Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 694809, 10 pages
http://dx.doi.org/10.1155/2013/694809

Research Article

A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE

Feng Hu and Hang Li

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Feng Hu; hufeng@cqupt.edu.cn

Received 24 April 2013; Accepted 27 August 2013

Academic Editor: Wei-Chiang Hong

Rough set theory is a powerful mathematical tool introduced by Pawlak to deal with imprecise, uncertain, and vague information. The Neighborhood-Based Rough Set Model expands the rough set theory; it could divide the dataset into three parts. And the boundary region indicates that the majority class samples and the minority class samples are overlapped. On the basis of what we know about the distribution of original dataset, we only oversample the minority class samples, which are overlapped with the majority class samples, in the boundary region. So, the NRSBoundary-SMOTE can expand the decision space for the minority class; meanwhile, it will shrink the decision space for the majority class. After conducting an experiment on four kinds of classifiers, NRSBoundary-SMOTE has higher accuracy than other methods when C4.5, CART, and KNN are used but it is worse than SMOTE on classifier SVM.

1. Introduction

The imbalanced dataset problem in classification domains occurs when the number of instances that represent one class is much larger than that of the other classes. The minority class is usually more interesting from the point of view of the learning task. There are many situations in which imbalance occurs between classes, such as satellite image classification [1], risk management [2], and medical diagnosis [3, 4]. When studying problems with imbalanced data, using the classifiers produced by standard machine learning algorithms without adjusting the output threshold may well be a critical mistake [5].

At present, the solutions for the problem of imbalanced dataset classification are developed at both the data and algorithmic levels [6]. At the data level, the objective is to rebalance the class distribution by resampling the data space, such as oversampling the minority class and undersampling the prevalent class. At the algorithm level, solutions try to adapt existing classifier learning algorithms to strengthen learning with regard to the minority class, such as cost-sensitive learning and ensemble learning. Resampling is convenient and effective; therefore, it is an often-used method in dealing with the class imbalance problem.

Previous research improved resampling methods in many aspects and proposed some effective resampling algorithms. SMOTE is an intelligent oversampling algorithm that was proposed by Chawla et al. [7]. Its main idea is to form new minority class samples by interpolating between several minority class samples that lie together. Thus, the overfitting problem is avoided and the decision space for the minority class spread further; meanwhile, it reduces the decision space for the majority class, so many researchers proposed different improved methods. Dong and Wang [8] proposed the Random-SMOTE, which is different from SMOTE, which obtained new minority class samples by interpolating among three minority class samples. Yang et al. [9] proposed ASMOTE algorithm which chose not only the minority class samples but also the majority class samples that are near to minority class sample, avoiding synthetic sample overlapping the majority class samples. Han et al. [10] proposed the Borderline-SMOTE. Their study considered the borderline minority samples which were most easily misclassified. They found out the borderline minority samples and thereby

generated synthetic samples from them. Compared with SMOTE, Borderline-SMOTE maintained the decision space for the majority class and enlarged the decision space for the minority class. However, when the number of minority class samples is particularly smaller than the one of majority class samples, most of the minority class samples are regarded as noise. Thus, few synthetic samples are generated which makes the method improve little accuracy. For these reasons, it is urgent to study more effective oversampling methods to generate high quality synthetic samples, particularly giving a better way to distinguish the borderline minority class samples.

It is important to find an effective mathematical theory to express and process the uncertainty of the minority class samples. Rough set theory is a powerful mathematical tool introduced by Pawlak [11–14] to deal with imprecise, uncertain, and vague information. It has been successfully applied to such fields as machine learning, data mining, intelligent data analysis, and control algorithm acquiring. Basically, the idea is to approximate a concept by three description sets, namely, the lower approximation, upper approximation, and boundary region. The rough set theory put the uncertain samples in the boundary region, and the boundary region can be calculated by upper approximation minus lower approximation, and they all can be calculated. Until now, there are many researchers who brought rough set theory to process imbalanced data. Liu et al. [15] proposed the weighted rough set model to process imbalanced data. It gave the minority class samples a higher weight to let the classifier focus on them. Ramentol et al. [16] introduced a hybrid preprocess approach by combining SMOTE with upper approximation. This method filtered the generated synthetic samples by comparing them with the majority class samples in upper approximation. Once the synthetic sample was similar to the majority class samples in upper approximation, they removed it to ensure that the synthetic samples approximate the minority class samples. Grzymala-Busse et al. [17] altered LEM2 algorithm by strengthening the rules to improve the classification of minority class samples.

The remainder of this paper is organized as follows. The basic concepts on neighborhood rough set models are shown in Section 2. By using the oversampling strategy of minority class samples in boundary region, the NRSBoundary-SMOTE algorithm is developed in Section 3. Section 4 presents the experimental evaluation on 15 imbalanced UCI datasets [18] by 10-fold cross validation, which shows the validity of the proposed method. The paper is concluded in Section 5.

2. Neighborhood-Based Rough Set Model

Neighoborhoods and neighborhood relations are a class of important concepts in topology. Lin [19] pointed out that neighborhood spaces are more general topological spaces than equivalence spaces and introduced neighborhood relation into rough set methodology. Hu et al. [20] discussed the properties of neighborhood approximation spaces and proposed the neighborhood-based rough set model. And then they used the model to build a uniform theoretic framework for neighborhood based classifiers.

For the convenience of description, some basic concepts of the neighborhood rough set model are introduced here at first.

Definition 1 (see [20]). Given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of x_i in the subspace B is defined as

$$\delta_B(x_i) = \left\{x_j \mid x_j \in U, \Delta_B\left(x_i, x_j\right) \le \delta\right\}, \tag{1}$$

where Δ is a metric function. $\forall x_1, x_2, x_3 \in U$, it satisfies

(1) $\Delta(x_1, x_2) \ge 0$;

(2) $\Delta(x_1, x_2) = 0$ if and only if $x_1 = x_2$;

(3) $\Delta(x_1, x_2) = \Delta(x_2, x_1)$;

(4) $\Delta(x_1, x_3) \le \Delta(x_1, x_2) + \Delta(x_2, x_3)$.

Consider that x_1 and x_2 are two objects. $A = \{a_1, a_2, \ldots, a_N\}$ is a sample-dimensional space, where $f(x, a_i)$ denotes the value of sample on the ith dimension a_i. Then, a general metric, named Minkowsky distance, is defined as

$$\Delta_p(x_1, x_2) = \left(\sum_{i=1}^{N} \left|f(x_1, a_i) - f(x_2, a_i)\right|^p\right)^{1/p}. \tag{2}$$

When $p = 2$, it is the Euclidean distance Δ_2.

But Euclidean distance can only be used to compute continuous features; the nominal features are invalid. Here, we compute them by using Value Difference Metric (VDM) proposed by Stanfill and Waltz [21] in 1986. The distance Δ between two corresponding feature values is defined as follows:

$$f(x_1, V_1) - f(x_2, V_2) = \sum_{i=1}^{n} \left|\frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2}\right|^k. \tag{3}$$

In the previous equation, V_1 and V_2 are the two corresponding feature values. C_1 is the total number of occurrences of feature value V_1, and C_{1i} is the number of occurrences of feature value V_1 for class i. A similar convention can also be applied to C_{2i} and C_2. k is a constant, which is usually set to 1.

Definition 2 (see [20]). Given a set of samples U, N is a neighborhood relation on U, and $\{\delta(x_i) \mid x_i \in U\}$ is the family of neighborhood granules. Then, we call $\langle U, N\rangle$ a neighborhood approximation space.

Definition 3 (see [20]). Given $\langle U, N\rangle$, for arbitrary $X \subseteq U$, two subsets of objects, called lower approximation and upper approximation of X in terms of relation N, are defined as

$$\begin{aligned} \underline{N}X &= \left\{x_i \mid \delta(x_i) \subseteq X,\ x_i \in U\right\}, \\ \overline{N}X &= \left\{x_i \mid \delta(x_i) \cap X \ne \phi,\ x_i \in U\right\}. \end{aligned} \tag{4}$$

The boundary region in the approximation space is formulated as

$$BN(X) = \overline{N}X - \underline{N}X. \tag{5}$$

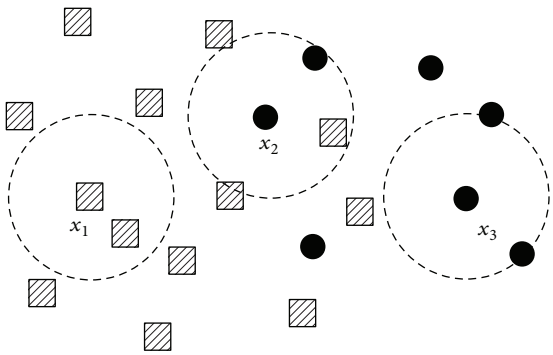

Figure 1: A sample with two classes.

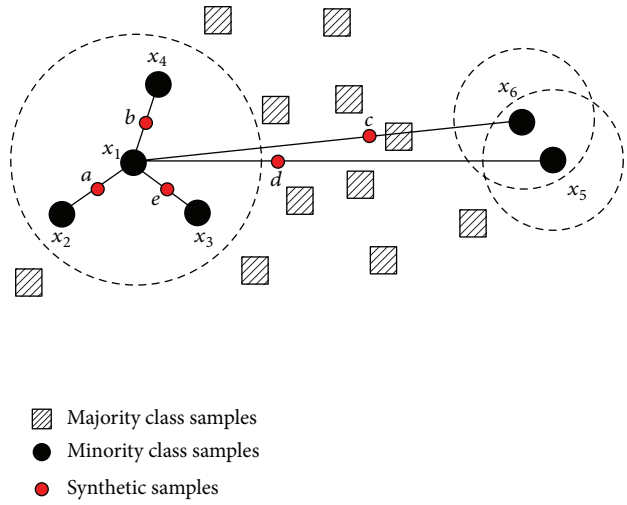

Figure 2: SMOTE leads to a poor prediction on majority class.

Definition 4 (see [20]). Given a neighborhood decision table, $NDT = \langle U, C \cup D, V, f\rangle$, $X_1, X_2, \ldots, X_n$ are the object subset with decisions 1 to N and $\delta_B(x_i)$ is the neighborhood information granules including x_i and generated by attributes $B \subseteq C$. Then the lower and upper approximations of the decision D with respect to attributes B are defined as

$$\underline{N_B}D = \bigcup_{i=1}^{N} \underline{N_B}X_i, \qquad \overline{N_B}D = \bigcup_{i=1}^{N} \overline{N_B}X_i, \tag{6}$$

where

$$\begin{aligned} \underline{N_B}X &= \{x_i \mid \delta_B(x_i) \subseteq X, x_i \in U\}, \\ \overline{N_B}X &= \{x_i \mid \delta_B(x_i) \cap X \neq \phi, x_i \in U\}. \end{aligned} \tag{7}$$

The decision boundary region of D with respect to attributes B is defined as

$$BN_B(X) = \overline{N_B}X - \underline{N_B}X. \tag{8}$$

Decision boundary region is the object subset whose neighborhoods come from more than one decision class. On the other hand, the lower approximation of the decision, also called the positive region of decision, denoted by $\mathrm{POS}_B(D)$, is the subset of objects whose neighborhoods decision only belongs to one of the decision classes.

To explain the samples in lower approximation of decision and boundary region, here we give a sample in Figure 1.

Example 5. Figure 1 gives a sample of binary classification in 2D space, where U_1 represent the majority class samples which are labeled by box and U_2 represent the minority class samples which are labeled by circle. Consider samples x_1, x_2, and x_3; we assign circle neighborhoods to these samples. We can find $\delta(x_1) \subseteq U_1$, $\delta(x_3) \subseteq U_2$, while $(\delta(x_2) \cap U_1 \neq \phi) \wedge (\delta(x_2) \cap U_2 \neq \phi)$. According to the aforementioned definitions, $x_1 \in \underline{N}U_1$, $x_3 \in \underline{N}U_2$, and $x_2 \in BN(U)$.

3. Neighborhood Rough Set Boundary SMOTE Algorithm

3.1. SMOTE Algorithm. SMOTE, proposed by Chawla et al., is a popular oversampling method. Its main idea is to construct new minority class samples by interpolating and selecting a near minority class neighbor randomly. The method can be described as follows. Firstly, for each minority class sample x, one gets its k-nearest neighbors from other minority class samples. Secondly, one chooses one minority class sample $\tilde{x}$ among the k neighbors. Finally, one generates the synthetic sample x_{new} by interpolating between x and $\tilde{x}$ as follows:

$$x_{\text{new}} = x + \text{rand}(0, 1) \times (\tilde{x} - x), \tag{9}$$

where rand $(0, 1)$ refers to a random number between 0 and 1.

In view of of geometry, SMOTE can be regarded as interpolating between two minority class samples. The decision space for the minority class is expanded that allows the classifier to have a higher prediction on unknown minority class samples.

The SMOTE algorithm is simple and effective while generating synthetic samples, and the overfitting problem is avoided. It expands the decision space for the minority class, but it may shrink the decision space for the majority class with high confidence in the meanwhile. Thereby, it will lead to poor prediction on the unknown majority class samples. Now, we will give an example to illustrate the drawback of SMOTE (see Figure 2).

In Figure 2, we apply SMOTE to generate synthetic samples for the minority class sample x_1. We generate randomly k ($k = 5$) nearest minority class samples of x_1 denoted by x_2, x_3, x_4, x_5, and x_6. According to Definition 3, x_1, x_5, and x_6 belong to lower approximation of the decision. Furthermore, x_5 and x_6 are farther from x_1 than x_2, x_3, x_4. If ones generates synthetic samples between x_1 and x_5 or between x_1 and x_6, the synthetic samples (such as points c and d) will overlap with (or very close to) the majority class samples. Thereby,

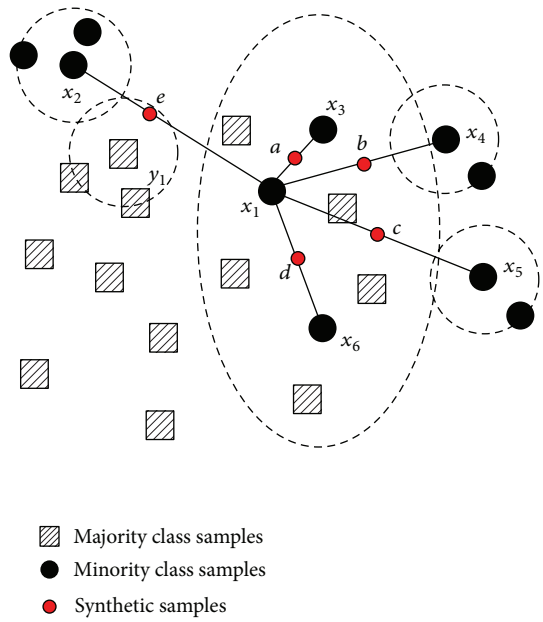

Figure 3: The schematic of NRSBoundary SMOTE algorithm.

misclassification will occur easily. Therefore, it is important to find the rational neighborhoods of minority class samples while oversampling.

3.2. Neighborhood Rough Set Boundary SMOTE Algorithm. In order to solve the aforementioned problem, we propose a new oversampling method, namely, Neighborhood Rough Set Boundary SMOTE (NRSBoundary SMOTE). The proposed method consists of three steps. First, we compute the minority class samples in boundary region and the majority class samples in lower approximation of decision. Second, for every minority class sample, we generate synthetic samples by calling SMOTE algorithm. Third, we select the rational synthetic samples without affecting the decision space of the majority class samples in lower approximation of decision.

In Figure 3, an example is given to explain NRSBoundary SMOTE further. The samples in the ellipse all belong to boundary region, while the ones outside belong to the lower approximation of decision. Now, we choose the minority samples in the boundary region for oversampling. We also find their k ($k = 5$) nearest neighbors of x_1, namely, $\{x_2, x_3, x_4, x_5, x_6\}$. Assume that the synthetic samples are a, b, c, d, and e, respectively. Obviously, $e \in \delta(y_1)$; that is, there is a risk that y_1 (a majority sample) can be classified into minority classification. Therefore, some effective methods should be adopted to avoid the risk.

It is an effective way that the synthetic sample cannot be in the neighborhood of any majority sample while oversampling. How to measure the neighborhood radius of sample is a primary issue. According to Definition 1, we should obtain the threshold δ firstly. Here we compute δ as follows [20]:

$$\delta = \min\left(\Delta\left(x_i, s\right)\right) + w \times \operatorname{range}\left(\Delta\left(x_i, s\right)\right), \quad 0 \le w \le 1, \tag{10}$$

where x_i ($i = 1, 2, \ldots, n$) is a training sample, $\min(\Delta(x_i, s))$ denotes the minimal value of distance between x_i and the remaining samples excluding s, and $\operatorname{range}(\Delta(x_i, s))$ denotes the value domain of $\Delta(x_i, s)$. In this case, δ is dynamically generated in terms with the whole training samples. In Section 4, we can afford a value domain of w, combined with the experimental analysis.

Here we give the NRSBoundary SMOTE (see Algorithm 1) as follows.

Time Complexity Analysis of Algorithm 1. Assume that $|TrainSet| = n$ and the number of features is m. The time complexity of step 1 is $O(1)$. The time complexity of step 2 is $O(n)$. The time complexity of step 3 is $O(m \times n^2)$. The time complexity of step 4 is $O(k \times m \times n^2)$. The time complexity of step 5 is $O(n)$. So the time complexity of Algorithm 1 is $O(k \times m \times n^2)$.

Space Complexity Analysis of Algorithm 1. The space complexity of Algorithm 1 is $O(m \times n)$.

4. Experimental Designing and Analysis

In this section, we first present the experimental setup, including the UCI datasets and the evaluation in imbalanced domains. Then we introduce the experimental analysis, which is divided into two parts: first we carry out an analysis of the parameters for our method, and then we develop the comparative analysis with other oversampling methods and some classifiers.

4.1. Datasets. In order to test the proposed algorithm, 15 UCI datasets are downloaded from the machine learning data repository, University of California at Irvine, with different imbalanced rates that from 0.20 to 0.804. There are four multiclass datasets and eleven two-class datasets. Multiclass datasets are modified to obtain two-class imbalance problems, by the union of one or more classes of the minority class and the union of one or more of the remaining classes which are labeled as the majority class. For the missing values, if they are continuous features, we fill them with average values; if they are nominal features, we fill them with values that appear most frequently. The datasets are outlined in Table 1 and sorted by imbalanced rates from low to high.

4.2. Experimental Evaluation in Imbalanced Domains. The traditional evaluation usually uses Confusion Matrix, showed in Table 2, where TP means the number of positive samples that are classified into positive, TN means the number of negative samples that are classified into negative, FN means the number of positive samples that are misclassified, and FP means the number of negative samples that are misclassified.

From Table 2, one could get some useful evaluation as follows.

Precision $= TP/(TP + FP)$; *Recall* $= TP/(TP + FN)$; *F-value* $= 2RP/(R + P)$, where R and P refer to *Recall* and *Precision*, respectively.

Input: the training sample set: $TrainSet$, the radius of neighborhood: w.
Output: new training sample set: $NewTrainSet$.
Step 1: **(Initialization)**
$SampleSet = \phi$; //$SampleSet$ is the generated synthetic sample set.
$BoundSet = \phi$; //$BoundSet$ is the minority class sample set in boundary region which needs over-sampling.
$PosSet = \phi$; //$PosSet$ is the majority class sample set in lower approximation of decision.
Step 2: (**Compute the majority class sample set and minority class sample set**)
According to the decision values 1 to N, divide $TrainSet$ into N subsets: $X_1, X_2, \ldots, X_N$;
Compute the minority class sample set $X_{\min} = \min(|X_1|, |X_2|, \ldots, |X_N|)$;
Compute the majority class sample set $X_{\max} = TrainSet - X_{\min}$;
Step 3: (**Compute boundary region and lower approximation of decision**)
FOR *each* x_i *in* $TrainSet$ DO
According to formulas (2) and (3), compute the distance $\Delta_2(x_j, x_i)$ between x_i and the other sample x_j in $TrainSet$ $(1 \le j \le |TrainSet|, j \ne i)$;
$d\text{Max}_i = \max\left\{\Delta_2\left(p_j, p_i\right) \mid j = 1, 2, \ldots, |TrainSet|\right\}$;
$d\text{Min}_i = \min\left\{\Delta_2\left(p_j, p_i\right) \mid j = 1, 2, \ldots, |TrainSet|\right\}$;
According to formula (10), compute the threshold δ_i of x_i;
Compute the neighborhood $\delta\left(x_i\right)$ of x_i, $\delta\left(x_i\right) = \{x_j \mid x_j \in U, \Delta\left(x_i, x_j\right) \le \delta_i\}$;
IF $(x_i \in X_{\min}) \wedge (\delta\left(x_i\right) \not\subset X_{\min})$ //minority class sample x_i which belongs to boundary region.
THEN $BoundSet = BoundSet \cup \{x_i\}$;
ELSE IF $(x_i \notin X_{\min}) \wedge (\delta\left(x_i\right) \subseteq X_{\max})$//majority class sample x_i which belongs to lower approximation of decision.
THEN $PosSet = PosSet \cup \{x_i\}$;
END IF
END FOR
Step 4: **(Generate synthetic samples from BoundSet)**
FOR *each* x_i *in BoundSet* DO
BOOL $IsConflict = false$;
Compute x_i's k $(k = 5)$ nearest neighborhoods with the same classification: $\{y_1, y_2, \ldots, y_k\}$;
$TempObjectSet = \{y_1, y_2, \ldots, y_k\}$;
WHILE $TempObjectSet \ne \phi$ DO
Choose one sample denoted by y $(y \in TempObjectSet)$ randomly;
$TempObjectSet = TempObjectSet - \{y\}$;
//Generate a synthetic sample.
$x_{\text{new}} = x_i + \text{rand}(0, 1) \times \left(y - x_i\right)$;
//Judge whether x_{new} affects the lower approximation of decision.
FOR *each* x_j *in PosSet* DO
IF $x_{\text{new}} \in \delta\left(x_j\right)$ THEN
$IsConflict = true$;
BREAK;
END IF
END FOR
//Add x_{new} to SampleSet
IF $IsConflict == false$ THEN
$SampleSet = SampleSet \cup \{x_{\text{new}}\}$;
END IF
END WHILE
END FOR
Step 5: (**Return**)
$NewTrainSet = SampleSet \cup TrainSet$;
RETURN $NewTrainSet$.

Algorithm 1: Neighborhood rough set boundary SMOTE algorithm for oversampling (NRSBoundary SMOTE).

Table 1: Data description.

	Datasets	Size	Attribute	Class label (minority : majority)	Class distribution
1	Austra	690	14C	1 : 0	307/383
2	Heart-s	270	6C 7N	Present : absent	120/150
3	Bupa	345	6C	1 : 2	145/200
4	Auto-mpg	398	5C 2N	Others : 1	149/249
5	Colic	368	7C 15N	No : yes	136/232
6	Ionosphere	351	34C	b : g	126/225
7	Machine	209	7C	Others : 2	74/135
8	Labor	57	8C 8N	Bad : good	20/37
9	Pima	768	8C	1 : 0	268/500
10	Vertebral column (VC)	310	7C	Normal : abnormal	100/210
11	German	1000	24C	2 : 1	300/700
12	Haberman	306	3C	2 : 1	81/225
13	Transfusion	748	4C	1 : 0	178/570
14	Contraceptive method choice (CMC)	1473	9C	2 : others	333/1140
15	Yeast	1484	8C	MIT : others	244/1240

C: continuous, N: nominal.

Table 2: Confusion matrix.

	Predict to positive	Predict to negative
Positive	TP	FN
Negative	FP	TN

There are three evaluations as the formulas called Precision, Recall, and F-value. Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while Recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. From the previous formulas, we can decrease FP to increase Precision and increase TP to increase Recall. But in fact they are conflicted. So we use the F-value to consider them comprehensively. Only when Precision and Recall are both higher, F-value will be higher.

Another appropriate metric that could be used to measure the performance of classification over imbalanced datasets is the Receiver Operating Characteristic (ROC) graphics [22]. In these graphics, the tradeoff between the benefits (TP) and costs (FP) can be visualized, and it acknowledges the fact that the capacity of any classifier cannot increase the number of true positives without also increasing the false positives. The area under the ROC curve (AUC) [23] corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single-number summary of the performance of learning algorithms.

4.3. The Experimental Results and Analysis. In this paper, we use Recall, F-value, and AUC to evaluate our algorithm. The oversampling method SMOTE [7] and the classifiers (such as C4.5, KNN, CART, and SVM [24]) are used in our experiment, whose source codes are afforded by Weka software [25]. We also use Java programming language to implement some other oversampling methods, such as ASMOTE [9], Borderline-SMOTE [10], and SMOTE-RSB* [16]. For the objective comparison, the minority class was over-sampled at 100% and the value of k is set to 5 like SMOTE. All results are computed by 10-fold cross validation.

(1) NRSBoundary SMOTE: Parameter Analysis. In the NRSBoundary SMOTE algorithm, it is important to set a proper value of w. Here, we conduct a series of experiments to find the optimal parameter w which is used to control the radius of the neighborhood. We try w from 0 to 0.2 with step 0.01 to compute the F-value by using the 10-fold cross validation. Figure 4 presents the F-value curves varying with w for some datasets: Pima, VC, Haberman, Transfusion, Colic, and CMC. From Figure 4, we can find that there is a similar trend in these curves: F-value increases at first and decreases after a threshold. So we recommend that w should take values in the range [0.01, 0.05].

(2) Comparative Analysis on C4.5. Tables 3, 4, and 5 give the comparative results of Recall, F-value, and AUC which are computed in different oversampling methods, respectively. Furthermore, none represents the original dataset without resampling.

From Tables 1, 2, and 3, we can figure out that the NRSBoundary-SMOTE has higher accuracy for most datasets. The average value of Recall increases to 0.7182 while the values of the others are between 0.6130 and 0.6886. The average value of F-value increases to 0.6978 while the values of the others are between 0.6505 and 0.6638. The average value of the AUC is up to 0.7882 while the values of the others are between 0.7615 and 0.7695. The NRSBoundary-SMOTE has higher accuracy on three evaluations than the others when the classifier is decision tree C4.5. It shows that our method is feasible by oversampling and strengthening the minority class samples in boundary region.

SMOTE over-samples for all the minority class samples. It can expand the decision space of minority class, but it will

Table 3: Recall.

	Dataset	None	SMOTE	ASMOTE	Borderline-SMOTE	SMOTE-RSB*	NRSBoundary-SMOTE
1	Austra	0.8013	0.8404	0.8339	0.8436	0.7948	**0.8502**
2	Heart-s	0.7333	0.7583	0.7417	0.7667	0.7333	**0.7833**
3	Bupa	0.5310	0.6483	0.6345	0.6207	0.5310	**0.6621**
4	Auto-mpg	0.8188	0.9060	0.8993	0.8859	0.8255	**0.9195**
5	Colic	0.6912	0.7500	**0.8088**	0.7500	0.6912	0.7574
6	Ionosphere	0.8254	0.8571	**0.8730**	0.8492	0.8254	**0.8730**
7	Machine	0.8514	0.8919	**0.9054**	0.8919	0.8514	0.8919
8	Labor	0.6500	0.5000	0.7500	0.7000	0.7000	**0.8000**
9	Pima	0.5970	0.7276	**0.7388**	0.6866	0.5970	0.7313
10	VC	0.6400	0.7700	0.6900	0.7100	0.6400	**0.7900**
11	German	0.4867	0.4233	0.4267	0.5133	0.4700	**0.5500**
12	Haberman	0.2963	0.5185	0.5432	0.5432	0.3457	**0.5926**
13	Transfusion	0.4326	0.4775	0.4438	0.4607	0.3989	**0.5000**
14	CMC	0.3153	0.4835	0.4535	0.4174	0.3153	**0.4985**
15	Yeast	0.4754	0.5779	**0.5861**	0.5123	0.4754	0.5738
	Average	0.6097	0.6754	0.6886	0.6768	0.6130	**0.7182**

w = 0.01 to 0.05.

Table 4: F-value.

	Dataset	None	SMOTE	ASMOTE	Borderline-SMOTE	SMOTE-RSB*	NRSBoundary-SMOTE
1	Austra	0.8132	**0.8459**	0.8352	0.8355	0.8093	0.8433
2	Heart-s	0.7364	0.7712	0.7265	0.7510	0.7364	**0.7932**
3	Bupa	0.5878	0.6045	0.5732	0.5590	0.5878	**0.6076**
4	Auto-mpg	0.8271	0.8491	0.8481	0.8381	0.8367	**0.8589**
5	Colic	0.7520	0.7445	0.7458	0.7473	0.7460	**0.7630**
6	Ionosphere	**0.8739**	0.8504	0.8730	0.8526	0.8739	0.8730
7	Machine	0.8690	0.8571	0.8816	**0.8980**	0.8690	**0.8980**
8	Labor	0.7027	0.5714	0.7500	0.6829	0.6829	**0.8205**
9	Pima	0.6142	0.6489	0.6397	0.6301	0.6142	**0.6555**
10	VC	0.6919	0.7163	0.6330	0.6961	0.6919	**0.7248**
11	German	0.5280	0.4544	0.4444	0.5159	0.5127	**0.5428**
12	Haberman	0.3582	0.4828	0.4972	0.5116	0.3944	**0.5393**
13	Transfusion	0.4813	0.4749	0.4745	0.4852	0.4551	**0.5000**
14	CMC	0.3896	0.4613	0.4333	0.4257	0.3896	**0.4723**
15	Yeast	0.5577	0.5732	0.5813	0.5274	0.5577	**0.5749**
	Average	0.6522	0.6604	0.6625	0.6638	0.6505	**0.6978**

w = 0.01 to 0.05.

decrease the decision space of majority class. Although it can improve Recall of minority class, many majority class samples will be misclassified as minority class, which thereby results in the decreasing of Precision. Thus, the results of F-value have not been improved too much.

ASMOTE, similar to SMOTE, considers the near neighborhood of majority class. It can reduce the confliction between synthetic samples and majority class samples and also expand the coverage space of minority class samples. However, some of the synthetic samples are similar to majority class samples, so the decision space of majority class samples decreases as well.

Both Borderline-SMOTE and SMOTE-RSB* sift the synthetic samples more strict than SMOTE, because few synthetic samples are generated when datasets are highly imbalanced. Thus, compared with SMOTE, its improvement is not obvious.

NRSBoundary-SMOTE uses the neighborhood rough set model, which emphasizes oversampling the minority class samples in boundary region, and thereby expands the coverage space of minority class samples in boundary region. Furthermore, it can improve the confidence degree of decision rules by minority class samples in boundary region (or uncertain area). What is more, it has little influence on the

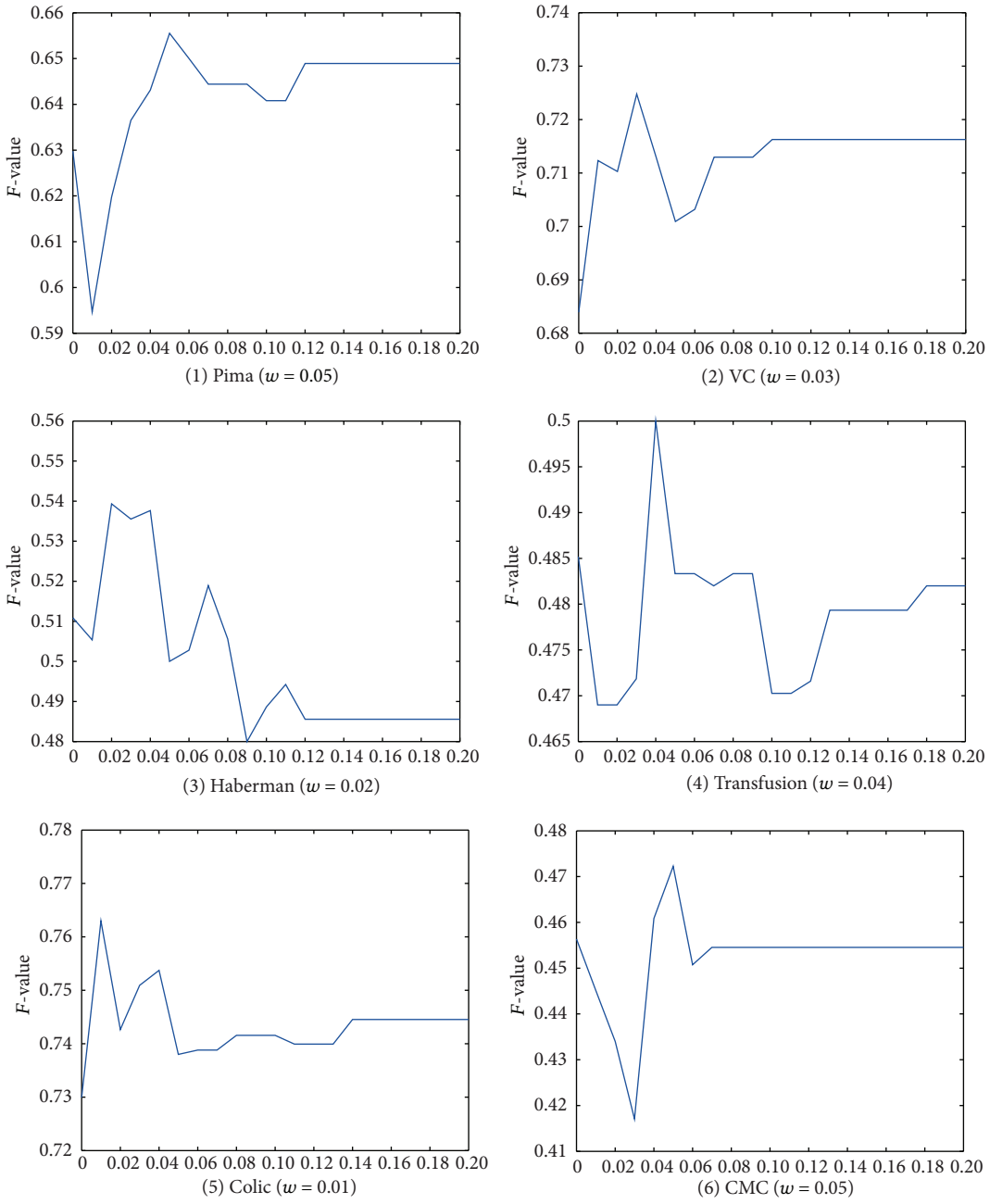

Figure 4: F-value curves varying with w.

majority class samples in lower approximation of decision; in other words, it has little influence on changing the decision space of majority class. Thus, the results of F-value have been improved.

(3) Comparative Analysis on KNN, CART, and SVM. In addition, in order to test the validity of the proposed method on different classifiers, KNN ($k = 3$), CART, and SVM are adopted on 15 UCI datasets. The experimental results of F-value are shown in Figure 5, where the F-value is the average value of F-values on 15 UCI datasets.

From Figure 5, we find out that NRSBoundary-SMOTE has higher accuracy than other methods when C4.5, CART, and KNN are used. On the contrary, it is worse than SMOTE on SVM. In the course of classification of C4.5, CART, and KNN, they are all based on measuring the distance between the unknown samples and one of the train samples or rules; at the same time, the process of computing neighborhoods in NRSBoundary-SMOTE is similar to these classifiers. Therefore, NRSBoundary-SMOTE can perform better. But SVM works by constructing a separating hyperplane with the maximal margin, which has not been taken into consideration by NRSBoundary-SMOTE algorithm; it has no better effect on SVM.

In NRSBoundary-SMOTE algorithm, one can expand the decision space of the minority samples in boundary region

Table 5: AUC.

	Dataset	None	SMOTE	ASMOTE	Borderline-SMOTE	SMOTE-RSB*	NRSBoundary-SMOTE
1	Austra	0.8483	**0.8739**	0.8433	0.8635	0.8458	0.8657
2	Heart-s	0.7443	0.7947	0.7223	0.7458	0.7448	**0.8146**
3	Bupa	0.6650	0.6468	0.6132	0.6355	**0.6652**	0.6401
4	Auto-mpg	0.9282	0.8973	0.8874	0.9015	**0.9303**	0.9197
5	Colic	0.7873	0.7740	**0.8170**	0.7971	0.7855	0.8102
6	Ionosphere	0.8923	0.8879	**0.9130**	0.8778	0.8930	0.9078
7	Machine	0.9359	0.9199	0.9268	**0.9538**	0.9359	0.9430
8	Labor	0.7588	0.7500	0.8047	0.7655	0.7655	**0.8243**
9	Pima	**0.7514**	0.7417	0.7321	0.7158	0.7513	0.7419
10	VC	**0.8380**	0.8107	0.7688	0.8044	0.8380	0.8277
11	German	**0.6884**	0.6333	0.6115	0.6853	0.6620	0.6791
12	Haberman	0.6087	0.6255	0.6570	0.6536	0.6174	**0.6636**
13	Transfusion	0.7001	0.6813	0.6836	0.7075	0.7007	**0.7048**
14	CMC	0.6373	0.6866	0.6723	0.6710	0.6375	**0.6980**
15	Yeast	0.7500	0.7801	0.7698	0.7649	0.7500	**0.7831**
	Average	0.7689	0.7669	0.7615	0.7695	0.7682	**0.7882**

w = 0.01 to 0.05.

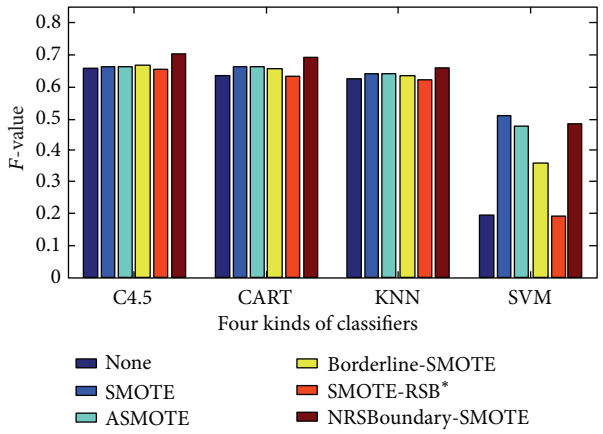

Figure 5: Comparison of different kinds of classifiers.

by oversampling them. The boundary region of the training sample set is computed based on neighborhood-based rough set model. Furthermore, the distance between two samples is regarded as an important factor for classification, which is suitable for C4.5, CART, and KNN. However, SVM works by constructing a separating hyperplane. The distance from one sample to the hyperplane is the main factor for classification, not the distance between two samples. That is, the computation of boundary region of the sample set has nothing to do with the hyperplane of SVM. Only when two samples are in the same side of the hyperplane, they will be classified into the same category. But, two closed samples with different classifications will be regarded as the same classification in neighborhood-based rough set model. For example, for a minority sample x_1 near to the hyperplane of SVM, let x_2 be the synthetic sample around x_1. Since x_2 is near to x_1, x_2 should be denoted as the minority classification by our algorithm. In fact, x_2 may be a majority sample which is in the other side of the hyperplane. Obviously, it is wrong due to denoting error classification for the synthetic sample. Therefore, the proposed algorithm is not suitable for SVM, because the hyperplane is not considered in oversampling.

5. Conclusions

In this paper, we present a new oversampling method, called NRSBoundary-SMOTE, to process imbalanced dataset. In this method, only the minority class samples in the boundary region should be over-sampled. It can expand the decision space of minority class samples, while it has little influence on the decision space of majority class samples. The experimental evaluation on 15 UCI datasets with different imbalanced rates shows that the proposed method has better performance than SMOTE, when combining with C4.5, CART, and KNN. But SMOTE is better than NRSBoundary-SMOTE when using SVM. The proposed method is an effective method for oversampling. However, it will spend more time to filter the synthetic samples. Thus, it will be difficult to process large dataset, due to the long running time. Studying and developing new fast algorithms for oversampling will be our future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 61073146, 61272060, 61203308, 61309014, and 61379114, Natural Science Foundation Project of CQ CSTC under Grant nos. cstc2012jjA40032, cstc2012jjA40047, cstc2013jcyjA40063, and cstc2013jcyjA40009, and Doctor Foundation of Chongqing University of Posts and Telecommunications under Grant no. A2012-08.

References

[1] S. Suresh, N. Sundararajan, and P. Saratchandran, "Risk-sensitive loss functions for sparse multi-category classification problems," *Information Sciences*, vol. 178, no. 12, pp. 2621–2638, 2008.

[2] G. Wang, "Asymmetric random subspace method for imbalanced credit risk evaluation," in *Software Engineering and Knowledge Engineering: Theory and Practice*, vol. 114 of *Advances in Intelligent and Soft Computing*, pp. 1047–1053, 2012.

[3] J. M. Malof, M. A. Mazurowski, and G. D. Tourassi, "The effect of class imbalance on case selection for case-based classifiers: an empirical study in the context of medical decision support," *Neural Networks*, vol. 25, pp. 141–145, 2012.

[4] M. H. Wei, C. H. Cheng, and C. S. Huang, "Discovering medical quality of total hiparthroplasty by rough set classier with imbalanced class," *Quality & Quantity*, vol. 47, no. 3, pp. 1761–1779, 2013.

[5] F. Provost, "Machine learning from imbalanced data sets," in *Proceedings of the AAAI-2000 Workshop Learning from Imbalanced Data Sets*, pp. 1–3, 2000.

[6] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[7] 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.

[8] Y. Dong and X. Wang, "A new over-sampling approach: random-SMOTE for learning from imbalanced data sets," in *Proceedings of the 5th International Conference on Knowledge Science, Engineering and Management (KSEM '11)*, pp. 343–352, Springer, Berlin, Germany, 2011.

[9] Z. M. Yang, L. Y. Qiao, and X. Y. Peng, "Research on datamining method for imbalanced dataset based on improved SMOTE," *Acta Electronica Sinica*, vol. 35, no. B12, pp. 22–26, 2007 (Chinese).

[10] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," *Advances in Intelligent Computing*, vol. 2, no. 5, pp. 878–887, 2005.

[11] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[12] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, no. 1, pp. 3–27, 2007.

[13] Z. Pawlak and A. Skowron, "Rough sets: some extensions," *Information Sciences*, vol. 177, no. 1, pp. 28–40, 2007.

[14] Z. Pawlak and A. Skowron, "Rough sets and Boolean reasoning," *Information Sciences*, vol. 177, no. 1, pp. 41–73, 2007.

[15] J. Liu, Q. Hu, and D. Yu, "A weighted rough set based method developed for class imbalance learning," *Information Sciences*, vol. 178, no. 4, pp. 1235–1256, 2008.

[16] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RS B*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowledge and Information Systems*, vol. 33, no. 2, pp. 245–265, 2012.

[17] J. W. Grzymala-Busse, J. Stefanowski, and S. Wilk, "A comparison of two approaches to data mining from imbalanced data," *Journal of Intelligent Manufacturing*, vol. 16, no. 6, pp. 565–573, 2005.

[18] C. Blake, E. Keogh, and C. J. Merz, *UCI Repository of Machine Learning Databases*, Department of Information and Computer Science, University of California, Irvine, Calif, USA, 1998, http://archive.ics.uci.edu/ml/.

[19] T. Y. Lin, "Granular computing on binary relations I: data mining and neighborhood systems," *Rough Sets in Knowledge Discovery*, vol. 1, pp. 286–318, 1998.

[20] Q. Hu, D. Yu, and Z. Xie, "Neighborhood classifiers," *Expert Systems with Applications*, vol. 34, no. 2, pp. 866–876, 2008.

[21] C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Communications of the ACM*, vol. 29, no. 12, pp. 1213–1228, 1986.

[22] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[23] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[24] X. Wu, V. Kumar, R. S. Quinlan et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[25] "Wikipedia Weka (machine learning)," 2010, http://en.wikipedia.org/wiki/Weka.