Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 876875, 13 pages
http://dx.doi.org/10.1155/2014/876875

Research Article

A Novel Algorithm for Imbalance Data Classification Based on Neighborhood Hypergraph

Feng Hu, Xiao Liu, Jin Dai, and Hong Yu

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 25 March 2014; Revised 29 June 2014; Accepted 21 July 2014; Published 11 August 2014

Academic Editor: Maria Jose del Jesus

The classification problem for imbalance data is paid more attention to. So far, many significant methods are proposed and applied to many fields. But more efficient methods are needed still. Hypergraph may not be powerful enough to deal with the data in boundary region, although it is an efficient tool to knowledge discovery. In this paper, the neighborhood hypergraph is presented, combining rough set theory and hypergraph. After that, a novel classification algorithm for imbalance data based on neighborhood hypergraph is developed, which is composed of three steps: initialization of hyperedge, classification of training data set, and substitution of hyperedge. After conducting an experiment of 10-fold cross validation on 18 data sets, the proposed algorithm has higher average accuracy than others.

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, ensemble learning, and hypernetwork [7].

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. [8]. 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 [9] 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. [10] 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. [11] proposed the Borderline-SMOTE. Hu and Li [12] proposed NRSBoundary-SMOTE algorithm which can expand the decision space for the minority class; meanwhile, it will shrink the decision space for the majority class.

While in recent years, with the rapid developing of ensemble methods for classification, they have been applied to imbalanced data classification, ensemble learning is a machine learning paradigm where multiple learners (called base learners) are trained to solve the same problem [13]. Due to the outstanding performance of ensemble methods, they

are applied to imbalanced dataset by combining with other techniques. Chawla et al. have developed SMOTEBoost algorithm by integrating Adaboost (the most famous boosting algorithm) and synthetic minority oversampling technique (SMOTE) [14]. Similarly to SMOTEBoost, RUSBoost also introduces data sampling into the Adaboost algorithm, while it applies random undersampling to the majority class; but SMOTEBoost creates synthetic new minority class instances by operating in the feature space [15]. Błaszczyński et al. integrate a selective data preprocessing method SPIDER with Ivotes ensemble algorithm developing the framework called IIvotes [16]. Besides, cost-sensitive learning becomes an effective tool to solve class imbalanced problem, which involves two types: binary classification problem and multiclassification problem. It can be implemented by two ways that are rescaling and reweighted, respectively. Both of them aim at making the trained classification algorithms cost sensitive. Rescaling changes the distribution of samples in training data. It has been used in cost-based sampling [17], REBALANCE [18], Rescalenew [19], and so on. Differing from rescaling, reweighted adjusts the class probability distribution in classifier based on costs. It has been used in MetaCost [20] proposed by Domings and AdaCost [21], which is improved by Fan et al. according to AdaBoost.

In the 1970s, Rumelhart and Norman proposed three types of human learning: accretion, tuning, and restructuring [22]. Based on their study, professor Zhang proposed three basic principles of cognitive learning [23]: (1) continuity, (2) glocality, and (3) compositionality. He used a hypergraph as presentation form and proposed a hypernetwork model, which can be used for cognitive learning and memory. Hypernetwork is a probabilistic graph with numbers of hyperedges. A hyperedge can be regarded as a component, a subject, or even a circuit. From the perspective of data, a hyperedge is the combination of sample attributes and class. So far, hypernetwork can just deal with discrete data. Dataset must be discrete before using for building a hypernetwork classifier. A hypernetwork model includes three steps: (1) initializing of a hypernetwork model according to training dataset, (2) evolutionary learning of a hypernetwork, and (3) classification of test dataset using the evolutionary hypernetwork. In step (1), a sample is used to generate many hyperedges through inheriting some attributes of the sample and its class. In step (2), operations of match, selection, and replacement are repeatedly executed for hyperedges. It is started from a randomly initialized hypernetwork; in each iteration, fitness of a hyperedge is calculated for evaluating and ordering. Hyperedges which own low fitness are replaced with new generated hyperedges. In this way, hyperedges with high class discernibility in pattern space can be found out by the hypernetwork [7]. After the above steps, a hypernetwork model is built and classifies the test data through joint probability.

Although hypernetwork has been widely used in solving various machine learning problems, it usually produces poor overall classification performance when dealing with class imbalance problems. Like most of the traditional classification algorithms, hypernetwork assumes that the class distribution of datasets is balanced. The goal of the hypernetwork learning is to extract hyperedges (or decision rules) that can cover as many samples as possible. Hyperedges are critical for differentiating class membership which are copied and added while hyperedges with poor differential ability are discarded. However, within the context of class imbalance, many samples in minority class are usually viewed as noises. Therefore, the number of hyperedges corresponding to the majority significantly surpasses that of hyperedges corresponding to the minority. As a result, most of the minority samples are misclassified in a traditional hypernetwork. Thus, this paper attempts to combine hypernetwork with rough set to address the problem.

Rough set theory is a powerful mathematical tool introduced by Pawlak [24–27] 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 puts 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 [28, 29].

The remainder of this paper is organized as follows. The basic concepts on neighborhood rough set models are shown in Section 2. The neighborhood hypergraph algorithm is developed in Section 3. Section 4 presents the experimental evaluation on 18 imbalanced UCI datasets [30] by 10-fold cross validation, which shows the validity of the proposed method. The paper is concluded in Section 5.

2. Hypergraph and Neighborhood Hypergraph Model

2.1. The Definition of Hypergraph. In 1970, Berge [31] used hypergraph to define hypernetwork. And it was the first time to establish undirected hypergraph theory systematically and it was applied on the operations research by matroid.

Definition 1 (see [31, 32]). Given $V = \{v_1, v_2, \ldots, v_n\}$ is finite set, if

(a) $E_i \neq \phi$ $(i = 1, 2, \ldots, m)$,

(b) $\bigcup_{i-1}^{m} E_i = V$,

then the binary relation $H = \{E, V\}$ is defined as a hypergraph. The elements $v_1, v_2, \ldots, v_n$ of V are defined as vertices of the hypergraph; $E = \{E_1, E_2, \ldots, E_m\}$ is defined as the edge set of hypergraph. $E_i = \{E_{i_1}, E_{i_2}, \ldots, E_{i_j}\}$ is defined as hyperedge (see Figure 1).

2.2. The Neighborhood Hypergraph. Neighborhoods and neighborhood relations are a class of important concepts in topology. Lin [33] pointed out that neighborhood spaces are more general topological spaces than equivalence spaces and introduced neighborhood relation into rough set methodology. Hu et al. [34] discussed the properties of neighborhood

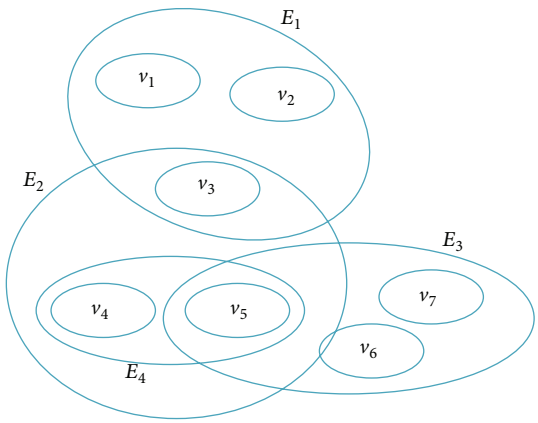

Figure 1: An example of hypergraph.

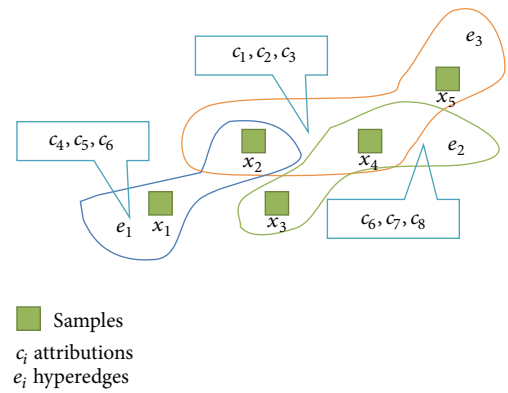

Figure 2: An example of neighborhood hypergraph.

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 2 (see [34]). Given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\partial_B(x_i)$ of x_i in the subspace B is defined as

$$\partial_B(x_i) = \left\{x_j \mid x_j \in U, \Delta_B\left(x_i, x_j \leq \delta\right)\right\}, \quad (1)$$

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

(1) $\Delta(x_1, x_2) \geq 0$,

(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) $\Delta(x_1, x_3) \leq \Delta(x_1, x_2) + \Delta(x_2, x_3)$.

Consider that x_1 and x_2 are two objects. $A = \{a_1, a_2, \ldots, a_m\}$ 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

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

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 [35] in 1986. The distance between two corresponding feature values is defined as follows:

$$f(x_1, V_1) - f(x_2, V_2) = \sum_{i=1}^{n} \left|\frac{C_{1i}}{C_1} - \frac{C_{2i}}{C2}\right|^k. \quad (3)$$

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 be also applied to C_{2i} and $C2$. k is continuous, which is usually set to 1.

Definition 3. Given $G = \langle X, E\rangle$ is a neighborhood hypergraph, then $X = \{x_1, x_2, \ldots, x_n\}$ is the vertex set of G, indicating that it has n vertices, where x_i denotes a sample. $E = \{e_1, e_2, \ldots, e_n\}$ is hyperedge set, and each e_i in E is a hyperedge which connects k vertices $(x_{i1}, x_{i2}, \ldots x_{ik})$. $C = \{c_1, c_2, \ldots, c_m\}$ is the attribute set, and D is the decision set.

Vertices of hypergraph represent the attribution of samples in some literatures like literature [36] and so on. However, in this paper, vertices of hypergraph are denoted as samples and different samples on one hyperedge have the same attributes set. An example of neighborhood hypergraph is as in Figure 2.

Definition 4. Given $x = \{c_1(x), c_2(x), c_3(x), \ldots, c_n(x), D(x), \delta\}$ is a sample, where $c_1(x), c_2(x), c_3(x), \ldots, c_n(x)$ denote the values of x at the attributes set C, $D(x)$ denotes the decisions of x, and δ is the radius of a neighborhood.

Definition 5. Given $G = \langle X, E\rangle$ and the attribute set C, the hyperedges which are included by sample x, consisting of the set of hyperedges, are defined as

$$\delta_C(x) = \{e \mid (e \in E) \wedge \Delta(e, x) \leq \delta\}. \quad (4)$$

Definition 6. Given $G = \langle X, E\rangle$ and attributes set B $(B \subseteq C)$, for arbitrary $e \in E$, the sample set $in_B(e)$ which related to e is defined as $in_B(e) = \{x \mid e \in \delta(x), x \in X\}$. Given arbitrary $Y \subseteq E$ and attributes set $B(B \subseteq C)$, the sample set $in_B(Y)$ related to Y is defined as $in_B(Y) = \{in_B(e) \mid e \in Y\}$.

Definition 7. Given $G = \langle X, E\rangle$, for arbitrary $e \in E$, one knows $D(e) \in \{P, N\}$, where P denotes minority decision and N denotes majority decision. Then sets of decision P and decision N in hyperedge set E are, respectively, defined as

$$\begin{aligned} nE(P) &= \{e \mid D(e) = P, e \in E\}, \\ nE(N) &= \{e \mid D(e) = N, e \in E\}; \end{aligned} \quad (5)$$

thus, the degree of imbalance for E is defined as

$$inbanE = \frac{|nE(N)|}{|nE(P)|}. \quad (6)$$

Definition 8. Given $G = \langle X, E\rangle$, for arbitrary $e \in E$, assume $D(e) \in \{P, N\}$, where P denotes minority decision and N denotes majority decision. $in_B(e)$ is sample set related to hyperedge e on attributes set $B \subseteq C$. According to decisions D, $in_B(e)$ is divided into p equivalence classes: $X_1, X_2, \ldots, X_p$; when $in_B(e) \neq \phi$, the confidence degree of e is defined as follows.

(1) If $D(e) = P$, then

$$\begin{aligned}&\mathrm{Conf}_B(e)\\ &= \left|\{x \mid x \in in_B(e), D(x) = D(P)\}\right| \times inbanE\\ &\quad \times \left(\left|\{x \mid x \in in_B(e), D(x) = D(N)\}\right|\right.\\ &\qquad \left. + \left|\{x \mid x \in in_B(e), D(x) = D(P)\}\right| \times inbanE\right)^{-1}.\end{aligned} \quad (7)$$

(2) If $D(e) = N$, then

$$\begin{aligned}&\mathrm{Conf}_B(e)\\ &= \left|\{x \mid x \in in_B(e), D(x) = D(N)\}\right|\\ &\quad \times \left(\left|\{x \mid x \in in_B(e), D(x) = D(N)\}\right|\right.\\ &\qquad \left. + \left|\{x \mid x \in in_B(e), D(x) = D(P)\}\right| \times inbanE\right)^{-1}.\end{aligned} \quad (8)$$

Definition 9. Given $G = \langle X, E\rangle$, C is the attitudes set of samples, and D is the samples decision. For arbitrary hyperedge set E' ($E' \subseteq E$), according to decisions D, the hyperedge set E' is divided into p equivalence classes: $E_1, E_2, \ldots, E_p$. For arbitrary $B \subseteq C$, the upper approximation, lower approximation, boundary region, and negative domains of decision D related to set of attributes B are, respectively, defined as

$$\begin{aligned}\overline{N}_B(D) &= \bigcup_{i=1}^{p}\left\{e \mid in_B(e) \cap in_B(E_i) \neq \phi \vee (e \in E_i), e \in E'\right\},\\ \underline{N}_B(D) &= \bigcup_{i=1}^{p}\{e \mid \mathrm{Conf}_B(e) = 1, e \in E_i\},\\ BN_B(D) &= \overline{N}_B(D) - \underline{N}_B(D),\\ \mathrm{Neg}_B(D) &= E - \overline{N}_B(D).\end{aligned} \quad (9)$$

The lower approximation of decision D that related to the set of attributes C is also called positive domain, denoted by $\mathrm{POS}_B(D)$. The size of positive domain reflects the separable degree of classification problem in a given attribute space; the bigger the positive region, the smaller the border.

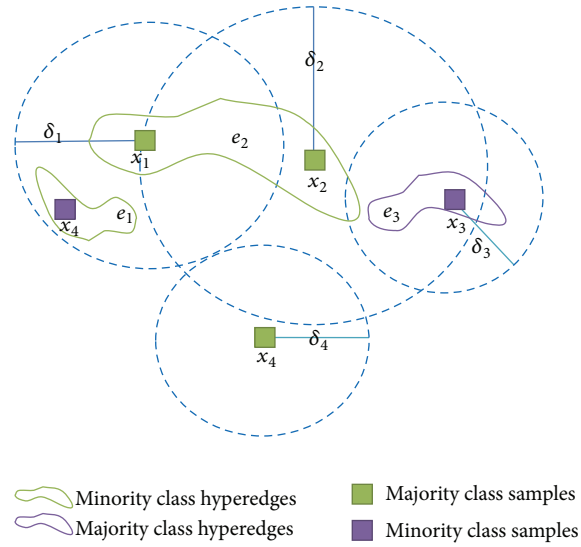

FIGURE 3: An example of upper, lower approximation, and boundary region.

To explain how to divide the upper approximation, lower approximation, and boundary region, here we give an example (Example 1) in Figure 3.

In Figure 3, the hyperedge e_2 is simultaneous in the neighborhood of samples x_1 and x_2; in other words, it links x_1 and x_2. From the graph, we can know easily that whether a hyperedge is in the neighborhood is up to the fact that whether the symbol e_i of the hyperedge is inside of the neighborhood of the sample.

First, one calculates the sample set $in_B(e_i)$ of each hyperedge: $in_B(e_1) = \{x_1\}$, $in_B(e_2) = \{x_1, x_2\}$, $in_B(e_3) = \{x_2, x_3\}$ according to Formula (4) and Definition 6. Second, one calculates the confidence degrees of each hyperedge, according to Formula (8): $\mathrm{Conf}_B(e_1) = 0$, $\mathrm{Conf}_B(e_2) = 1$, $\mathrm{Conf}_B(e_3) = 0.5$. Third, according to Formula (9), one gets the final result on Figure 3: the upper approximation $\overline{N}_B(D) = \{e_1, e_2, e_3\}$, the lower approximation $\underline{N}_B(D) = \{e_2\}$, and the boundary region $BN_B(D) = \{e_1, e_3\}$.

3. Neighborhood Hypergraph Classification Algorithm

Traditional hypernetwork model has limit on some aspects as follows: (1) discretized datasets. (2) There is no special processing to the samples in boundary region. However, some advantages will appear when rough set theory is combined with hypernetwork: (1) hypernetwork can directly deal with numeric data, which avoids information loss of data. (2) In the process of hyperedge learning, hyperedge set is divided into three parts that are upper approximation, lower approximation, and boundary region. In addition, hyperedges in

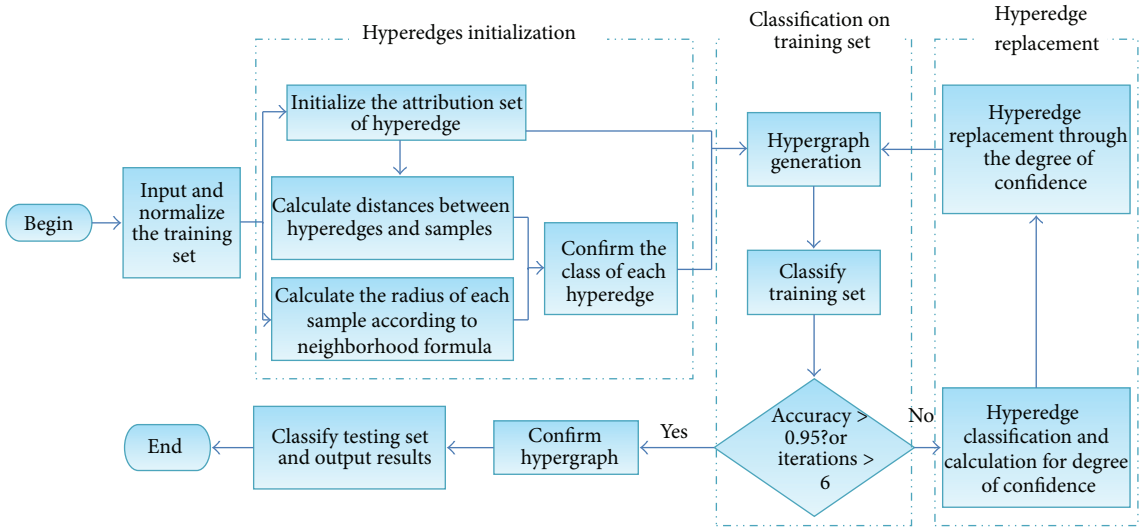

Figure 4: The flow chart of algorithm.

boundary region will be processed specially, which will result in the improvement of classification accuracy.

The proposed algorithm aims at tackling imbalanced data classification problem including two aspects as follows.

(1) Improve the degree of imbalance of hyperedge set. The class of traditional hyperedge is inherited from samples directly, which is helpless to improve the degree of imbalance of hyperedge set. In the paper, when initializing hyperedges, classes of fractional hyperedges are determined according to the classes of samples, which reduces the degree of hyperedge set to some extent.

(2) Set classification condition. The classification process of traditional Hypernetwork does not take the degree of hyperedge set into consideration, resulting in a low accuracy of minority class. However, one sets a threshold, which equals the square of the degree of imbalance, as a classification condition. This method makes the classifier pay more attention to minority class and thus can deal with class imbalance problem appropriately.

The flow chart of the algorithm is shown in Figure 4. Then, one analyzes each part of the flow chart of the remaining section specifically as follows.

3.1. Hyperedge Initialization. Hyperedges are generated based on the samples, which reserve the real distribution of the sample set and thereby provide a foundation for hyperedge selection. Meanwhile, one can change some attribute values while generating hyperedge. Thus, more decision rules are generated for sample classification, which can improve the accuracy of sample classification to some extent.

In this paper, attribution set of hyperedges is C (namely, the universal set of attributions for samples); that is to say, a hyperedge is exactly a sample and denoted by a small dot in the figure. In hyperedge initialization, we can assign a value on each attribute and determine the classification of each hyperedge.

x	1	2	3	4	5	6	7	8	9	10
e	1	2	24	4	5	11	7	8	9	55

Figure 5: Attribution inheritance.

In order to process imbalanced dataset, two classes will be considered in the following definitions.

Definition 10. Given $G = \langle X, E\rangle$, for arbitrary $x \in X$, assume $D(x) \in \{P, N\}$, where P denotes minority decision and N denotes majority decision. Then sets of decision P and N decision in hyperedge set are, respectively, defined as

$$nX(P) = \{x \mid D(x) = P, x \in E\},$$
$$nX(N) = \{x \mid D(x) = N, x \in E\}. \quad (10)$$

So the degree of imbalance for E is defined as

$$inbanX = \frac{|nX(N)|}{|nX(P)|}. \quad (11)$$

In this paper, the process of hyperedge generation consisted of two stages: attribution inheritance and class confirmation.

(1) Attribution inheritance: hyperedge and sample have the same attribution set, and the attribute values of the hyperedge are assigned partly based on the sample. One selects 7/10 of all attributes of one hyperedge which are selected randomly and they inherit the corresponding attribution value of the sample. The remaining attribute values are generated randomly in the range of the corresponding attribute values of the sample. In Figure 5, x is a sample and e is a hyperedge.

Definition 11. Given $G = \langle X, E\rangle$, for arbitrary $e \in E$, the majority sample set and the minority sample set related to hyperedge e in the sample set are defined as

$$inN(e) = \{x \mid e \in \delta(x), D(x) = N, x \in X\},$$
$$inP(e) = \{x \mid e \in \delta(x), D(x) = P, x \in X\}. \quad (12)$$

(2) Class confirmation: the class of a hyperedge is confirmed by the whole dataset. There are two cases as follows.

(a) According to formula (11), if $(|inN_B(e)|/|inP_B(e)|) \geq inbanX$, then $D(e) = N$;

(b) otherwise, generate a random number r in $[0, 1]$. If $r \geq (|X_N|/(|X_P| + |X_N|))$ (X_N, X_P, respectively, represent majority sample set and minority sample set), then $D(e) = N$; else, $D(e) = P$.

3.2. Classification on Training Set. One uses the generated hyperedge set to classify the training set. Through analyzing the classification result, one can know the classification accuracy of hyperedge set and determine whether to replace the hyperedge for the hyperedge set or not. By repeating the process of training sample classification and hyperedge replacement, we can make the distribution of hyperedge set approach the real distribution of training sample set gradually.

Definition 12. Given $G = \langle X, E\rangle$, the majority hyperedge set and minority hyperedge set in neighborhood of sample x are defined as

$$ne(x, N) = \{e \mid D(e) = N, e \in \delta(x)\},$$
$$ne(x, P) = \{e \mid D(e) = P, e \in \delta(x)\}. \quad (13)$$

One should consider the factor when we use neighborhood hypergraph to classify the samples: (1) the amounts of majority hyperedge and minority hyperedge in the neighborhood of a sample; (2) the degree of the imbalance of hyperedge set. Combining with the above, one presents the classification method.

Given sample set X, for arbitrary $x \in X$ and attribution set $B \subseteq C$,

(1) if $ne(x, N)/ne(x, P) \geq inbanE^2$, then $D(x) = N$;

(2) if $ne(x, N)/ne(x, P) < inbanE^2$, then $D(x) = P$.

One uses the classification rules above to classify the training set. If the accuracy is higher than 0.95, one can output the hyperedge set. Otherwise, the hyperedge replacement operation should be adopted (see Section 3.3).

In experimental evaluation, we conclude that $inbanE^2$ is a good choice to enhance the accuracy of minority class, while $inbanE$ is a poor one to classify the minority class samples.

3.3. Hyperedge Replacement. In the process of hyperedge initialization, one generates part of the attribute values randomly. As a result, some of hyperedges are not suitable for sample classification. In order to acquire better performance, one should replace the poor hyperedges by generating new hyperedges, namely, hyperedge replacement.

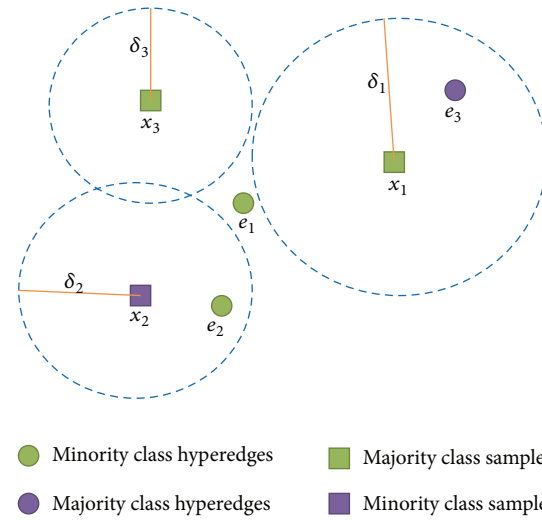

Figure 6: An example of situation 1.

The algorithm divided hyperedge set into upper approximation, lower approximation, boundary region, and negative region. The confidence degree of hyperedges in lower approximation is 1. The confidence degree of hyperedges in boundary region is between 0 and 1. Hyperedges whose confidence degree is 0 belong to negative region. Hyperedges in lower approximation are all retained because they are very helpful for classification. On the contrary, since hyperedges in negative region are counteractive for classification, they will be replaced. For the hyperedges in boundary region, they will be dealt with by a threshold λ. When the confidence degree of a hyperedge is less than λ, it will be replaced. Through the above, one can enhance the pertinence and validity of hyperedge replacement.

It is composed of three steps.

(1) Set the confidence degree threshold of each hyperedge (in this paper $\lambda \in (0.75, 1)$).

(2) Find out those hyperedges whose confidence degree is under the threshold from the hyperedge set.

According to Definitions 7 and 8, one can calculate the confidence degree of each hyperedge following the three cases below (e_i denotes a hyperedge).

Case 1. If $in_B(e) = \phi$, then e_1 is not in any neighborhood of samples, as shown in Figure 6.

In this case, one can assume that e_1 is in the overlapped neighborhood of the nearest five samples. Then the confidence degree of e_1 can be calculated according to Formula (6).

Case 2 ($\mathrm{Conf}_B(e) \geq \lambda$). It means that samples surrounding e_1 have the same class with e_1. Thus e_1 is helpful for the sample classification and should remain.

Case 3 ($0 \leq \mathrm{Conf}_B(e) < \lambda$). Now, let us give an example below to explain the situation (see Figure 7).

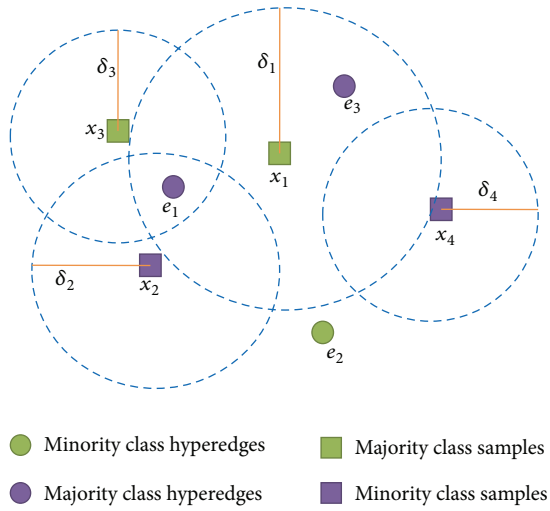

FIGURE 7: Hyperedges whose confidence degree is under λ.

According to Formula (4) and Definition 6, we know $in_C(e_1) = \{x_1, x_2, x_3\}$, $in_C(e_2) = \phi$, and $in_C(e_3) = \{x_1\}$. Then according to formula (8), $\text{Conf}_B(e_1) = 1/3$ and $\text{Conf}_B(e_3) = 0$ can also be calculated. However, as it is difficult to determine the nearest five samples surrounding e_2, we cannot calculate the confidence degree of e_2.

This kind of hyperedges has the same class with few samples surrounding them, which results in the poor effect on classification. Thus, they should be replaced.

(3) Generate new hyperedge and hyperedge replacement.

One selects a hyperedge e_i from the hyperedge set which should be replaced and generates a sample x. Then a new hyperedge e_j is initialized by using x. After that, one can replace e_i with e_j. Repeat the process above until no hyperedge needs to be replaced.

3.4. Neighborhood Hypergraph Algorithm. In this paper, sample classification and hyperedge replacement are based on the neighborhood radius of sample. According to Definition 1, the computational formula of the neighborhood radius of sample, denoted by δ, is as follows [34]:

$$\delta = \min\left(\Delta\left(x_i, s\right)\right) + w \times \text{range}\left(\Delta\left(x_i, s\right)\right), \quad 0 \le w < 1, \tag{14}$$

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 $\text{range}(\Delta(x_i, s))$ denotes the value domain of $\Delta(x_i, s)$.

Here we give the N-HyperGraph (see Algorithm 1).

There are two main parameters in the algorithm: (1) the radius of neighborhood w; (2) threshold of the confidence degree λ. The former is important to control the number of hyperedges. The number is increasing with the increasing of w. The latter is vital to ensure the quality of hyperedges. The higher λ is, the better hyperedges can be obtained. Of course, when the λ is too big, there is no sense that almost all the hyperedges will be replaced.

4. Experimental Designing and Analysis

4.1. Datasets. In order to test the proposed algorithm in this paper, one selects 18 UCI datasets which are downloaded from the machine learning data repository, University of California, at Irvine. The imbalanced rate is from 1.37 to 28.10. There are seven 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.

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 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 [37]. 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) [38] 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. Experimental Methods. In order to evaluate the performance of N-HyperGraph in this paper, one compares it with some other algorithms in related literatures: SVM and J48 (C4.5) [39] implemented on Weka [40], NRSBoundary-SMOTE [12]+C4.5, SMOTE-RSB* [29]+C4.5, CS-EN-HN [7], and N-HyperGraph implemented by Java programming

Input: the training sample set: x, the radius of neighborhood: w.
Output: hyper-edge set: E.
Step 1. (**Initialization**)
$E = \phi$; //Initialize the hyper-edge set.
According to the formula (2), (14), calculate the radius of each sample.
FOR each x in X DO
$j = 0$; //Count the number of hyper-edges generated by each sample.
WHILE ($j < 5$) //Each of samples generates five hyper-edges.
Generate hyper-edge e according to and seven tenth attributions e of inherit attribution values of x;
Calculate the distance between e and each sample according to Formula (2) respectively;
Calculate $inbanX$, $inN(e)$ and $inP(e)$ respectively, according to Definitions 10 and Definition 11;
IF $|inN(e)|/|inP(e)| \geq inbanX$ THEN $D(e) = N$;
ELSE $q = random()$; //q is a random number in $[0, 1]$.
Calculate $nX(N)$ and $nX(P)$ respectively, according to Definition 10;
IF $q \geq \dfrac{|nX(N)|}{|nX(N)| + |nX(P)|}$ THEN $D(e) = N$;
ELSE $D(e) = P$;
END IF
END IF
$E = E \cup \{e\}$; j++;
END WHILE
END FOR
Step 2. (**Training Set Classification**)
Calculate $inbanE$ of hyper-edge set according to Definition 7 and Formula (6);
FOR each x in X DO
Calculate $ne(x, N)$ and $ne(x, P)$ according to Definition 12;
IF $\dfrac{ne(x, N)}{ne(x, P)} \geq inbanE^2$ THEN $D(x) = N$;
ELSE $D(x) = P$;
END IF
END FOR
Calculate the classification accuracy of training data set: *Train-accuracy*.
IF *Train-accuracy* > *0.95* THEN GOTO Step 4;
ELSE GOTO Step 3;
END IF
Step 3. (**Hyper-edge Replacement**)
$hy\_re = 0$; //Number of hyper-edges that should be replaced.
FOR each e_i in E DO
Calculate the confidence-degree $\text{Conf}_B(e_i)$ of e_i according to Definition 8 and Formula (8);
IF $\text{Conf}_B(e_i) < \lambda$ THEN hy_re ++; $E = E - \{e_i\}$;
END IF
END FOR
WHILE ($hy\_re \neq 0$)
Generate a new hyper-edge e_j according to Step 1;
$E = E \cup \{e_j\}$; hy_re −−;
END WHILE
GOTO Step 2;
Step 4. (**Return Hypergraph**)
RETURN E.

Algorithm 1: Neighbor hypergraph (N-HyperGraph).

language. Among these algorithms, SVM classifies the source datasets directly. J48 (C4.5) classifies those datasets after oversampling by NRSBoundary-SMOTE and SMOTE-RSB. The oversampling rate is 100%. CS-EN-HN is an ensemble method of cost-sensitive hypernetwork. It deals with datasets which are discretized by optimal class-dependent discretization (OCDD) [41]. Ones take the value of win the range [0.001–0.6] in N-HyperGraph. We use 10-fold cross validation as validation method.

4.4. Experimental Results and Analysis. Contrastive experiment results on Precision, Recall, F-value, G-means, and AUC among each algorithm are shown in Table 3 to Table 7.

Table 1: Data description.

Dataset	Size	Attribute	Class label (minority : majority)	Class distribution
Bupa	345	6C	01 : 02	145/200
Colic	368	7C 15N	No : yes	136/232
Reprocessed	294	13C	01 : 00	106/188
Machine	209	7C	Others : 2	74/135
Labor	57	8C 8N	Bad : good	20/37
Tic	958	9N	Negative : positive	332/626
Iris	150	4C	Iris-virginica : others	50/100
Seed	210	7C	02 : others	70/140
Vc	310	6C	Normal : Abnormal	100/210
Glass	214	9C	01, 02 : others	68/146
Haberman	306	3C	02 : 01	81/225
Transfusion	748	4C	01 : 00	178/570
Abalone (7 : 15)	494	7C 1N	15 : 07	103/391
Balance-scale	625	4C	B : others	49/576
Abalone (9 : 18)	731	7C 1N	18 : 9	42/689
Yeast (POX : CTY)	483	8C	POX : CYT	20/463
Car	1728	6N	Good : others	69/1659
Yeast (ME2 : others)	1484	8C	ME2 : others	51/1433

C: continuous, N: nominal.

Table 2: Confusion matrix.

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

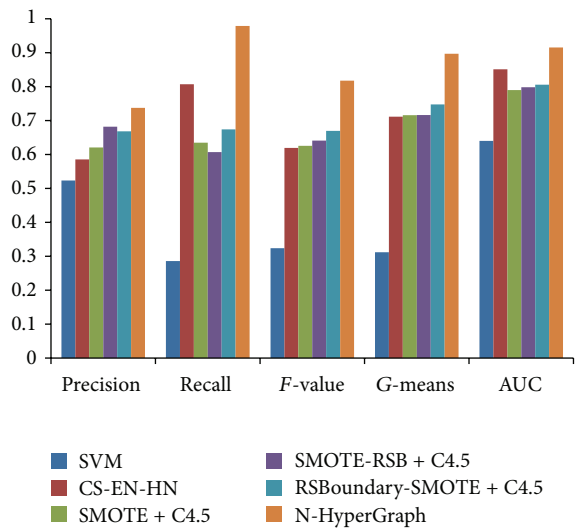

Figure 8: The average value of each indicator.

In order to view the performance on 5 algorithms, the average accuracies of different indicator of 5 algorithms are showed in Figure 8.

Tables 3, 4, 5, 6, and 7 point out that N-HyperGraph has high performance than other four algorithms on most datasets mentioned above. The average result of Precision is enhanced to 0.7374 while it ranges between 0.5233 and 0.6816 for the other four algorithms. The average value of Recall increases to 0.9785 while it changes between 0.2858 and 0.8068. The average value of F-value is up to 0.8173 while it ranges between 0.3235 and 0.6695. Meanwhile, the average values of G-means and AUC increase to 0.8968 and 0.9152 while their values take from 0.3118 to 0.7475 and 0.6398 to 0.8509, respectively.

We can find out from Tables 3–7 and Figure 8 that the Precision performance is unsteady for the proposed algorithms N-HyperGraph. As it is mentioned before, the process of hyperedge initialization is based on the degree of imbalance of training set. The generation of hyperedges depends on the imbalanced degree, which results in the fact that the generated hyperedges incline to the minority sample. Thus, the proposed algorithm is not steady on Precision. But, it can work better than SVM and J48 (C4.5) for Recall, F-value, G-means, and AUC.

In total, the experimental results of N-HyperGraph are better than all of the other algorithms. Since the rough set theory is used in N-HyperGraph, it is more efficient to process the uncertain samples, especially in boundary region of hyperedge set. What is more, weights are calculated through the neighborhood rough set model; thus it makes more hyperedges involve in the class decision of a hyperedge, improving the accuracy. Due to the two aspects above in the proposed algorithm, the results of classification are improved.

As SMOTE oversamples all minority class samples, it decreases the decision space of majority class. Although it can improve Recall of minority class, many majority class samples will be misclassified as minority class, thereby resulting in the decreasing of Precision. SMOTE-RSB filters

Table 3: Precision.

Dataset	SVM	CS-EN-HN	SMOTE + C4.5	SMOTE-RSB* + C4.5	NRSBoundary-SMOTE + C4.5	N-HyperGraph
Bupa	**0.8571**	0.6827	0.5663	0.6581	0.5614	0.5497
Colic	0.6857	0.5887	0.7391	0.8103	0.7687	**0.9151**
Reprocessed	0.0000	0.6344	0.7030	**0.7187**	0.6979	0.5558
Machine	0.0000	0.7542	0.8250	0.8873	**0.9041**	0.9025
Labor	0.9411	**1.0000**	0.6667	0.6667	0.8421	0.7733
Tic	0.9908	0.6534	0.6882	0.8044	0.8100	**0.9942**
Iris	**0.9245**	0.5061	0.9057	0.8703	0.8888	0.8310
Seed	0.9295	**0.9750**	0.9577	0.9577	0.9577	0.8614
Vc	0.0000	0.6133	0.6696	**0.7529**	0.6695	0.4842
Glass	0.8775	**1.0000**	0.6933	0.8214	0.7428	0.8652
Haberman	0.4999	0.2334	0.4516	0.4590	0.4948	**0.7433**
Transfusion	0.4186	0.3139	0.4722	0.5299	0.5000	**0.7116**
Abalone (7 : 15)	0.7951	0.7552	0.8056	0.8155	0.8100	**0.9413**
Balance-scale	0.0000	0.1605	0.0000	0.0000	0.0000	**0.4942**
Abalone (9 : 18)	0.0000	0.3910	0.4167	0.4347	0.5000	**0.9500**
Yeast (POX : CTY)	**1.0000**	0.6371	0.6000	0.9268	0.7736	0.7000
Car	0.5000	0.5100	0.6849	0.6849	**0.6857**	0.6731
Yeast (ME2 : others)	0.0000	0.1272	0.3214	**0.4706**	0.4200	0.3265
Average	0.5233	0.5853	0.6204	0.6816	0.6682	**0.7374**

w = 0.001 to 0.6.

Table 4: Recall.

Dataset	SVM	CS-EN-HN	SMOTE + C4.5	SMOTE-RSB* + C4.5	NRSBoundary-SMOTE + C4.5	N-HyperGraph
Bupa	0.0413	0.7856	0.6483	0.5310	0.6621	**0.8405**
Colic	0.1764	0.6531	0.7500	0.6912	0.7574	**1.0000**
Reprocessed	0.0000	0.5999	0.6698	0.6509	0.6320	**1.0000**
Machine	0.0000	0.8404	8919	0.8514	0.8919	**0.9732**
Labor	0.8000	0.5000	0.5000	0.7000	0.8000	**0.9000**
Tic	0.6536	**1.0000**	0.7711	0.7680	0.7319	**1.0000**
Iris	0.9800	0.5000	0.9600	0.9400	0.9600	**1.0000**
Seed	0.9428	0.9428	0.9714	0.9714	0.9714	**1.0000**
Vc	0.0000	0.9500	0.7700	0.6400	0.7900	**1.0000**
Glass	0.6323	**0.9790**	0.7647	0.6764	0.7647	0.9000
Haberman	0.0246	0.7732	0.5185	0.3457	0.5926	**1.0000**
Transfusion	0.1011	0.9117	0.4775	0.3989	0.5000	**1.0000**
Abalone (7 : 15)	0.6407	**1.0000**	0.8447	0.8155	0.7864	**1.0000**
Balance-scale	0.0000	0.5500	0.0000	0.0000	0.0000	**1.0000**
Abalone (9 : 18)	0.0000	0.8666	0.3571	0.4347	0.3809	**1.0000**
Yeast (POX : CTY)	0.1372	0.8000	0.4500	0.4751	0.8039	**1.0000**
Car	0.0145	**1.0000**	0.7246	0.7246	0.6956	**1.0000**
Yeast (ME2 : others)	0.0000	0.8700	0.3529	0.3137	0.4118	**1.0000**
Average	0.2858	0.8068	0.6346	0.6071	0.6740	**0.9785**

w = 0.001 to 0.6.

the synthetic samples more strictly than SMOTE, because few synthetic samples are generated when datasets are highly imbalanced. Thus, compared with SMOTE, its improvement is not obvious. RSBoundary-SMOTE takes neighborhood rough set into consideration and emphasizes resampling for minority class samples which belong to boundary region and thus improves the F-value. However, N-HyperGraph replaces hyperedges repetitively according to neighborhood rough set. The distribution of hyperedge set draws near to the true distribution of samples gradually, which makes a more obvious improvement on classification performance. Besides, since CS-EN-HN can just deal with discrete data, too much information loss of data makes its performance worse than N-HyperGraph.

Table 5: F-value.

Dataset	SVM	CS-EN-HN	SMOTE + C4.5	SMOTE-RSB* + C4.5	NRSBoundary-SMOTE + C4.5	N-HyperGraph
Bupa	0.0789	**0.7193**	0.6045	0.5878	0.6076	0.6422
Colic	0.2807	0.5688	0.7445	0.7460	0.7630	**0.9530**
Reprocessed	0.0000	0.5374	0.6698	0.6831	0.6633	**0.7118**
Machine	0.0000	0.8404	0.8571	0.8690	0.8980	**0.9242**
Labor	**0.8648**	0.6666	0.5714	0.6829	0.8205	0.8171
Tic	0.7876	0.7900	0.7273	0.7858	0.7689	**0.9926**
Iris	**0.9514**	0.4835	0.9320	0.9038	0.9230	0.9045
Seed	0.9361	0.9545	**0.9645**	**0.9645**	**0.9645**	0.9209
Vc	0.0000	**0.7295**	0.7163	0.6919	0.7248	0.6501
Glass	0.7350	**0.9890**	0.7273	0.7419	0.7536	0.8808
Haberman	0.0470	0.3475	0.4828	0.3944	0.5393	**0.8396**
Transfusion	0.1628	0.4582	0.4749	0.4551	0.5000	**0.8119**
Abalone (7 : 15)	0.7096	0.8595	0.8246	0.8155	0.7980	**0.9683**
Balance-scale	0.0000	0.2326	0.0000	0.0000	0.0000	**0.6447**
Abalone (9 : 18)	0.0000	0.5071	0.3846	0.3076	0.4324	**0.9709**
Yeast (POX : CTY)	0.2413	0.6493	0.5143	**0.8261**	0.7885	0.8066
Car	0.0282	0.6728	0.7042	0.7042	0.6906	**0.7976**
Yeast (ME2 : others)	0.0000	0.1924	0.3364	0.3765	0.4158	**0.4750**
Average	0.3235	0.6191	0.6252	0.6409	0.6695	**0.8173**

w = 0.001 to 0.6.

Table 6: G-means.

Dataset	SVM	CS-EN-HN	SMOTE + C4.5	SMOTE-RSB* + C4.5	NRSBoundary-SMOTE + C4.5	N-HyperGraph
Bupa	0.2026	**0.7456**	0.6441	0.6518	0.6433	0.5592
Colic	0.4008	0.5828	0.7740	0.7910	0.8100	**0.9687**
Reprocessed	0.0000	0.5973	**0.7503**	0.7466	0.7311	0.7247
Machine	0.0000	0.8371	0.8941	0.8949	0.9196	**0.9377**
Labor	0.8000	0.7071	0.6576	0.7534	**0.8574**	0.8232
Tic	0.8015	0.8466	0.7926	0.8318	0.8156	**0.9959**
Iris	0.4427	0.5112	**0.9550**	0.9349	0.9499	0.9427
Seed	0.6473	0.9623	**0.9750**	**0.9750**	**0.9750**	0.9511
Vc	0.0000	0.7729	0.7941	0.7589	**0.8021**	0.6831
Glass	0.7141	**0.9892**	0.8026	0.7938	0.8187	0.8897
Haberman	0.2957	0.4884	0.6332	0.5431	0.6808	**0.9005**
Transfusion	0.1628	0.4582	0.6308	0.5956	0.6496	**0.8119**
Abalone (7 : 15)	0.6626	0.9565	0.8940	0.8808	0.8649	**0.9909**
Balance-scale	0.0000	0.6016	0.0000	0.0000	0.0000	**0.9428**
Abalone (9 : 18)	0.0000	0.8521	0.5884	0.4833	0.6100	**0.9978**
Yeast (POX : CTY)	0.3704	0.7088	0.6665	0.8552	0.8630	**0.9879**
Car	0.1195	0.9790	0.8453	0.8453	0.8285	**0.9888**
Yeast (ME2 : others)	0.0000	0.5086	0.5861	0.5566	0.6352	**0.9485**
Average	0.3118	0.7113	0.7158	0.7162	0.7475	**0.8968**

w = 0.001 to 0.6.

5. Conclusion

In this paper, one proposed a new algorithm based on hypernetwork called N-HyperGraph to solve the problem of classifying imbalance dataset. At first, hyperedge set is divided according to rough set theory. Then, some poor hyperedges are replaced by combining with the imbalanced degree, in order to improve the accuracy. The experimental results on 18 UCI datasets with different degree of imbalance show that the classification result of the proposed algorithm N-HyperGraph improves obviously in contrast with another four algorithms. However, the algorithm N-HyperGraph will

Table 7: AUC.

Dataset	SVM	CS-EN-HN	SMOTE + C4.5	SMOTE-RSB* + C4.5	NRSBoundary-SMOTE + C4.5	N-HyperGraph
Bupa	0.5181	**0.8928**	0.6468	0.6652	0.6401	0.6427
Colic	0.5645	0.8265	0.7960	0.7855	0.8102	**0.9699**
Reprocessed	0.5000	0.6750	**0.7896**	0.7565	0.7817	0.7662
Machine	0.5000	0.9201	0.9199	0.9359	0.9430	**0.9437**
Labor	0.8864	0.7500	0.7500	0.7655	0.8243	**0.8875**
Tic	0.8252	**1.0000**	0.8638	0.8941	0.8848	0.9959
Iris	**0.9700**	0.7500	0.9408	0.9197	0.9468	0.9450
Seed	0.9535	0.9780	0.9730	**0.9782**	**0.9782**	0.9535
Vc	0.5000	**0.9750**	0.8107	0.8380	0.8277	0.7381
Glass	0.7956	**0.9894**	0.8442	0.8640	0.8637	0.9400
Haberman	0.5079	0.5866	0.6255	0.6174	0.6636	**0.9186**
Transfusion	0.5286	0.8558	0.6813	0.7007	0.7048	**0.9114**
Abalone (7 : 15)	0.7986	**1.0000**	0.8901	0.9046	0.8627	0.9910
Balance-scale	0.5000	0.6256	0.5000	0.5000	0.5000	**0.9452**
Abalone (9 : 18)	0.6611	0.9333	0.7294	0.6514	0.7065	**0.9978**
Yeast (POX : CTY)	0.5000	0.8999	0.6881	0.8685	0.8762	**0.9881**
Car	0.5069	0.9792	0.9775	0.9775	**0.9911**	0.9888
Yeast (ME2 : others)	0.5000	0.6790	0.7894	0.7412	0.6916	**0.9504**
Average	0.6398	0.8509	0.7898	0.7980	0.8054	**0.9152**

w = 0.001 to 0.6.

cost much time, due to calculating the distance between hyperedge and sample. Thus, how to reduce the running time of the algorithm is our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

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

References

[1] 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.

[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, Springer, Berlin, Germany, 2012.

[3] 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.

[4] A. Addis, G. Armano, and E. Vargiu, "Experimentally studying progressive filtering in presence of input imbalance," in *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 272, pp. 56–71, Springer, Berlin, Germany, 2013.

[5] F. Provost, "Machine learning from imbalanced data sets," in *Proceedings of the Workshop Learning from Imbalanced Data Sets (AAAI '02)*, 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] K. W. Sun, *Evolutionary Hypernetwork Based Classification of High Dimensional and Imbalanced Data*, Chongqing University of Posts and Telecommunications, Chongqing, China, 2013 (Chinese).

[8] 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.

[9] 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.

[10] Z. M. Yang, L. Y. Qiao, and X. Y. Peng, "Research on data mining method for imbalanced dataset based on improved SMOTE," *Acta Electronica Sinica B*, vol. 35, no. 2, pp. 22–26, 2007 (Chinese).

[11] 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.

[12] F. Hu and H. Li, "A novel boundary oversampling algorithm based on neighborhood rough set model: NRSBoundary-SMOTE," *Mathematical Problems in Engineering*, vol. 2013, Article ID 694809, 10 pages, 2013.

[13] Z. H. Zhou, "Ensemble learning," in *Encyclopedia of Biometrics*, S. Z. Li, Ed., pp. 270–273, Springer, Berlin, Germany, 2009.

[14] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD*, vol. 2838 of *Lecture Notes in Computer Science*, pp. 107–119, Springer, Berlin, Germany, 2003.

[15] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: improving classification performance when training data is skewed," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, Tampa, Fla, USA, December 2008.

[16] J. Błaszczyński, M. Deckert, J. Stefanowski, and S. Wilk, "IIvotes ensemble for imbalanced data," *Intelligent Data Analysis*, vol. 16, no. 5, pp. 777–801, 2012.

[17] P. K. Chan and J. S. Stilfo, "Toward scalable learning with non-uniform class and cost distribution: a case study in credit card fraud detection," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pp. 164–168, 1998.

[18] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI '01)*, pp. 973–978, New York, NY, USA, August 2001.

[19] Z. Zhou and X. Liu, "On multi-class cost-sensitive learning," *Computational Intelligence. An International Journal*, vol. 26, no. 3, pp. 232–257, 2010.

[20] P. Momingos, "A general method for making classifiers cost-sensitive," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155–164, 1999.

[21] W. Fan, S. Stolfo, and J. Zhang, "AdaCost: misclassification cost-sensitive boosting," in *Proceedings of the 16th International Conference on Machine Learning*, pp. 97–105, 1999.

[22] D. E. Rumelhart and D. A. Norman, *Accretion, Tuning and Restructuring: Three Modes of Learning*, Defense Technical Information Center, Fort Belvoir, Va, USA, 1976.

[23] B. Zhang, "Hypernetworks: a molecular evolutionary architecture for cognitive learning and memory," *IEEE Computational Intelligence Magazine*, vol. 3, no. 3, pp. 49–63, 2008.

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

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

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

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

[28] 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.

[29] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB$_*$: 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.

[30] 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.

[31] C. Berge, *Graphs and Hypergraphs*, Elsevier, New York, NY, USA, 1973.

[32] C. Berge, *Hypergraphs: The Theory of Finite Sets*, Elsevier Science, Amsterdam, The Netherlands, 1989.

[33] T. Y. Lin, "Granular computing on binary relations I: data mining and neighborhood systems," in *Rough Sets in Knowledge Discovery*, vol. 1, pp. 289–318, 1998.

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

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

[36] J. Wang, L. X. Jin, and K. W. Sun, "Chinese text categorization based on evolutionary hyper network," *Journal of Jiangsu University*, vol. 34, no. 2, pp. 196–201, 2013.

[37] 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.

[38] 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.

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

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

[41] L. L. Liu, A. K. C. Wang, and Y. Wang, "A global optimal algorithm for class-dependent discretization of continuous data," *Intelligent Data Analysis*, vol. 8, no. 2, pp. 151–170, 2004.