspective of robustness to outliers, all other methods suffer from outliers except MWMOTE; because MWMOTE uses clusters before generating synthetic samples. Then for B-SMOTE2, it is robuster to outliers than remained methods; the reason is that B-SMOTE2 does not generate synthetic samples for the one that all its m nearest neighbours are majority samples. Next, for our method DBO, it is robuster to outliers than other methods except MWMOTE and B-SMOTE2; because for most outliers, their corresponding local decision boundary areas are covered by local boundary majority samples that no synthetic sample generated. Second from the perspective of the number of synthetic samples in the decision boundary, our method DBO owns the largest one; because DBO only generates synthetic samples in the decision boundary and is robust to some of outliers. Then for B-SMOTE2, ADASYN, MWMOTE and GDO, they own the larger number than remained methods; because B-SMOTE2 only selects boundary minority samples for generating synthetic samples; ADASYN, MWMOTE and GDO assigns high weights of selection for boundary minority samples; and for MWMOTE, many slightly interior boundary minority samples are selected, so its number of synthetic samples in decision boundary is smaller than B-SMOTE2, ADASYN and GDO. Special for INOS, it generates several overlapped samples like ADASYN; because it borrows nearly half percentage of synthetic samples from ADASYN; and for the linear interpolation of synthetic samples, INOS chooses 15 nearest minority samples for ADASYN while ADASYN itself only chooses 5 nearest minority samples.

experiment, all datasets are preprocessed by the standardized z-scores. For all methods, $N_{maj} - N_{min}$ synthetic samples are generated for the minority class. Two classifiers including SVM and AdaBoostM1 (Method: AdaBoostM, NLearn: 10, Learners: decision tree) are used. For each classifier, we apply a twofold SKFCV(stratified k-fold cross validation, and setting k=2) for 30 times. In total, 60 runs are conducted. Corresponding mean and standard deviation are recorded 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. Because imbalanced classification cares more about the minority class. Thus, we select g-mean as the measurement to evaluate the classification performance. Besides, precision and recall are also selected for the comparison between different methods.

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ g-mean &= \sqrt{\frac{TP \times TN}{(TP+FN) \times (TN+FP)}} \end{aligned} \tag{27}$$

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 II and III, the performance of each method on two classifiers SVM and AdaBoostM1 are respectively displayed. Where Ori means the classification performance that directly sending the imbalanced dataset to the classifier. 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). For example in experiment, we apply a twofold SKFCV for 30 times for the method DBO on the Balance-scale middle dataset. There are 60 values of g-mean for the DBO on Balance-scale middle dataset, where the mean and standard deviation of 60 values are respectively as 0.3573 and 0.0670. Obviously, DBO with the average g-mean (0.3573) ranks the 4th best. And the best rank is highlighted as bold like B-SMOTE2 on Balance-scale middle dataset.

As shown in Table IV and VI, mean ranks of those method on precision, recall and g-mean are computed for the further comparison. For example in Table IV, we compute the mean rank of DBO on all datasets as 1.84. And the best mean rank is highlighted as bold. 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 in Table IV, the actual value among all methods on g-mean is 105.23 that is larger than the table look-op value 17.04 (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 on recall and g-mean (only consider DBO achieving the best

mean rank). For example in Table IV, the gap of mean rank on g-mean between B-SMOTE@ and DBO is 1.84 (3.68-1.84) that is larger than the critical value 1.67; 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 shown in Table V and VII, 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 method 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. For example in Table V, the p-value is 0.0242 on recall between DBO and SWIM that smaller than 0.05; thus, significant difference exists; in other words, DBO outperforms SWIM; and denote 0.0242 as bold.

TABLE I
BASIC PROPERTIES OF USED REAL-WORLD DATASETS

Data set	*Attri*	*Min;Maj*	*Min: Maj*	*IR*
Balance-scale middle	4	-	49:576	11.8
Biomed diseased	5	-	67:127	1.9
Housing MEDV¿35	13	-	48:458	9.5
Diabetes absent	8	-	268:500	1.9
Iris setosa	4	-	50:100	2.0
Iris virginica	4	-	50:100	2.0
Thyriod hyperfunction	21	-	191:3581	18.7
Vowel 1	10	-	48:480	10.0
Vowel 9	10	-	48:480	10.0
Abalone58	8	5,8;rest	683:3494	5.1
BreastTissue3	8	3;rest	18:88	4.9
BreastTissue4	8	4;rest	16:90	5.6
Ecoli2	8	im;rest	77:259	3.4
Ecoli3	8	pp;rest	52:284	5.5
Glass7	9	7;rest	29:185	6.4
ImageSegmentation1	19	1;rest	330:1980	6.0
LibrasMovement6	90	6;rest	24:336	14.0
LibrasMovement15	90	15;rest	24:336	14.0
Pageblocks45	10	4,5;rest	203:5270	26.0
Pageblocks34	10	3,4;rest	116:5357	46.2
StatlogVehicleSilhouettes4	18	4;rest	199:647	3.3
Vowel 1 2 3	10	-	144:384	2.7
WallFollowingRobotNavigation4	24	4;rest	328:5128	15.6
Wine1	13	1;rest	59:119	2.0
Yeast569	8	5,6,9;rest	115:1369	11.9
GLRCNBI1	698	hyperplasic;rest	55:21	2.619
Colon 1	1908	-	22:40	1.8
Leukemia 1	3571	-	25:47	1.9
Metas 1	4919	-	46:99	2.2
DrivFace3	6399	3;rest	33:573	17.3636
ARBT1	8265	Buddhism;rest	46:544	11.8261

C. Performance analysis

In this subsection, we analyse different performances of methods based on their mean ranks in Table IV and VI. From the perspective of precision and recall seen in Eq. 27, better precision means the smaller number of false positives, where false positives denote majority samples being wrongly classified; better recall means the smaller number of false negatives, where false negatives denote minority samples being wrongly classified. Basically, good recall is connected with bad precision or good precision is connected with bad recall for all methods. Because more synthetic samples that generated near or in the decision boundary may increase the rate of minority recognition but meanwhile decrease the rate of majority recognition.

Roughly, those methods can be divided into two types respectively as the one of better recall such as B-SMOTE2, SWIM and DBO; another one of worse recall such as Ori, MWMOTE and INOS. Obviously, methods in the first type generate more samples in the decision boundary than methods in the second type. For example, B-SMOTE2 generates synthetic samples between majority and minority samples, SWIM generates synthetic samples with the same Mahalanobis distance from the majority class mean as the minority sample and DBO just generates synthetic samples in the boundary minority area. Specially for SWIM, it may generate synthetic samples along the borderline for the boundary minority sample when its Mahalanobis distance meets the covariance of majority class. And for the second type of methods, although MWMOTE picks out hard to learn minority samples but in which many slightly interior one are also picked. INOS generates one percentage of synthetic samples by self and borrows another percentage of synthetic samples from ADASYN, thus tend to be affected by ADASYN.

Different from above two types of method, ADASYN is more sensitive to outliers and tends to generate many overlapped synthetic samples. Moreover for DBO, it uses the Gaussian distribution of each minority sample to generate synthetic sample; thus it tends to generate many overlapped samples when its corresponding (μ, σ) of Gaussian distribution does not match current data distribution.

From the perspective of g-mean as seen in Eq. 27, better g-mean means the good recognition rate of both minority and majority classes. As seen in Table IV and VI, DBO achieves the best mean rank on g-mean. This implies DBO can well perform the balance on both recognitions rates of minority and majority samples to cope with the imbalanced problem.

V. CHARACTERISTICS OF DBO

A. Robust to outliers

As seen in Fig. 6, 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; specially 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 of outliers and only generates synthetic samples in very near regions or temporary non-majority existed regions for other outliers.

B. Parameter setting and time consuming

As seen in Fig. 7, one parameter K which means K nearest majority samples of the target boundary minority sample is involved in DBO. Obviously, larger value of K means larger sizes of both the local decision boundary area and local boundary minority area. To maintain the local property for DBO, we set K=5 for all datasets.

As shown in Table VIII, the time consuming of different methods is displayed. In experiment, we run the code of eigen decomposition on NVIDIA GeForce GTX 1050Ti and remained code on Intel Core i9 CPU for the last five high-dimension datasets as Colon 1, Leukemia, Metas 1, DrivFace3 and ARBT1. For remained datasets, we run the code on

TABLE II
SVM: AVERAGE G-MEAN ON REAL-WORLD DATASETS

Dataset	*Ori*	*B-SMOTE2*	*ADASYN*	*MWMOTE*	*INOS*	*SWIM*	*GDO*	*DBO*
Balance-scale middle	0.0000±0.0000(8)	0.3919±0.0666(3)	**0.4026±0.0696(1)**	0.3987±0.0749(2)	0.3007±0.1052(7)	0.3468±0.0727(6)	0.3546±0.0686(5)	0.3673±0.0670(4)
Biomed diseased	0.8336±0.0444(8)	0.8581±0.0375(4)	**0.8624±0.0344(1)**	0.8587±0.0367(3)	0.8537±0.0376(7)	0.8572±0.0413(5)	0.8572±0.0377(6)	0.8596±0.0326(2)
Housing MEDV¿35	0.6803±0.0784(8)	0.8160±0.0503(6)	0.8327±0.0545(4)	0.7996±0.0702(7)	0.8272±0.0587(5)	**0.8593±0.0343(1)**	0.8524±0.0397(3)	0.8591±0.0398(2)
Diabetes absent	0.6267±0.0340(8)	**0.7119±0.0218(1)**	0.7055±0.0247(4)	0.7058±0.0227(3)	0.6904±0.0272(7)	0.7004±0.0220(5)	0.6980±0.0255(6)	0.7075±0.0246(2)
Iris setosa	0.9906±0.0102(6)	0.9906±0.0102(6)	0.9906±0.0102(6)	0.9906±0.0102(6)	0.9906±0.0102(6)	0.9923±0.0099(2)	0.9909±0.0101(3)	**1.0000±0.0000(1)**
Iris virginica	0.9076±0.0523(8)	0.9484±0.0365(6)	0.9598±0.0287(2)	0.9523±0.0313(4)	0.9494±0.0347(5)	0.9398±0.0365(7)	0.9530±0.0269(3)	**0.9637±0.0222(1)**
Thyriod hyperfunction	0.0000±0.0000(8)	0.7913±0.0544(3)	0.8444±0.0371(2)	0.7633±0.0456(5)	0.6704±0.0689(7)	0.6947±0.0523(6)	0.7649±0.0364(4)	**0.8726±0.0425(1)**
Vowel 1	0.0978±0.1579(8)	0.7857±0.0614(7)	0.8151±0.0465(6)	0.8328±0.0497(4)	0.8261±0.0528(5)	0.8428±0.0413(2)	0.8417±0.0420(3)	**0.8525±0.0296(1)**
Vowel 9	0.6475±0.0877(8)	0.8645±0.0676(7)	0.8940±0.0544(5)	0.8689±0.0737(6)	0.9013±0.0469(4)	0.9078±0.0315(2)	0.9046±0.0359(3)	**0.9096±0.0263(1)**
Abalone58	0.0000±0.0000(8)	0.6676±0.0124(5)	**0.6780±0.0111(1)**	0.6608±0.0161(6)	0.6536±0.0183(7)	0.6709±0.0155(3)	0.6690±0.0123(4)	0.6739±0.0121(2)
BreastTissue3	0.0000±0.0000(8)	0.6190±0.1035(5)	0.6251±0.1373(4)	0.6188±0.1079(6)	0.6071±0.1355(7)	**0.6680±0.0946(1)**	0.6346±0.0921(3)	0.6482±0.1006(2)
BreastTissue4	0.0292±0.1170(8)	0.7649±0.1327(7)	0.7957±0.0765(3)	0.7781±0.0739(6)	0.7935±0.0845(4)	0.7972±0.0428(2)	0.7892±0.0728(5)	**0.7977±0.0547(1)**
Ecoli2	0.7972±0.0448(8)	0.8698±0.0341(2)	0.8611±0.0337(3)	0.8429±0.0335(6)	0.8415±0.0327(7)	0.8503±0.0267(5)	0.8610±0.0367(4)	**0.8750±0.0280(1)**
Ecoli3	0.7224±0.0643(8)	0.8759±0.0303(3)	0.8804±0.0280(2)	0.8668±0.0368(6)	0.8663±0.0374(7)	0.8675±0.0329(5)	0.8735±0.0326(4)	**0.8837±0.0295(1)**
Glass7	0.8834±0.0462(8)	0.8974±0.0474(6)	0.9057±0.0516(4)	0.8950±0.0502(7)	0.9088±0.0407(3)	**0.9205±0.0334(1)**	0.9025±0.0511(5)	0.9204±0.0322(2)
ImageSegmentation1	0.9842±0.0054(7)	0.9925±0.0042(3)	0.9921±0.0056(4)	0.9903±0.0066(6)	**0.9931±0.0043(1)**	0.9917±0.0038(5)	0.9822±0.0132(8)	0.9930±0.0036(2)
LibrasMovement6	0.6386±0.1095(8)	0.8073±0.0872(3)	0.7789±0.1034(4)	0.7655±0.1195(6)	0.7742±0.0997(5)	**0.8286±0.0631(1)**	0.7639±0.1063(7)	0.8199±0.0748(2)
LibrasMovement15	0.6265±0.0939(8)	0.8056±0.0981(2)	0.7934±0.0953(4)	0.7974±0.0992(3)	0.7278±0.0993(7)	0.7622±0.0846(6)	0.7835±0.0993(5)	**0.8148±0.0842(1)**
Pageblocks45	0.5334±0.0386(8)	0.8866±0.0523(5)	0.9158±0.0319(2)	0.9085±0.0248(3)	0.8760±0.0357(6)	0.9051±0.0156(4)	0.8666±0.0792(7)	**0.9289±0.0159(1)**
Pageblocks34	0.7223±0.0834(8)	0.9647±0.0181(2)	0.9603±0.0173(4)	0.9467±0.0255(7)	0.9587±0.0243(5)	0.9536±0.0202(6)	0.9621±0.0232(3)	**0.9659±0.0126(1)**
StatlogVehicleSilhouettes4	0.9184±0.0209(8)	**0.9625±0.0095(1)**	0.9543±0.0136(5)	0.9502±0.0164(7)	0.9539±0.0142(6)	0.9604±0.0089(3)	0.9592±0.0099(4)	0.9612±0.0101(2)
Vowel 1 2 3	0.8071±0.0350(8)	0.8635±0.0275(3)	0.8622±0.0276(4)	0.8468±0.0270(7)	0.8494±0.0198(6)	0.8568±0.0212(5)	**0.8696±0.0244(1)**	0.8645±0.0220(2)
WallFollowingRobotNavigation4	0.4680±0.0810(8)	0.9004±0.0115(3)	0.8587±0.0223(6)	0.8928±0.0130(4)	0.8914±0.0123(5)	**0.9139±0.0088(1)**	0.8429±0.0171(7)	0.9077±0.0082(2)
Wine1	0.9828±0.0208(5)	0.9811±0.0220(7)	0.9828±0.0208(5)	0.9828±0.0208(5)	**0.9880±0.0137(1)**	0.9858±0.0154(3)	0.9790±0.0199(8)	0.9860±0.0130(2)
Yeast569	0.5626±0.0691(8)	0.8642±0.0224(3)	0.8555±0.0244(6)	0.8534±0.0272(7)	0.8570±0.0251(5)	**0.8726±0.0195(1)**	0.8623±0.0202(4)	0.8702±0.0207(2)
GLRCNBI1	0.6776±0.1241(5)	**0.7500±0.1208(1)**	0.6776±0.1241(5)	0.6776±0.1241(5)	0.6543±0.1202(8)	0.7192±0.0926(3)	0.6774±0.1242(7)	0.7276±0.1120(2)
Colon 1	0.5708±0.1250(5.5)	0.6928±0.1067(2)	0.5708±0.1250(5.5)	0.5708±0.1250(5.5)	0.5696±0.1247(8)	0.6432±0.1246(3)	0.5708±0.1250(5.5)	**0.7949±0.0953(1)**
Leukemia 1	0.7277±0.1174(7)	0.8670±0.1000(2)	0.7277±0.1174(7)	0.7277±0.1174(7)	0.7291±0.1141(5)	0.8582±0.0860(3)	0.7296±0.1170(4)	**0.9523±0.0427(1)**
Metas 1	0.2454±0.1321(6)	0.4186±0.1223(2)	0.2454±0.1321(6)	0.2454±0.1321(6)	0.2452±0.1304(8)	0.3084±0.1342(3)	0.2465±0.1328(4)	**0.4497±0.0841(1)**
DrivFace3	0.7057±0.0983(6)	0.8702±0.0838(2)	0.7057±0.0983(6)	0.7057±0.0983(6)	0.6921±0.0998(8)	**0.8712±0.0695(1)**	0.7092±0.0997(4)	0.8695±0.0727(3)
ARBT1	0.4037±0.2243(6)	0.5359±0.2219(2)	0.4037±0.2243(6)	0.4037±0.2243(6)	0.4225±0.2201(3)	**0.5819±0.0796(1)**	0.4138±0.2313(4)	0.4020±0.2169(8)

For each table cell, the average value of evaluation metric is first recorded, the corresponding standard deviation is followed and the rank among methods is recorded in a bracket. And the best rank for each row is highlight as bold.

TABLE III
ADABOOSTM1: AVERAGE G-MEAN ON REAL-WORLD DATASETS

Dataset	*Ori*	*B-SMOTE2*	*ADASYN*	*MWMOTE*	*INOS*	*SWIM*	*GDO*	*DBO*
Balance-scale middle	0.0000±0.0000(7)	**0.0880±0.1135(1)**	0.0286±0.0821(4)	0.0818±0.1258(2)	0.0047±0.0362(5)	0.0000±0.0000(7)	0.0000±0.0000(7)	0.0578±0.1138(3)
Biomed diseased	0.8317±0.0418(7)	0.8340±0.0394(6)	**0.8398±0.0372(1)**	0.8391±0.0377(2)	0.8385±0.0379(3)	0.8352±0.0339(5)	0.8246±0.0435(8)	0.8383±0.0327(4)
Housing MEDV¿35	0.7793±0.0578(6)	0.8655±0.0469(3)	0.8672±0.0494(2)	0.8478±0.0541(5)	0.8511±0.0599(4)	0.5185±0.1826(8)	0.6392±0.1797(7)	**0.8817±0.0469(1)**
Diabetes absent	0.5839±0.0545(8)	0.6662±0.0433(3)	0.6696±0.0381(2)	**0.6743±0.0400(1)**	0.6429±0.0377(5)	0.6188±0.0471(6)	0.5951±0.0807(7)	0.6455±0.0556(4)
Iris setosa	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	**0.3487±0.4792(1)**	0.0498±0.2191(2)
Iris virginica	0.9054±0.1713(8)	0.9190±0.1234(5)	0.9075±0.1722(7)	0.9081±0.1721(6)	0.9250±0.1238(4)	**0.9413±0.0270(1)**	0.9286±0.1240(2)	0.9265±0.1239(3)
Thyriod hyperfunction	0.1163±0.3228(7)	0.9571±0.0040(4)	0.9842±0.0084(2)	0.9814±0.0083(3)	**0.9905±0.0160(1)**	0.1753±0.1390(6)	0.0599±0.0783(8)	0.9529±0.0047(5)
Vowel 1	0.4126±0.2046(8)	0.7701±0.0947(7)	0.8051±0.0764(4)	0.7990±0.0778(5)	0.8206±0.0670(3)	0.7757±0.0961(6)	0.8288±0.0645(2)	**0.8325±0.0792(1)**
Vowel 9	0.6198±0.1153(8)	0.8265±0.0739(6)	0.8451±0.0616(3)	0.8166±0.0769(7)	0.8284±0.0640(5)	**0.8550±0.0542(1)**	0.8531±0.0618(2)	0.8419±0.0534(4)
Abalone58	0.0000±0.0000(7)	0.2032±0.2445(2)	0.0000±0.0000(7)	**0.3464±0.2448(1)**	0.1055±0.1360(3)	0.0224±0.0846(4)	0.0000±0.0000(7)	0.0096±0.0747(5)
BreastTissue3	0.1430±0.1948(8)	0.4851±0.1736(2)	0.4751±0.1541(3)	0.4649±0.1661(4)	0.4628±0.1579(5)	0.3956±0.2138(7)	0.4066±0.2018(6)	**0.5332±0.1562(1)**
BreastTissue4	0.5796±0.1889(8)	0.7279±0.1598(5)	0.7705±0.0843(3)	0.7577±0.0993(4)	0.7891±0.0865(2)	0.6413±0.2365(7)	0.7275±0.1355(6)	**0.7905±0.1422(1)**
Ecoli2	0.7727±0.0578(8)	0.8535±0.0406(4)	0.8614±0.0336(2)	0.8540±0.0399(3)	0.8444±0.0459(5)	0.7789±0.0538(7)	0.7800±0.0723(6)	**0.8754±0.0435(1)**
Ecoli3	0.7713±0.0845(6)	0.8545±0.0448(3)	0.8392±0.0608(5)	**0.8598±0.0419(1)**	0.8551±0.0453(2)	0.7707±0.2227(7)	0.4679±0.4161(8)	0.8437±0.0402(4)
Glass7	0.8954±0.0493(8)	0.9090±0.0395(7)	0.9176±0.0384(5)	0.9143±0.0481(6)	0.9231±0.0338(4)	**0.9263±0.0346(1)**	0.9244±0.0360(3)	0.9251±0.0362(2)
ImageSegmentation1	0.9726±0.0216(6)	0.9845±0.0069(3)	0.9640±0.0197(7)	0.9855±0.0089(2)	0.9748±0.0197(5)	0.9754±0.0174(4)	0.9314±0.1057(8)	**0.9856±0.0046(1)**
LibrasMovement6	0.5098±0.1731(8)	0.7764±0.1091(4)	0.7864±0.0926(3)	0.7131±0.1246(7)	0.7748±0.1093(5)	0.7665±0.0959(6)	0.7890±0.0937(2)	**0.8034±0.0928(1)**
LibrasMovement15	0.6200±0.1367(8)	0.7820±0.1190(4)	0.7987±0.0907(3)	0.7740±0.1224(6)	0.7809±0.0878(5)	0.7718±0.0836(7)	0.8097±0.1011(2)	**0.8412±0.0815(1)**
Pageblocks45	0.7316±0.0544(7)	0.8476±0.0591(4)	0.8818±0.0430(2)	0.8693±0.0405(3)	0.8173±0.0634(5)	0.6940±0.0909(8)	0.7468±0.0981(6)	**0.8910±0.0404(1)**
Pageblocks34	0.8190±0.0443(8)	0.9460±0.0605(2)	0.9447±0.0271(3)	0.9122±0.0378(6)	0.9293±0.0486(4)	0.9180±0.0534(5)	0.9059±0.0762(7)	**0.9584±0.0173(1)**
StatlogVehicleSilhouettes4	0.6905±0.1354(8)	0.8652±0.0447(6)	0.8721±0.0325(4)	0.8525±0.0465(7)	0.8698±0.0427(5)	0.8788±0.0369(3)	**0.8891±0.0480(1)**	0.8884±0.0431(2)
Vowel 1 2 3	0.7211±0.0714(8)	0.8382±0.0510(5)	0.8344±0.0494(7)	0.8429±0.0415(4)	0.8524±0.0454(2)	**0.8552±0.0479(1)**	0.8364±0.0425(6)	0.8517±0.0471(3)
WallFollowingRobotNavigation4	0.9244±0.0343(8)	0.9761±0.0097(2)	**0.9779±0.0095(1)**	0.9555±0.0126(6)	0.9621±0.0113(5)	0.9534±0.0120(7)	0.9626±0.0090(4)	0.9726±0.0092(3)
Wine1	0.9633±0.0263(8)	0.9664±0.0196(4)	0.9643±0.0241(7)	0.9659±0.0233(6)	0.9692±0.0223(3)	0.9699±0.0198(2)	0.9661±0.0183(5)	**0.9719±0.0144(1)**
Yeast569	0.5235±0.2316(6)	0.8201±0.0315(2)	0.8108±0.0345(5)	0.8190±0.0314(3)	0.8122±0.0287(4)	0.4193±0.2966(7)	0.2679±0.1547(8)	**0.8238±0.0350(1)**
GLRCNBI1	0.7526±0.1117(5)	0.7787±0.0899(3)	0.7794±0.1029(2)	**0.7953±0.0916(1)**	0.7695±0.1117(4)	0.7226±0.1036(8)	0.7493±0.1005(6)	0.7433±0.1103(7)
Colon 1	0.6991±0.0941(8)	0.7310±0.0862(4)	0.7299±0.0861(5)	0.7026±0.0963(7)	0.7391±0.0798(3)	0.7535±0.0725(2)	0.7027±0.0933(6)	**0.7579±0.0774(1)**
Leukemia 1	0.4643±0.4852(7)	0.6826±0.4346(4)	0.4607±0.4813(8)	0.4645±0.4854(6)	0.5566±0.4756(5)	**0.9199±0.0489(1)**	0.8518±0.2889(2)	0.7972±0.3607(3)
Metas 1	0.4247±0.1078(8)	0.5205±0.0638(2)	0.4981±0.0882(4)	0.4871±0.0912(7)	0.4918±0.0866(6)	**0.5245±0.0929(1)**	0.4954±0.1029(5)	0.5011±0.0853(3)
DrivFace3	0.6823±0.1091(8)	0.8288±0.0792(2)	0.7845±0.0921(6)	0.7699±0.0958(7)	0.8007±0.0850(4)	**0.8493±0.0712(1)**	0.8217±0.0812(3)	0.8000±0.0854(5)
ARBT1	0.4696±0.0875(5)	0.4700±0.1316(4)	**0.5130±0.0981(1)**	0.4899±0.1090(2)	0.4723±0.1014(3)	0.3097±0.1306(7)	0.4676±0.1091(6)	0.2858±0.1405(8)

For each table cell, the average value of evaluation metric is first recorded, the corresponding standard deviation is followed and the rank among methods is recorded in a bracket. And the best rank for each row is highlight as bold.

TABLE IV
SVM: MEAN RANKS OF RECALL, PRECISION AND G-MEAN

Measurement	Actual value(Friedman test)	Ori	B-SMOTE2	ADASYN	MWMOTE	INOS	SWIM	GDO	DBO
precision	57.21(reject)	**3.10**	4.73	4.48	3.10	3.27	5.63	5.61	6.08
recall	125.31(reject)	7.55†	3.82†	4.23†	5.68†	5.97†	3.21	3.94†	**1.61**
g-mean	105.23(reject)	7.40†	3.68†	4.11†	5.40†	5.65†	3.29	4.63†	**1.84**
the Friedman Test: F=14.07, (n=8-1,alpha=0.05)									
the Bonferroni-Dunn test: critical values=1.67									

TABLE V
SVM: WILCOXON PAIRED SIGNED-RANK TEST FOR PAIRWISE COMPARISONS

recall		g-mean	
Ours vs.	p-V	Ours vs.	p-V
SWIM	**0.0242**	SWIM	**0.0042**

TABLE VI
ADABOOSTM1: MEAN RANKS OF RECALL, PRECISION AND G-MEAN

Measurement	Actual value(Friedman test)	Ori	B-SMOTE2	ADASYN	MWMOTE	INOS	SWIM	GDO	DBO
precision	43.18(reject)	**2.69**	4.76	5.05	3.37	4.08	5.44	4.61	6.00
recall	80.02(reject)	7.37†	3.55	4.24†	4.63†	4.40†	4.69†	5.00†	**2.11**
g-mean	65.13(reject)	7.27†	3.82	3.98	4.37†	4.02	4.79†	5.06†	**2.68**
the Friedman Test: F=14.07, (n=8-1,alpha=0.05)									
the Bonferroni-Dunn test: critical values=1.67									

TABLE VII
ADABOOSTM1: WILCOXON PAIRED SIGNED-RANK TEST FOR PAIRWISE COMPARISONS

recall		g-mean	
Ours vs.	p-V	Ours vs.	p-V
B-SMOTE2	**0.0093**	B-SMOTE2	0.1124
		ADASYN	**0.0073**
		INOS	**0.0108**

Intel Core i7 CPU. Obviously, DBO costs many seconds for last four high-dimension datasets as Leukemia, Metas 1, DrivFace3 and ARBT1; costs several time for the dataset as WallFollowingRobotNavigation4 with the middle dimension (as 24) and large numbers of minority and majority samples (respectively as 328 and 5128). For rest datasets, DBO costs very few time.

VI. CONCLUSION

In this paper, a novel Decision Boundary Computation-based Oversampling(DBO) method is proposed to address the imbalanced problem to take full use of information in decision boundary. First DBO computes the decision boundary area and the boundary majority area; and then obtains 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 divide the decision boundary area into two partitions for majority and minority classed. And experimental results on real-world datasets show the good performance on recall and g-mean when compared to other methods. Especially on recall, DBO can greatly enhance the rate of minority recognition.

In the future, some works will be attached to improve the robustness towards outliers and good structure representation of the boundary majority area to decrease the risk of much lose on precision.

REFERENCES

[1] 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. PP, pp. 1–14, 2017.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] A. Manukyan and E. Ceyhan, "Classification of imbalanced data with a

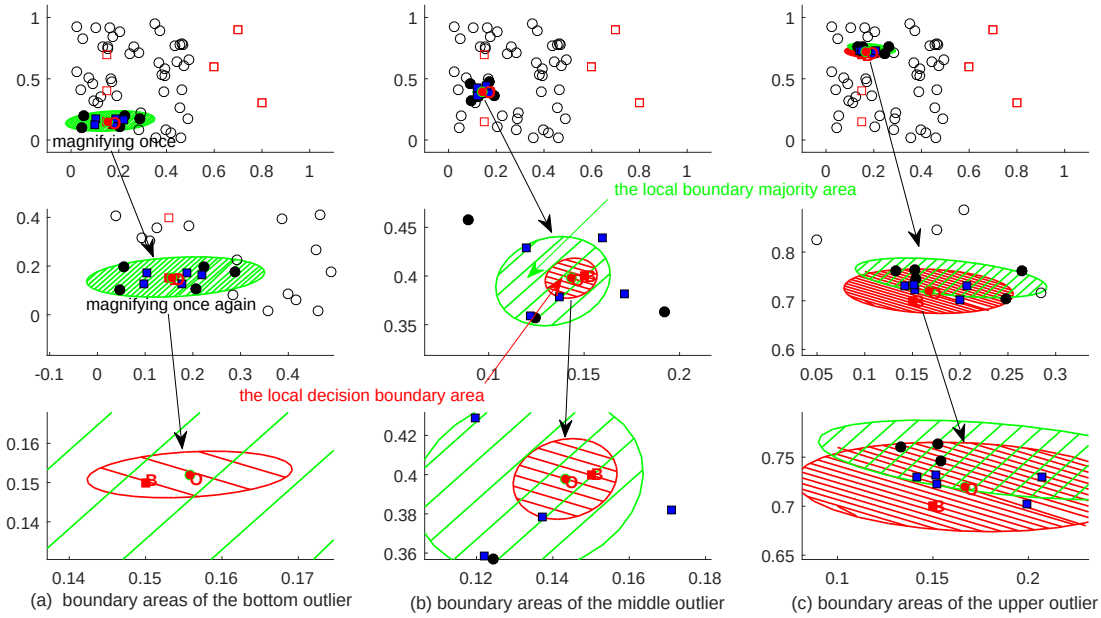

Fig. 6. Robustness of DBO to outliers. The black circle denotes majority samples and the red square denotes minority samples in the original data, the solid red square denotes the target boundary minority sample B, the solid black circle denote k nearest majority samples for B, the solid blue square denotes mean points between B and its k nearest boundary majority samples, the solid red circle denotes the center of those mean points and B, the red slash area denotes the local decision boundary area and the blue slash area denotes the local boundary majority area. For the first and second outlier, no local boundary minority area exists for the reason that the local boundary majority area covers the whole local decision boundary area. For the third outlier, middle size of local boundary minority area exists for the reason that its local boundary majority area covers nearly half of corresponding local decision boundary area. Obviously, k nearest majority samples of the third outlier are of the same side when compared to it, while k nearest majority samples of the first and second outlier are surrounding them. In other word, the remained local boundary minority area of the third outlier distributes in the non-majority existed region (a certain region that no majority samples existed).

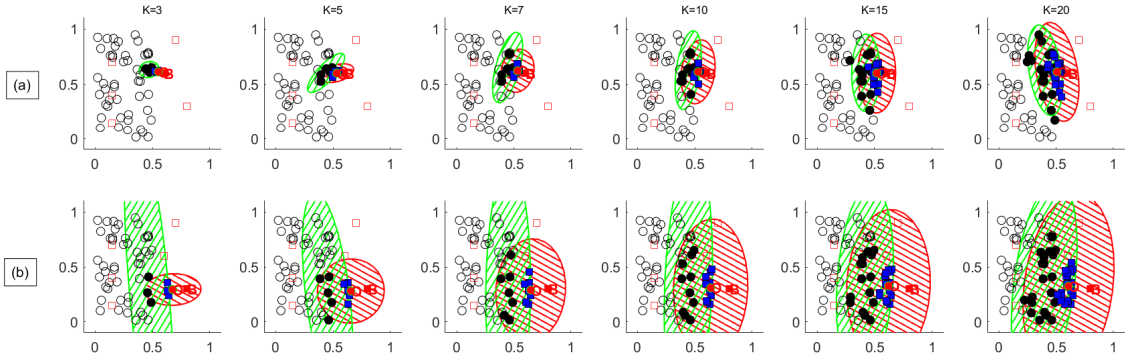

Fig. 7. Parameter setting of DBO. One parameter K which means K nearest majority samples of the target boundary minority sample is involved in DBO. Different K values are tested for two boundary minority samples respectively in (a) and (b). Obviously, larger the value of K is, larger sizes of both the local decision boundary area and local boundary minority area are. To maintain the local property for DBO, we set K=5 for all datasets.

geometric digraph family," *J. Mach. Learn. Res.*, vol. 17, pp. 1–40, Jan. 2016.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote—majority weighted minority oversampling technique for imbalanced data set

TABLE VIII
COMPUTATION TIME(SECONDS)

Dataset	B-SMOTE2	ADASYN	MWMOTE	INOS	SWIM	GDO	DBO
Balance-scale middle	0.00	0.02	0.06	0.15	0.00	0.01	0.07
Biomed diseased	0.00	0.03	0.05	0.00	0.00	0.00	0.01
Housing MEDV¿35	0.00	0.02	0.04	0.03	0.00	0.01	0.03
Diabetes absent	0.00	0.14	0.30	0.02	0.00	0.01	0.03
Iris setosa	0.00	0.02	0.03	0.00	0.00	0.00	0.00
Iris virginica	0.00	0.02	0.04	0.00	0.00	0.00	0.00
Thyriod hyperfunction	0.01	0.08	0.17	1.31	0.00	0.12	0.37
Vowel 1	0.00	0.03	0.06	0.05	0.00	0.01	0.03
Vowel 9	0.00	0.03	0.06	0.05	0.00	0.01	0.03
Abalone58	0.03	3.82	6.10	8.02	0.01	0.24	0.33
BreastTissue3	0.00	0.01	0.02	0.01	0.00	0.00	0.01
BreastTissue4	0.00	0.01	0.02	0.01	0.00	0.00	0.01
Ecoli2	0.01	0.03	0.07	0.01	0.00	0.01	0.03
Ecoli3	0.01	0.02	0.04	0.01	0.00	0.01	0.03
Glass7	0.00	0.01	0.02	0.01	0.00	0.00	0.02
ImageSegmentation1	0.84	0.42	0.90	3.44	0.02	0.06	0.20
LibrasMovement6	0.00	0.02	0.03	0.31	0.01	0.15	0.54
LibrasMovement15	0.00	0.02	0.04	0.50	0.01	0.15	0.55
Pageblocks45	0.09	3.76	6.75	17.87	0.01	4.41	0.61
Pageblocks34	0.05	1.48	2.74	14.92	0.01	3.39	0.59
StatlogVehicleSilhouettes4	0.12	0.19	0.40	0.81	0.01	0.02	0.08
Vowel 1 2 3	0.11	0.29	0.37	0.10	0.01	0.01	0.04
WallFollowingRobotNavigation4	1.29	0.54	1.46	23.14	0.03	6.01	18.42
Wine1	0.01	0.02	0.04	0.01	0.00	0.01	0.02
Yeast569	0.00	0.09	0.17	0.27	0.00	0.03	0.10
GLRCNBI1	0.00	0.02	0.04	0.18	0.04	0.12	0.83
Colon 1	0.00	0.01	0.02	0.47	0.16	0.27	2.31
Leukemia 1	0.00	0.02	0.03	2.81	1.23	1.46	17.73
Metas 1	0.00	0.05	0.07	9.43	2.60	7.50	48.10
DrivFace3	0.05	0.16	0.29	148.54	5.09	144.00	156.84
ARBT1	0.06	0.27	0.37	324.50	9.85	272.35	383.70

learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[31] 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.

[32] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, "Gaussian distribution based oversampling for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, pp. 2020, doi: 10.1109/TKDE.2020.2 985 965.

[33] 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.

[34] "One-class classifier results." [Online]. Available: http://homepage.tudelft.nl/n9d04/occ/index.html

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 imbalance learning.

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.

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.