Hindawi
Complexity
Volume 2021, Article ID 6877284, 9 pages
https://doi.org/10.1155/2021/6877284

Research Article

HSDP: A Hybrid Sampling Method for Imbalanced Big Data Based on Data Partition

Liping Chen, Jiabao Jiang, and Yong Zhang

School of Information Engineering, Chaohu University, Chaohu, China

Correspondence should be addressed to Liping Chen; clp_luck@163.com

Received 30 April 2021; Accepted 6 June 2021; Published 22 June 2021

Academic Editor: Huihua Chen

The classical classifiers are ineffective in dealing with the problem of imbalanced big dataset classification. Resampling the datasets and balancing samples distribution before training the classifier is one of the most popular approaches to resolve this problem. An effective and simple hybrid sampling method based on data partition (HSDP) is proposed in this paper. First, all the data samples are partitioned into different data regions. Then, the data samples in the noise minority samples region are removed and the samples in the boundary minority samples region are selected as oversampling seeds to generate the synthetic samples. Finally, a weighted oversampling process is conducted considering the generation of synthetic samples in the same cluster of the oversampling seed. The weight of each selected minority class sample is computed by the ratio between the proportion of majority class in the neighbors of this selected sample and the sum of all these proportions. Generation of synthetic samples in the same cluster of the oversampling seed guarantees new synthetic samples located inside the minority class area. Experiments conducted on eight datasets show that the proposed method, HSDP, is better than or comparable with the typical sampling methods for F-measure and G-mean.

1. Introduction

In the era of big data, tremendous amount of data generated by various real-world applications brings the challenges to data mining. Among the challenges, classification of imbalanced datasets has drawn interest in various application areas. A dataset is imbalanced when the number of samples in one category is much less than the number of samples in other categories. If the samples come from two classes, the data of the larger number is called the majority class, and the data of the smaller number is called the minority class. Our research focuses on the binary classification problem (the two-class classification) and the prediction of minority samples is more important, because the cost of misclassification for minority samples is greater than the cost of misclassification for majority samples. The issue of binary classification of imbalanced data exists in various applications, such as medical diagnosis.

Most classifiers aim at maximizing the overall classification accuracy of a dataset. Therefore, when classifying imbalanced data, the classifier is biased to meet the classification accuracy of the majority samples, causing low classification accuracy over the minority class. In addition, an imbalanced dataset combination with other difficulty factors such as class overlapping, presence of outliers, and small disjunctions will be more difficult for the classifier to predict minority class [1]. Figure 1(a) shows the skewness distribution between classes. Figure 1(b) shows class overlapping, and Figure 1(c) shows the small disjunctions of minority class. Therefore, how to improve the classification accuracy of minority samples while ensuring the overall classification performance of the classifier for imbalance data is an urgent problem to be solved.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 describes the proposed HSDP method. Section 4 introduces the experimental settings. Section 5 presents the experimental results and compares our approach with some typical techniques. Finally, the conclusion is drawn in Section 6.

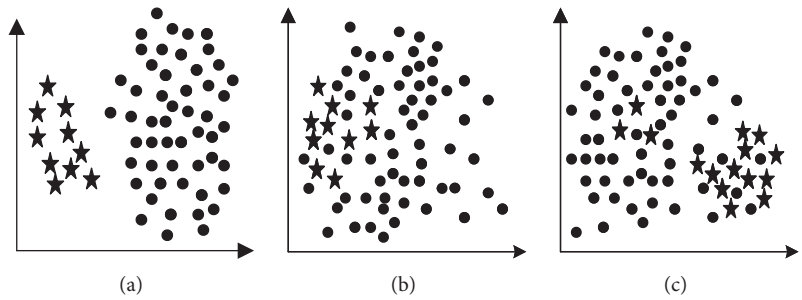

Figure 1: Imbalanced date distribution. (a) Skewness distribution. (b) Class overlapping. (c) Small disjunctions.

2. Related Works

The techniques proposed to improve classification for imbalanced data can be categorized into two major groups: data-level methods and algorithm-level methods. The algorithm-level methods modify the classifier in order to improve the accuracy of imbalanced data. The algorithm-level methods mainly include cost-sensitive methods and ensemble learning methods. The data-level methods mainly include undersampling for the majority class [2], oversampling for the minority class. In contrast, the data-level methods are conducive to enhancing the generalization ability of the model, and the oversampling methods have more advantages because they do not lose the data sample information [3, 4]. SMOTE is the one of most popular oversampling algorithms [5]. SMOTE first selects a random seed x from the minority samples and then randomly selects sample y among its k neighbors in the same class, Finally, a new synthetic sample s is generated by the linear interpolation. This can be expressed as

$$s = x + \text{gap} \times (x - y), \tag{1}$$

where gap is a random number between 0 and 1.

Although the SMOTE algorithm has shown successful performance in various classification scenes, the SMOTE algorithm also has some weaknesses: (1) if the noisy sample is selected, oversampling process may generate more noisy samples. (2) It does not consider the data distribution when generating the synthetic sample, thereby increasing the overlaps between different classes [6]. (3) It oversamples uninformative minority samples because it chooses a minority sample seed to oversample with uniform probability. However, those minority samples on the boundary area contain more information than ones far from the boundary [7]. Therefore, researchers have proposed some improved versions of SMOTE. The Borderline-SMOTE algorithm [8] oversamples the borderline minority samples. However, sometimes the Borderline-SMOTE generates new synthetic samples in unsuitable areas, such as noise regions and overlapped areas. ADASYN algorithm [9] pays more attention to the minority samples that are difficult to learn. It can adaptively generate minority samples according to the ratio of samples of majority class in the neighborhood samples. The K-means-SMOTE algorithm [10] combats between-class imbalance and within-class imbalance. But it does not provide a strategy for determining the optimal number of clusters k, which has a great impact on the performance of oversampling. The MWMOTE technique [11] analyzes the hard-to-learn minority samples and assign them weights according to their importance in learning.

In summary, the methods above have mitigated some of the problems of SMOTE, but neither of them has effectively solved all the three problems. So, the proposed hybrid sampling method based on data partition attempts to overcome all three problems. It is able to select proper minority samples for oversampling and improve the synthetic sample generation scheme. The generation scheme includes the size of synthetic samples for selected minority samples and the control of the location of the generated samples in data space.

3. Proposed Method

3.1. Overview. The data samples present different distribution characteristics in data space, and the data distribution can be considered when undersampling or oversampling. Different sampling methods are used in different regions that may improve classification performance, and we propose the hybrid sampling method of imbalanced data based on data partitioning (HSDP). The method consists of four stages: (1) partitioning space of the input imbalanced data into five regions; (2) removing the samples in the noise minority samples region; (3) using agglomerative hierarchical clustering method to cluster the minority samples; (4) oversampling process. In the first stage, the data space is divided into five regions: the boundary minority samples region, the noise minority samples region, the safe minority samples region, the boundary majority samples region, and the safe majority samples region. And the first two stages are performed because our aim is to oversample the borderline minority samples while ignoring the noisy minority samples. The basic idea is that the borderline samples are apt to be misclassified. In the third stage, clustering the minority samples is to ensure that the generated samples must be inside the minority class regions. In the fourth stage, the oversampling process is performed, which adaptively generates synthetic samples for borderline minority samples in the same cluster of the oversampling seed. The oversampling process in the same cluster ensures that the generated samples locate inside the minority class regions.

3.2. Data Partition. According to the proportion of minority samples in neighborhoods of each minority sample, the data space is divided into five regions [12]: the boundary minority samples region, the noise minority samples region, the safe minority samples region, the boundary majority samples region, and the safe majority samples region, as shown in Figure 2.

Given an imbalanced training dataset S and minority class label class (min), the training dataset is divided into majority class set S_{maj} and minority class set S_{min} firstly. Then, for each sample x_{i} in the minority class set S_{min}, we calculate k neighbors around x_{i} through the K-nearest neighbor algorithm. Next, in these k neighbors, the number of the minority class samples $N_{b\,\text{min}}$ is computed and the majority class samples are put into the boundary majority samples region S_{maborder}. Finally, by judging $N_{b\,\text{min}}$, each sample is added to corresponding region. If $N_{b\,\text{min}} = 0$, the minority class sample is added to the noise minority samples region S_{miout}. If $N_{b\,\text{min}} = k$, the minority class sample is added to the safe minority samples region S_{misafe}. If $1 < N_{b\,\text{min}} < k$, the minority class sample is added to the boundary minority samples region S_{miborder}. The safe majority samples region S_{masafe} and the set $S$with the sample in S_{miout}removed are determined at the end. The DP algorithm for the data partitioning is described as follows (Algorithm 1):

3.3. Clustering Minority Class Samples Based on Hierarchical Clustering. Most of the existing oversampling methods are the K-NN based approach. To generate a synthetic sample from the minority class sample B and $k = 5$, sample A may be chosen (as shown in Figure 3). By this way, the generation of a synthetic sample (shown by square) may locate in the majority class region.

Our proposed method chooses the sample from the same cluster (Cluster1) of B. It ensures that A will not be chosen, because B and A are not in the same cluster. Thus, the oversampling process is performed in a safe range and the generation of minority samples must locate inside the minority class region.

The hierarchical clustering algorithm is used to cluster the minority class samples in this work. And, the key steps of the agglomerative hierarchical clustering algorithm are described as follows:

(1) Assign each data sample to a cluster initially.

(2) Find the two closest clusters and merge them into a single cluster. And, this will reduce the total number of clusters by one.

(3) Compute the distance between the newly generated cluster and all the previous clusters.

(4) Repeat steps 2-3 until a certain termination condition is reached. The termination condition is the number of clusters set in advance or distance threshold.

However, using agglomerative hierarchical clustering algorithm to cluster the minority class samples, whether the two minority clusters are merged or not, not only the distance between the minority clusters but also the distribution of the majority class samples should be examined. If the distance between two minority class clusters is d_{min}, the distances from a certain majority class cluster to these two minority clusters are d_1 and d_2, respectively; where $d_1 < d_{\text{min}}$ and $d_2 < d_{\text{min}}$. Then, these two minority class clusters cannot been merged.

Therefore, modifications to agglomerative hierarchical clustering algorithm have been made. First, agglomerative hierarchical clustering algorithm is used to cluster the majority class samples to obtain the majority cluster set C_{maj}. Then, the minority class samples are clustered. The minority class cluster algorithm based on hierarchical clustering algorithm [13] (MDH) is described below (Algorithm 2).

Because the distance threshold T_{hi} is the termination condition of the clustering process, the setting T_{hi} is particularly critical. In this work, T_{hi} is computed as follows:

$$d_{\text{avg}} = \frac{1}{\left|S_{\text{min}}\right|} \sum_{x \in S\,\text{min}} \frac{\sum_{y \neq x, y \in S_{\text{min}}} \text{dist}(x, y)}{\left|S_{\text{min}}\right| - 1},$$
$$T_{hi} = d_{\text{avg}} \times r. \tag{2}$$

The parameter d_{avg} represents the average of the distance from each minority class sample to any other minority samples in the Set S. The parameter r is used to tune the output distance of the cluster algorithm. And the specific value analysis of r is discussed in Section 5.

3.4. Description of Hybrid Sampling Algorithm Based on Data Partition. We proposed a hybrid sampling algorithm based on data partition. Firstly, the boundary region can be obtained by DP algorithm. Then, the total number of synthetic samples generated in the boundary region is calculated. Next, the weight g_i of each sample x_i in the boundary minority samples region S_{miborder} is computed by the ratio between the proportion of majority class in the neighbors of this selected sample and the sum of all these proportions. Finally, for each sample x_{i} in the boundary minority samples region S_{miborder}, g_i synthetic samples should be generated in the same cluster of x_i. The hybrid sampling algorithm based on data partition (HSDP) is implemented as follows (Algorithm 3):

3.5. The Time Complexity Analysis of HSDP Algorithm. In the DP algorithm, supposing the number of minority samples is N_{min} and the number of majority samples is N_{maj}, each minority sample needs to calculate the distance from other samples to find neighbor samples. Therefore, the time complexity of calculating the distance between samples is

$$O\left(N_{\text{min}} \times \left(N_{\text{min}-1} + N_{\text{maj}}\right)\right). \tag{3}$$

In the MDH algorithm, the distance between each minority cluster and the majority clusters needs to be calculated. Suppose that the current number of minority clusters is n_{min} and the number of majority clusters is n_{max}. The time complexity of calculating the distance between minority

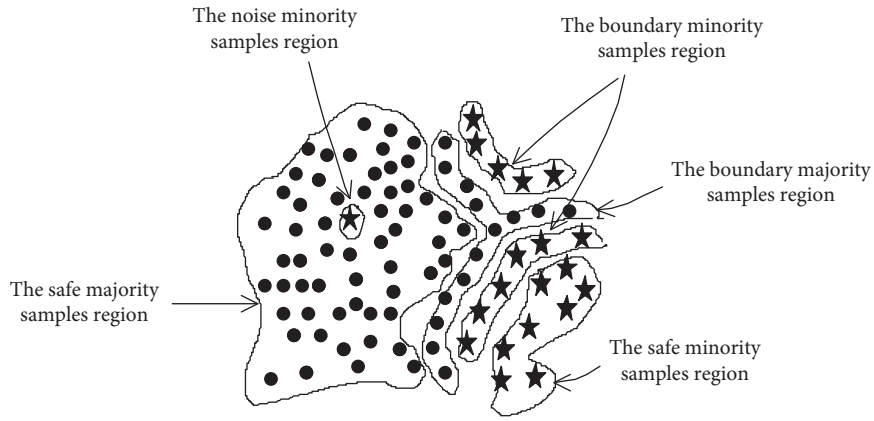

Figure 2: Data space partition.

Input: imbalanced training dataset S, the number of nearest neighbors k, Minority class label class (min)
Output: five regions: S_{misafe}, S_{masafe}, S_{miout}, S_{maborder}, S_{miborder}, and dataset S without the samples in S_{miout}
Procedure:

```
(1)  for x_i in S
(2)    if (class (x_i) = = class (min))//label of sample x_i is the class (min)
(3)      S_min.add(x_i)//add the sample x_i into minority class dataset S_min
(4)    else
(5)      S_maj.add(x_i)//add the sample x_i into majority class dataset S_maj
(6)      end if
(7)  end for
(8)  for x_i in S_min
(9)      list = neighbors(x_i, k)//find k neighbor samples of each minority class samples
(10)     N_bmin = 0//initialize the number of minority class samples
(11)     for z_j in list
(12)       if (class (z_j) = = class (min))//label of sample z_j is the class (min)
(13)         N_bmin + +
(14)       else
(15)         S_maborder.add(z_j)//add majority class sample into boundary majority samples region
(16)         N_bmaj + +
(17)       end if
(18)     end for
(19)     if (N_bmin == k)
(20)         S_misafe.add(x_i)//add minority class sample into the safe minority samples region
(21)     else if (N_bmin == 0)
(22)         S_miout.add(x_i)//add minority class sample into the noise minority samples region
(23)       else
(24)         S_miborder.add(x_i)//add minority class sample in to the boundary minority samples region
(25) end for
(26) for x_i in S_maj
(27)   if (!(x_i in S_maborder))
(28)   S_masafe.add(x_i)//add majority class sample into the safe majority samples region
(29)   end if
(30) end for
(31) for x_i in S
(32)   if (x_i in S_miout)
(33)     S.delete(x_i)//delete samples in the noise minority class region and retain samples from other regions
(34)     end if
(35) end for
```

Algorithm 1: DP Algorithm.

clusters is $O(n_{\min}^2)$, and find the two minority clusters with the smallest distance. Then, calculate the distance between the majority cluster and the two minority clusters with the smallest distance. So, the calculation time complexity to get the distance between the majority cluster and the minority cluster is $O(n_{\max})$, and then determine whether to merge the

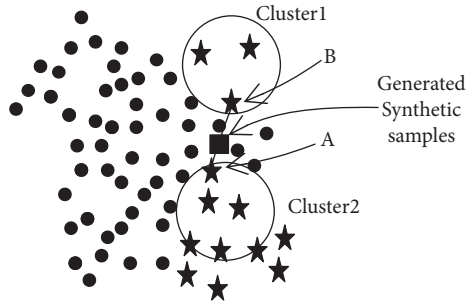

Figure 3: Illustrating of a generated synthetic sample.

Input: majority class cluster $C_{\text{maj}} = \{C_{\text{maj}\,1}, C_{\text{maj}\,2}, \ldots, C_{\text{maj}\,m}\}$, minority class without samples of the noise minority samples region $S = \{x_1, x_2 \ldots x_n\}$, threshold T_{hi}
Output: minority class cluster $C_{\text{min}} = \{C_{\text{min}\,1}, C_{\text{min}\,2}, \ldots, C_{\text{min}\,m}\}$
Process:
(1) $d = 0$;//initialize the minimum distance between clusters
(2) **for** $(i = 0; i < n; i++)$
(3) $C_{\text{min}\,i} = x_i$//initialize the minority class cluster
(4) **end for**
(5) **while** $(d < T_{hi})$
(6) Matrix = $d(C_{\text{min}})$//the distance matrix between clusters
(7) $d = \text{dis}(C_{\text{min}\,i}, C_{\text{min}\,j})$//the minimum distance d and corresponding cluster numbers $C_{\text{min}\,i}$ and $C_{\text{min}\,j}$
(8) **for** C_i **in** C_{maj}//each majority class cluster
(9) **if** $(\text{dis}(C_{\text{min}\,i}, C_i) < d)$ && $(\text{dis}(C_{\text{min}\,j}, C_i)) < d))$
(10) $U(C_{\text{min}\,i}, C_{\text{min}\,j}) =$ not//the flag of cluster mergence is not
(11) **else**
(12) $C_{\text{min}\,i} = C_{\text{min}\,i} \cup C_{\text{min}\,j}$//merge $C_{\text{min}\,i}$and $C_{\text{min}\,j}$into a single cluster
(13) $C_{\text{min}\,j} = \phi$
(14) C_{min}.length − −//reduce the total number of clusters by one
(15) **end if**
(16) **end for**
(17) **end while**

Algorithm 2: MDH Algorithm.

Input: imbalanced dataset S
Output: balanced dataset S
Process:
Step 1: S_{misafe}, S_{miborder}, S_{masafe}, S_{maborder}can be obtained by DP algorithm.
Step 2: count the number (m) of samples in the S_{misafe} and S_{miborder}. Count the number (n) of samples in the S_{masafe} and S_{maborder}. Meanwhile, count the number (s) of samples in the S_{miborder}.
Step 3: calculate the number of synthetic data samples that need to be generated for minority class: $GN = (n - m) \times b$, where $b \in [0, 1]$ is the synthesis scaling factor. $b = 1$ means a balanced dataset is obtained after the oversampling process.
Step 4: for each sample $x_i \in S_{\text{miborder}}$, calculate the ratio of majority class samples belonging to the k neighbors of x_i. This ratio r_i is defined as $r_i = N_i/k, i = 1, 2, \ldots, s$
Step 5: the weight is determined by $w_i = (r_i / \sum_{i=1}^{s} r_i)$.
Step 6: calculate the number of synthetic data samples for each sample x_i in the boundary minority samples regionS_{miborder}: $g_i = w_i \times GN$.
Step 7: for each sample x_{i} in the boundary minority samples region, generate g_i synthetic data samples according to the following steps:
Do the loop from 1 to g_i:
(a) Randomly select another sample y from the same cluster of x_i
(b) Generate a synthetic data sample:
$s = x_i + (y - x_i) \times p$,
where $p \in [0, 1]$
End loop

Algorithm 3: HSDP Algorithm.

two minority clusters according to the distance between the clusters, and the possible number of merger time is $n_{\min}$. Therefore, the time complexity of the MDH algorithm is $O(n_{\min} \times (n_{\min}^2 + n_{\max}))$.

In the HSDP algorithm, assuming that the number of boundary minority samples is N_{miborder}, the number of minority class samples is $N_{\min}$, and the number of majority class samples is N_{maj}, and the time complexity the step of determining the k neighbors of boundary minority samples is $O(N_{\text{miborder}} \times (N_{\min} + N_{\text{maj}}))$. In the step of sample generation, the computational time complexity is $O(N_{\text{miborder}} \times g_i)$.

According to the analysis of the above steps, the time complexity of the HSDP algorithm should be $O(N_{\min} \times (N_{\max} + N_{\min}^2))$.

4. Experiment Setup

4.1. Dataset Description. We test our algorithm on datasets from various filed, including 8 imbalanced datasets. All these datasets are available from KEEL Repository and UCI Repository. Table 1 describes the information of these datasets.

In this study, we research the binary classification. In the two-classification problem, the majority of samples are usually also marked as negative samples, and the minority samples are also marked as positive samples.

4.2. Evaluation Metrics. For the classification problem of imbalanced data, the overall classification accuracy is not suitable for evaluation of classifiers performance, because sometimes a classification algorithm with a better overall accuracy may be at the expense of large prediction error over the minority class. Therefore, *F*-measure and *G*-mean are usually used to evaluate the performance of imbalanced classification algorithms.

F-measure and *G*-mean are calculated based on the confusion matrix, as shown in Table 2.

Based on the confusion matrix, the following equations are derived:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

$$F - \text{measure} = \frac{\left(1 + \beta^2\right) \times \text{Recall} \times \text{Precision}}{\beta^2 \times \text{Recall} + \text{Precision}}, \tag{6}$$

$$G - \text{mean} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \times \frac{\text{TN}}{\text{TN} + \text{FP}}}. \tag{7}$$

F-measure is computed as shown in formula (6). *F*-measure is the harmonic mean between the Recall and Precision. The higher *F*-measure can ensure that both Recall and Precision are higher, where β is a coefficient to indicate the relative importance of Recall and Precision (usually $\beta = 1$). *G*-mean is calculated as shown in formula (7). *G*-mean is the geometric mean of the minority class accuracy and majority class accuracy, and it assigns equal importance to performance of the classifier on minority class and majority class.

5. Experiment and Result Analysis

The experiment platform is Anaconda. Since the purpose is evaluating the proposed sampling method, we do not choose any special classifier; rather, we apply several of them such as KNN and RF. In order to compare the performance of our proposed hybrid sampling method (HSDP) against the other techniques, comparative experiments were carried out, including SMOTE, ADASYN, and Borderline-SMOTE.

5.1. Analysis of Experimental Results. In order to guarantee the fair comparison, the experiment uses a 10-fold cross-validation method. Tables 3–6, respectively, show the F-measure and G-mean values of various algorithms on each dataset.

The best results of F-measure and G-mean are bold faced on each dataset in the above tables. It is evident that the KNN and RF combined with the sampling method are better than themselves without combining sampling method in most cases, On the F-measure, the HSDP algorithm obtained 5 best results on 8 datasets, and 6 best results on G-mean value. This shows that the HSDP algorithm proposed in this paper can improve the classification effect of minority class.

Compared with the SMOTE method and the ADASYN method, (1) the HSDP method does not oversample for all minority class samples but focuses on the minority samples in the boundary area that are more important in classification and (2) the HSDP method removes the noise data, thus avoiding the noisy samples generation.

In contrast to Borderline-SMOTE, our proposed HSDP method not only considers the importance of minority class samples in boundary area but also considers the distribution characteristics of data samples, avoiding any wrong synthetic sample generation.

5.2. Analyzing the Influence of the Parameter Value Used in HSDP Algorithm. The parameters involved in the proposed method (HSDP) include the number of neighbor samples k and the distance adjustment factor r.

The value of k cannot be too small, because this will take the boundary minority class samples as noisy data and delete them by mistake.

The value of r is used to control the number of clusters. With smaller r value, the number of clusters increases and the number of samples decreases in the clusters, which will result in a decrease in diversity when synthesizing samples.

In order to determine the optimal value range of r and k, we use Pima, Glass5, and Yeast3 as the test datasets. For k value ($k = 3, 5, 7, 9$ and 11), the *G*-mean are given as shown in Table 7. For pima dataset, *G*-mean obtains the maximum value when k is 5. When k is 9, the glass5 achieves the maximum *G*-mean value. And Yeast3 achieved the maximum *G*-mean value when k is 7. It is evident that the value of k is appropriate in the range of 5–9. For r value ($r = 0.6$, 0.8, 1.0, 1.2, and 1.6), the *G*-mean are given as shown in Table 8. The Pima dataset achieves the

Table 1: Dataset information.

Dataset	Samples	Attributes	Classes	Imbalance ratio
Pima	768	8	{Negative, positive}	1.87
Yeast3	1484	8	{Negative, positive}	8.1
Abalone19	4174	8	{Negative, positive}	129.44
Segment0	2308	19	{Negative, positive}	6.02
Page-blocks0	5472	10	{Negative, positive}	8.79
Glass5	214	9	{Negative, positive}	22.78
Ecoli4	336	7	{Negative, positive}	15.8
Haberman	306	3	{Negative: "1" other, positive: "2"}	2.78

Table 2: Confusion matrix.

	Predicted as positive	Predicted as negative
Actually positive	TP	FN
Actually negative	FP	TN

Table 3: *F*-measure values of KNN + various sampling methods.

Dataset	Algorithm				
	KNN	SMOTE + KNN	ADASYN + KNN	Borderline-SMOTE + KNN	HSDP (proposed) + KNN
Pima	0.5543	0.5766	0.5824	0.5726	**0.6105**
Yeast3	0.6343	0.6301	0.6294	0.6528	**0.6698**
Abalone19	0.2019	**0.3355**	0.2801	0.2438	0.3045
Segment0	0.8653	0.8734	0.8697	0.8843	**0.8928**
Page-blocks0	0.7146	0.7301	0.7099	**0.7401**	0.6988
Glass5	0.7132	0.7102	0.7129	**0.7265**	0.6827
Ecoli4	0.5401	0.5268	0.6011	0.5701	**0.6698**
Haberman	0.2922	0.3027	0.3417	0.3285	**0.3636**

Table 4: *G*-means values of KNN + various sampling methods.

Dataset	Algorithm				
	KNN	SMOTE + KNN	ADASYN + KNN	Borderline-SMOTE + KNN	HSDP (proposed) + KNN
Pima	0.6503	0.6673	0.6721	0.6635	**0.6932**
Yeast3	0.7536	0.7786	0.7798	0.7824	**0.8034**
Abalone19	0.3165	0.6028	**0.6037**	0.4366	0.5437
Segment0	0.9154	0.9379	0.9440	0.9261	**0.9528**
Page-blocks0	0.8153	0.8629	0.8601	0.8421	**0.8757**
Glass5	0.7887	0.7857	**0.7898**	0.7890	0.7715
Ecoli4	0.7012	0.7401	0.7928	0.7429	**0.8438**
Haberman	0.4502	0.4810	0.5211	0.4983	**0.5201**

Table 5: *F*-measure values of RF + various sampling methods.

Dataset	Algorithm				
	RF	SMOTE + RF	ADASYN + RF	Borderline-SMOTE + RF	HSDP (proposed) + RF
Pima	0.5891	0.5625	0.5632	0.5710	**0.6000**
Yeast3	0.7326	0.7411	0.7214	0.7366	**0.7611**
Abalone19	0.3054	0.3123	0.3652	0.2864	**0.3912**
Segment0	0.9102	**0.9375**	0.8865	0.9292	0.9301
Page-blocks0	0.7443	**0.7745**	0.7586	0.7723	0.7438
Glass5	0.8148	0.7826	0.7931	0.7725	**0.8201**
Ecoli4	0.5701	0.5737	0.5723	**0.5802**	0.5623
Haberman	0.3402	0.4011	0.4386	0.4154	**0.4489**

Table 6: *G*-means values of RF + various sampling methods.

Dataset	Algorithm				
	RF	SMOTE + RF	ADASYN + RF	Borderline-SMOTE + RF	HSDP (proposed) + RF
Pima	0.6173	0.6412	0.6315	0.6385	**0.6479**
Yeast3	0.8243	0.8502	0.8473	0.8362	**0.8532**
Abalone19	0.2782	0.6352	0.6529	0.4564	**0.7001**
Segment0	0.9344	0.9501	0.9313	0.9421	**0.9567**
Page-blocks0	0.8348	0.8990	**0.9078**	0.8846	0.8815
Glass5	0.8366	0.8201	0.8302	0.8172	**0.8401**
Ecoli4	0.6490	0.7442	**0.7751**	0.7403	0.7239
Haberman	0.4659	0.5686	0.5982	0.5743	**0.5991**

Table 7: *G*-mean for different k value.

	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$
Pima	0.6015	**0.6832**	0.6371	0.6483	0.6144
Glass5	0.6986	0.7902	0.8209	**0.8394**	0.6857
Yeast3	0.8351	0.8587	**0.8876**	0.8129	0.7576

Table 8: *G*-mean for different r value.

	$r=0.6$	$r=0.8$	$r=1.0$	$r=1.2$	$r=1.4$
Pima	0.6254	**0.6483**	0.6409	0.6457	0.6347
Glass5	0.7678	0.8045	**0.8246**	0.7892	0.7754
Yeast3	0.7984	0.8269	0.8266	**0.8548**	0.8026

maximum *G*-mean value when r is 0.8. Glass5 achieved the maximum *G*-mean value when r was 1.0. And Yeast3 achieved the maximum *G*-mean value when r was 1.2. It can be seen that the value of k is appropriate in the range of 0.8–1.2.

6. Conclusion

Data resampling method is one of the effective methods to deal with imbalanced data classification. Aiming at the problems of undersampling method and oversampling method, this paper proposes a hybrid sampling method, HSDP, based on data partition. This method uses the appropriate sampling methods for samples in different regions. And, it assigns reasonable weight to the boundary minority samples. Furthermore, it is able to oversample the selected samples inside the minority class area in the data space. The effectiveness of proposed method for the imbalanced data classification was confirmed by experiments, yet the values of the parameters used in the algorithm are selected through experiments many times. The future research direction is how to determine values of the parameters adaptively of HSDP for different datasets.

Data Availability

We use datasets from KEEL Repository and UCI Repository; our method and related parameters are provided in our paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Department of Education of Anhui Province, China (no. KJ2017A452).

References

[1] S. Das, S. Datta, and B. B. Chaudhuri, "Handling data irregularities in classification: foundations, trends, and future challenges," *Pattern Recognition*, vol. 81, pp. 674–693, 2018.

[2] P. Vuttipittayamongkol and E. Elyan, "Neighbourhood-based undersampling approach for handling imbalanced and overlapped data," *Information Sciences*, vol. 509, pp. 47–70, 2020.

[3] V. Garcia, J. S. Sanchez, A. I. Marqués, R. Florencia, and G. Rivera, "Understanding the apparent superiority of oversampling through an analysis of local information for class-imbalanced data," *Expert Systems with Applications*, vol. 158, 2020.

[4] H. Yin and K. Gai, "An empirical study on preprocessing high-dimensionalclass-imbal anced data for classification," in *Proceedings of the 2015 IEEE 12th International Conference on Embedded Software and Systems*, pp. 1314–1319, New York, NY, USA, August 2015.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[6] W. A. Rivera, "Noise reduction a priori synthetic over-sampling for class imbalanced data sets," *Information Sciences*, vol. 408, pp. 146–161, 2017.

[7] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE-IPF: addressing the noisy and borderline examples problem in

imbalanced classification by a re-sampling method with filtering," *Information Sciences*, vol. 291, pp. 184–203, 2015.

[8] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *Proceedings of the 2005 International Conference on Intelligent Computing. LNCS*, vol. 3644, pp. 878–887, Hefei, China, August 2005.

[9] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 1322–1328, IEEE, Hong Kong, China, June 2008.

[10] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, 2018.

[11] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[12] X. Gao, B. Ren, H. Zhang et al., "An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling," *Journal of Expert Systems with Applications*, vol. 160, no. 113660, pp. 1–18, 2020.

[13] Y. Xia, L. J. Li, Z. Xu, and B. Hae-Young, "Weighted oversampling of imbalanced data based on hierarchical clustering," *Computer Science*, vol. 46, no. 4, pp. 22–27, 2019.