Diversified Sensitivity-Based Undersampling for Imbalance Classification Problems

Wing W. Y. Ng, *Member, IEEE,* Junjie Hu, Daniel S. Yeung, *Fellow, IEEE,* Shaohua Yin, and Fabio Roli, *Fellow, IEEE*

***Abstract*—Undersampling is a widely adopted method to deal with imbalance pattern classification problems. Current methods mainly depend on either random resampling on the majority class or resampling at the decision boundary. Random-based undersampling fails to take into consideration informative samples in the data while resampling at the decision boundary is sensitive to class overlapping. Both techniques ignore the distribution information of the training dataset. In this paper, we propose a diversified sensitivity-based undersampling method. Samples of the majority class are clustered to capture the distribution information and enhance the diversity of the resampling. A stochastic sensitivity measure is applied to select samples from both clusters of the majority class and the minority class. By iteratively clustering and sampling, a balanced set of samples yielding high classifier sensitivity is selected. The proposed method yields a good generalization capability for 14 UCI datasets.**

***Index Terms*—Diversified sensitivity undersampling (DSUS), imbalance data, sample selection.**

I. Introduction

THE PROBLEM of class imbalance occurs in many pattern classification problems like network security, medical diagnostics, and digit recognition. In security and medical problems, the number of samples belonging to the normal class is usually much larger than that of the dangerous or abnormal class. For multiclass problems, the numbers of samples in different classes are inherently imbalanced. For instance, in a ten-digit recognition problem, the number of samples in each class could be one out of ten of the total number of samples which leads to a 1:9 imbalance pattern classification problem for each class [1].

Undersampling and oversampling are widely applied to balance the imbalance data for classifier training. Oversampling generates new artificial samples of the minority class. Undersampling selects a portion of samples from the majority class to balance the data. The major advantage of undersampling is that all training samples are real. Random-based undersampling and resampling at the decision boundary are two major categories of undersampling methods. Random-based undersampling selects samples from the majority class randomly without concerning the usefulness of samples. In contrast, resampling is sensitive to class overlapping. Moreover, both approaches ignore the distribution of samples in both classes which may yield a poor generalization to future samples in the majority class located far from the decision boundary. Therefore, in this paper, we propose the diversified sensitivity-based undersampling method (DSUS) which selects useful samples yielding high stochastic sensitivity with respect to the currently trained classifier. To prevent selecting samples around the decision boundary only and perverse the data distribution, samples in the majority class are clustered and only a representative sample of each cluster participates in the selection process. It also speeds up the selection process.

Section II provides a brief review on related works. The DSUS is presented in Section III. Section IV shows experimental comparisons between the DSUS and current methods, and we conclude this paper in Section V.

II. Related Works

If a classifier is trained using a highly imbalanced dataset directly, a poor performance is expected. When dealing with such problems, some classical algorithms (e.g., neural networks) tend to misclassify samples of the minority class to the majority class which yields a high average classification accuracy. However, the misclassification cost for the minority class is usually much greater than that of the majority one. In addition to the imbalance ratio, as pointed out in [2], multiclass problems may also introduce multimajority and multiminority classes. This further complicates the issue here. On the other hand, the overlapping between the majority and the minority classes introduces a further difficulty to the imbalance problem [3]–[5].

Resampling and classifier modification are two major approaches in dealing with the imbalance problem. The aim of resampling is to balance the dataset before the classifier learning. Resampling either oversamples the minority class by creating fake samples for the minority class or undersamples the majority class by selecting a portion of samples from it. The major advantages of resampling are that no modification

Manuscript received December 23, 2013; revised April 25, 2014 and November 1, 2014; accepted November 9, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61003171, Grant 61272201, and Grant 61003172, and in part by the Program for New Century Excellent Talents in University of China under Grant NCET-11-0162. This paper was recommended by Associate Editor Q. Shen.

W. W. Y. Ng, D. S. Yeung, and S. Yin are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: wingng@ieee.org).

J. Hu is with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

F. Roli is with the Department of Electrical and Electronic Engineering of the University of Cagliari, Cagliari 09123, Italy.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2372060

to classifier is needed and the balanced dataset can be reused in other applications or learning tasks [6]. Resampling could be further divided into random, ensemble [4], and active learning (resampling at the decision boundary)-based approaches.

For an imbalance pattern classification problem with N_n samples in the majority class and N_p samples in the minority class, where $N_n > N_p$, the random undersampling (RUS) [3] selects N_p samples from the majority class randomly to balance the two classes. In contrast, the random oversampling (ROS) [3] randomly create $N_n - N_p$ fake samples around samples of the minority class to balance the dataset. However, both the ROS and the RUS suffer from the problem of losing distribution information. The RUS may miss informative samples while the ROS may yield overfitting because of the creation of many duplicated training samples. When the imbalance ratio is large, undersampling may remove too many negative samples in the majority class and loose important information about the majority class [7].

Synthetic minority over-sampling technique (SMOTE) [8] creates artificial samples between two samples in the minority class. Gao *et al*. [6] combined the particle swarm optimization with the SMOTE to train radial basis function neural network (RBFNN). They preserve the distribution of the minority class. Several modifications are made to the SMOTE, e.g., borderline-SMOTE [9], safe-level SMOTE [10], local neighborhood-based SMOTE [11], and rough sets theory-based SMOTE [12]. In [13], the SMOTE is also applied to deal with imbalanced datasets for linguistic fuzzy rule-based classification problems.

Owing to the random nature in both RUS and ROS, ensemble methods are usually adopted to reduce the variance of performances and produce better final classifiers. Recently, an inverse RUS (IRUS) [14] is proposed to randomly undersample the majority class, but not to balance the two classes. In each of the base classifiers of the ensemble, the IRUS resamples r samples from the majority class, where $r < N_p$. Then, base classifiers trained by different inverse datasets are fused to create a composite decision boundary between the two classes. In contrast to the bagging-based IRUS, RUSBoost [15] and MSMOTEBoost [16] adopt boosting for RUS and SMOTE for undersampling and oversampling, respectively. Easyensemble (EE) and Balancecascade (BC) are two hybrid ensemble undersampling methods [17] which combine bagging and boosting. In EE, the dataset is divided into several subsets by random resampling with replacement and each subset is used to train a base classifier of the ensemble with AdaBoost. BC performs the bagging part in a supervised manner and removes samples that can be classified correctly with high confidence from future selections. Among ensemble methods, [18] finds that bagging outperforms boosting, especially when noise appears in the dataset.

Active learning-based method is also applied to solve the imbalance problem by selecting informative samples from both classes. In the support vector machine (SVM) training, only support vectors located near the decision boundary (hyperplane) are useful. Moreover, samples located near the decision boundary are less imbalance in comparison to the whole dataset. So, resampling at the decision boundary is an effective active learning method, especially for SVM [19]. The LA-SVM [20] retrains a SVM using informative samples selected from the current decision boundary iteratively. As an instance of the classifier modification approach, a partial least square-based asymmetric classifier is proposed in [21] to enhance the generalization accuracy of samples in the minority class. Another way to modify the SVM for imbalance problems is via margin calibration [22] by an inversed proportional regularized penalty to reweight the imbalance samples and a margin compensation to drift the margin. Similarly, neural networks can also be modified by output threshold moving to change the class decision of neural networks based on a cost-sensitive objective function [23].

III. Diversified Sensitivity Based Undersampling

The DSUS consists of three major components: 1) clustering samples in the majority class; 2) undersampling via a sample selection using the stochastic sensitivity measure (SM); and 3) a RBFNN trained by using training samples selected by the SM. Fig. 1 shows the workflow of the DSUS and Algorithm 1 shows details of the DSUS. Both clustering and RBFNN training use off-the-shelf methods. In this paper, the k-means is used which could be replaced by other clustering methods. The RBFNN training algorithm will be introduced in Section III-A, while the SM evaluation criterion will be introduced in Section III-B. The time complexity of the DSUS will be discussed in Section III-C.

Given an imbalance dataset with N_p candidate samples in the minority (positive) class (U_p) and N_n candidate samples in the majority (negative) class (U_n), the DSUS will first cluster samples in the majority class and samples in the minority class into k clusters separately, where $k = \sqrt{N_p}$. The value of k remains the same after iterations and is not updated when N_p becomes smaller after iterative sample selection. For each center of the clusters in both classes, the sample located closest to it is added to the initial training set to train the initial RBFNN. As shown in Fig. 1, samples of the initial training set will be removed from the candidate set. The SM for sample selection is computed based on both sample location in the input space and also the trained RBFNN. The choice of $k = \lfloor\sqrt{N_p}\rfloor$ is *ad-hoc* and it is intended to provide a balanced number of samples between two classes in each step. In Section IV-A, we show that the choice of $k = \lfloor\sqrt{N_p}\rfloor$ outperforms $k = \lfloor\sqrt{N_n}\rfloor$, $k = \lfloor\sqrt{N_p + N_n/2}\rfloor$ and $k = \lfloor\sqrt{N_p + N_n}\rfloor$. This is because N_n is too large and the later two choices make the DSUS selecting a large number of samples in each step which reduces the effectiveness of the DSUS.

The number of samples in the majority class is usually very large, so we cluster them into N_p clusters in each turn before the sample selection. It helps to reduce the time for evaluation and also provides a diversified sample set for selection. This step allows the DSUS to preserve the distribution information and is beneficial to the generalization for future unseen samples generated from the same distribution of training samples. Moreover, the clustering algorithm is selected by the user and could affect the effectiveness of the DSUS if a very bad clustering algorithm is used.

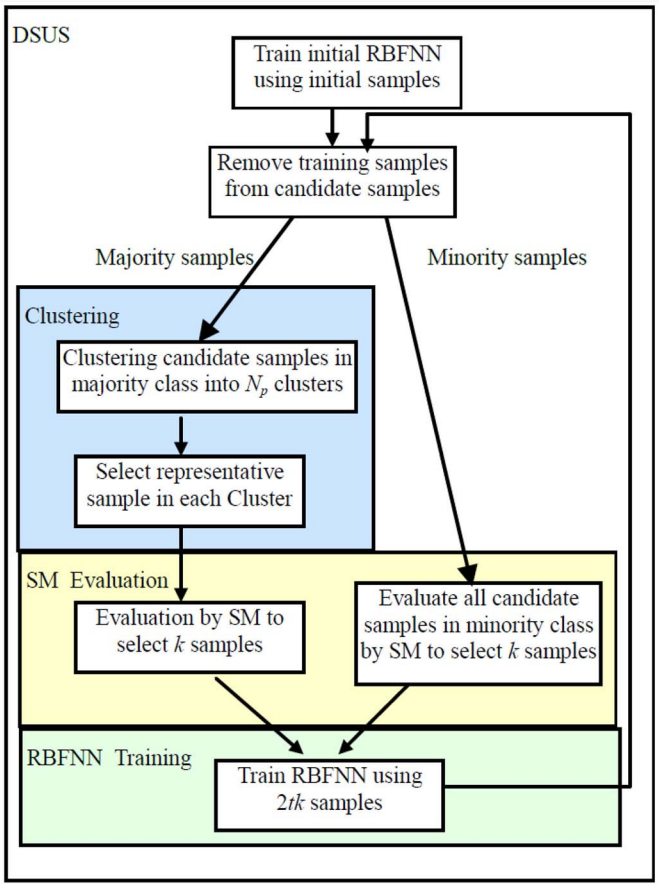

Fig. 1. Workflow of the DSUS.

Algorithm 1 DSUS

Step 1**:** Training the initial RBFNN
a) Cluster both U_n and U_p into $k = \lfloor\sqrt{N_p}\rfloor$ clusters each.
b) Set both P_0 and R_0 to be empty sets
c) For each of k clusters of the minority class, add the sample located closest to its center to P_0.
d) For each of k clusters of the majority class, add the sample located closest to its center to R_0.
e) $U_p = U_p - P_0$, $U_n = U_n - R_0$, $S = P_0 \bigcup R_0$ and $b = 0$
Step 2**:** Train a RBFNN using S
while $n_p \geq k$ **do**
  Step 3**:** Find representative samples in the majority class(C)
  a) Set C, R_b and P_b to be empty sets
  b) n_p = size(U_p) and $b = b + 1$
  c) Cluster U_n into n_p clusters.
  d) For each of n_p clusters of U_n, add the sample located closest to its center to C
  Step 4**:** Sample Selection using the Sensitivity Measure
  a) Compute the SM value for each sample in both C and U_P with the current RBFNN
  b) Add k samples from C yielding the largest SM values to R_b
  c) Add k samples from U_P yielding the largest SM values to P_b
  d) $U_p = U_p C P_b$ and $U_n = U_n C R_b$, $S = S \bigcup P_b \bigcup R_b$
  Step 5**:** Train a RBFNN using S
end while

The DSUS selects a sample located closest to the center of each of these N_p clusters as representative samples and then their SM values are computed. The k samples yielding the largest SM values will be selected from the majority class. Similarly, k samples yielding the largest SM values will also be selected from the minority class. These $2k$ samples are then added to the initial training set to form a balanced training dataset for the RBFNN. In each turn of iteration, the training dataset consists of $2tk$ samples, where t denotes the number of iterations including the initial turn. The value of t is at most equal to k. The DSUS is particularly suitable for RBFNN training because RBFNN training relies on distribution information to compute parameters of its hidden neurons. Selected samples are removed from the candidate set and these procedures repeat until the number of samples in the minority class is less than k.

A. RBFNN Training Algorithm

An RBFNN is defined as follows:

$$f(\mathbf{x}) = \sum_{j=1}^{M} w_j \exp\left(\frac{\| \mathbf{x} - \mathbf{u}_j \|^2}{-2v_j^2}\right) \quad (1)$$

where $\mathbf{x}$, $f(\mathbf{x})$, $\mathbf{u}_j$, v_j, w_j, and M denote the input, the RBFNN output of x, the center vector of the jth hidden neuron, the width of the jth hidden neuron, the weight between the jth hidden neuron and the output neuron, and the number of hidden neurons, respectively. We could represent (1) in a matrix form: $F = WH$, where W and H denote a vector consisting of all w_j and an $N \times M$ matrix consisting of all hidden neuron outputs of all training samples, respectively. An RBFNN consists of three layers: 1) input; 2) hidden; and 3) output layers. The number of neurons on the input layer is equal to the number of input features. For two-class problems, the output layer consists of one output neuron. The decision of majority or minority classification is made by thresholding of the value of the output neuron computed using (1). The classical two-phase training algorithm in [24] is used to select parameters of the RBFNN. In the first phase, hidden neuron parameters are determined using unsupervised clustering methods based on the distribution information of the training dataset. The connection weights are then determined using a supervised method to learn the input–output relationship in the second phase.

In the first phase, center vectors ($\mathbf{u}_j$) are computed via a k-means clustering algorithm with M equal to the square root of the number of training samples in the current turn. It was shown in [25] that selecting the number of hidden neurons using SM and the localized generalization error [25] yields a RBFNN with better generalization capability. However, we use a classical method instead of the SM-based method in this paper to select the number of hidden neurons for a fair comparison to other imbalance learning methods. The width

parameter (v_j) is selected to be the average value of distances between nearest centers.

In the second phase, the weight values have a linear relationship to the output neuron and hidden neuron outputs. So, instead of applying the recursive least square method [24], a simple least square method is used to compute the connection weights by $W = FH^{+}$, where H^{+} denotes the pseudo-inverse of the matrix H. All RBFNNs in our experiments follow the training algorithm as described here.

B. SM-Based Sample Selection

In [25], the SM of a RBFNN was proposed as part of the localized generalization error model (L-GEM) for the RBFNN architecture selection. However, the computation method in [25] does not compute the SM for individual sample. So, in [26] a new SM formulation is proposed for the SVM which measures sensitivity of individual sample for hyperparameter selection for the SVM. In this paper, we propose to use the SM to evaluate individual candidate sample for undersampling. The SM of a sample ($\mathbf{x}_b$) measures the output fluctuations of a given RBFNN when the n-dimensional inputs of the sample are perturbed by an n-dimensional vector $\Delta\mathbf{x}$. In the context of a pattern recognition problem, the input perturbation may be interpreted as future unseen samples in a neighborhood of a training sample such that, for a small perturbation, those future, unseen samples may be recognized similar to the training sample. Intuitively, this follows the basic assumption that unseen and training samples follow the same distribution pattern.

We define a Q-neighborhood for the candidate sample $\mathbf{x}_b$ in (2) which covers all unseen, nontraining samples similar to $\mathbf{x}_b$. In other words, unseen, nontraining samples with a difference from $\mathbf{x}_b$ in each feature less than a preselected value Q are considered to be similar to $\mathbf{x}_b$. The value of Q provides an upper bound to the maximum expected difference between candidate samples and future samples under investigation

$$S_Q(\mathbf{x}_b) = \{\mathbf{x} | \mathbf{x} = \mathbf{x}_b + \Delta\mathbf{x}, |\Delta\mathbf{x}_i| \le Q, i = 1, 2, \ldots, n\} \quad (2)$$

where $\mathbf{x}_i$ denotes the ith input feature of $\mathbf{x}$. For each candidate sample, we evaluate the expectation of squared RBFNN output differences (Δy) between all samples in $S_Q(\mathbf{x}_b)$ and the $\mathbf{x}_b$ as the SM of $\mathbf{x}_b$ in

$$\begin{aligned} SM &= \int_{S_Q(\mathbf{x})} (f(\mathbf{x}_b) - f(\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} \\ &= \int_{S_Q(\mathbf{x})} \left(f(\mathbf{x}_b)^2 - 2f(\mathbf{x}_b) f(\mathbf{x}) + f(\mathbf{x})^2 \right) p(\mathbf{x}) d\mathbf{x} \\ &= f(\mathbf{x}_b)^2 - 2f(\mathbf{x}_b) \int_{S_Q(\mathbf{x})} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \int_{S_Q(\mathbf{x})} f(\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x} \\ &= f(\mathbf{x}_b)^2 - 2f(\mathbf{x}_b) I_1 + I_2. \end{aligned} \quad (3)$$

I_1 and I_2 are defined in (4) and (5) with the assumption that input features are assumed to be independent to each other. Since we have no prior knowledge about unseen samples in the Q-neighborhood, all unseen samples are assumed to have the same chance to appear in future, i.e., uniformly distributed. So, $p(\mathbf{x}) = 1/(2Q)^n$. A detailed derivation of (3)–(5) could be found in [26]

$$\begin{aligned} I_1 &= \int_{S_Q(\mathbf{x})} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{(2Q)^n} \int_{S_Q(\mathbf{x})} f(\mathbf{x}_b + \Delta\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{(2Q)^n} \int_{S_Q(\mathbf{x})} \sum_{j=1}^{M} w_j \exp\left(\frac{\sum_{i=1}^{n} \left(\Delta x_i + x_{bi} - u_{ji} \right)^2}{-2v_j^2} \right) dx \\ &= \left(\frac{\sqrt{\pi}}{2\sqrt{2}Q} \right)^n \sum_{j=1}^{M} w_j \prod_{i=1}^{n} \left(v_j \left(\operatorname{erf}\left(\frac{x_{bi} - u_{ji} + Q}{\sqrt{2v_j^2}} \right) \right.\right. \\ &\qquad \left.\left. - \operatorname{erf}\left(\frac{x_{bi} - u_{ji} - Q}{\sqrt{2v_j^2}} \right) \right) \right) \end{aligned} \quad (4)$$

where u_{ji} denotes the ith element of the center vector u_j

$$\begin{aligned} I_2 &= \int_{S_Q(\mathbf{x})} f(\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{(2Q)^n} \int_{S_Q(\mathbf{x})} f(\mathbf{x}_b + \Delta\mathbf{x})^2 d\mathbf{x} \\ &= \frac{1}{(2Q)^n} \int_{S_Q(\mathbf{x})} \left(\sum_{j=1}^{M} w_j \exp\left(\frac{\sum_{i=1}^{n} \left(\Delta x_i + x_{bi} - u_{ji} \right)^2}{-2v_j^2} \right) \right)^2 d\mathbf{x} \\ &= \left(\frac{\sqrt{\pi}}{4Q} \right)^n \sum_{j,k=1}^{M} \left(\left(\frac{\sqrt{2v_j^2 v_k^2 \left(v_k^2 + v_j^2 \right)}}{v_k^2 + v_j^2} \right)^n \right. \\ &\qquad \prod_{i=1}^{n} \left(\left(\operatorname{erf}\left(\frac{\left(v_k^2 + v_j^2 \right)(x_{bi} + Q) - \left(v_k^2 u_{ji} + v_j^2 u_{ki} \right)}{\sqrt{2v_j^2 v_k^2 \left(v_k^2 + v_j^2 \right)}} \right) \right.\right. \\ &\qquad \left.\left. \left. - \operatorname{erf}\left(\frac{\left(v_k^2 + v_j^2 \right)(x_{bi} - Q) - \left(v_k^2 u_{ji} + v_j^2 u_{ki} \right)}{\sqrt{2v_j^2 v_k^2 \left(v_k^2 + v_j^2 \right)}} \right) \right) \right) \right). \end{aligned} \quad (5)$$

A training sample yielding a large SM value indicates that the RBFNN is not certain about its output classification with respect to a minor change in the input values of the sample. Hence, the RBFNN does not have enough knowledge about this particular training sample and it may call for further learning on this sample to improve the performance of the RBFNN. In contrast, RBFNN is well learnt of a training sample if its SM value is low. By iteratively adding samples with large SM values to the training sample set, the RBFNN learns more knowledge about the classification problem. Moreover, the samples selected by the SM ranking for imbalance learning are balanced by Steps 4(b) and (c) in Algorithm 1. The proposed DSUS method

can be adapted to widely used classifiers, e.g., LS-SVM and multilayer perceptron neural networks by replacing the SM of RBFNN by the SMs presented in [26] and [27], respectively.

C. Time Complexity of DSUS

The DSUS consists of three major steps: 1) clustering of samples in the majority class; 2) computation of the SM; and 3) the training of the RBFNN. Their time complexities are $O(N_pN_nld)$, $O(N_p^{1.5}(N_p+N_n)d)$, and $O(N_p^2ld)$, respectively, where l and d denote the number of steps used by k-means clustering and the number of input features, respectively. The overall time complexity of the DSUS is $O(Np^{1.5}(Np+Nn)d)$. This time complexity is worse than that of the RUS ($O(N_p^{1.5}ld)$) and the ROS ($O(N_n^{1.5}ld)$), but the DSUS yields significantly better results.

IV. Experiment

Firstly, we use four artificial imbalance datasets to compare training samples selected by the DSUS, the RUS, and the resampling at the decision boundary-based undersampling for SVM (LA-SVM) in Section IV-A. We show experiments on five artificial datasets with different ratios of overlapping and evaluate the DSUS for dealing with both the imbalance and the overlapping in Section IV-B. We will test the DSUS when clustering is not meaningful and the decision boundary is not clear in Section IV-C. Section IV-D provides experimental comparisons of the DSUS with other random-based, ensemble, and resampling at the decision boundary methods on 14 UCI datasets with different imbalance ratios and sizes.

A. Samples Selected by Different Undersampling Schemes

The proposed DSUS combines sample selection on the uncertain region with distribution information. So we compare it with resampling at the decision boundary (LA-SVM) [20] and random-based method to show the effectiveness of the DSUS. RUS is one of the key components of many current undersampling methods (see [17]), and it is selected as the representative random-based method. Both the RUS and the DSUS use a RBFNN classifier while the LA-SVM uses a SVM classifier.

The four 2-D artificial datasets shown in Fig. 2 represent four different situations of imbalance datasets. Fig. 2(a) shows the first imbalance dataset (uniform) created by a uniform distribution where one out of nine samples at the left-bottom corner belong to the minority class (red circle in figures) while others belong to the majority class (blue cross). The second imbalance dataset (Gaussian-1) is created by two Gaussian distributions with a 1:9 ratio of samples as shown in Fig. 2(b). The third imbalance dataset (Gaussian-2) consists of nine Gaussian distributions with the same number of samples being arranged in a 3×3 grid as shown in Fig. 2(c). Samples of the Gaussian located in the middle belong to the minority class while others belong to the majority class. Fig. 2(d) shows the fourth imbalance dataset (complex) which also has a 1:9 ratio

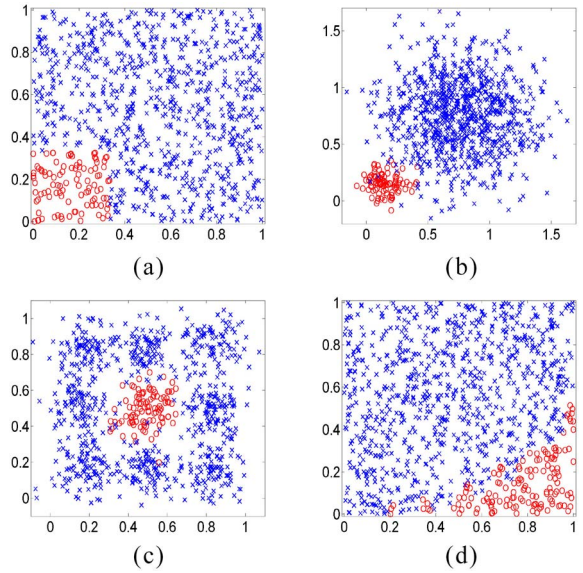

Fig. 2. Four artificial imbalance datasets. (a) Uniform. (b) Gaussian-1. (c) Gaussian-2. (d) Complex.

of samples in the minority and the majority classes. In comparison with simple distributions in the other three datasets, the complex dataset has a complicated decision boundary between the two classes.

Fig. 3 shows the first four turns of undersampling by (a) the DSUS, (b) the RUS, and (c) the LA-SVM for the uniform dataset. Figs. 4–6 show the undersampling for Gaussian-1, Gaussian-2, and complex datasets, respectively. Figs. 3(a), 4(a), 5(a), and 6(a) show that the DSUS maintains the distribution of the original dataset roughly while it tends to select more samples around the decision boundary. Random-based RUS method select samples from both classes equally yet randomly. In contrast, LA-SVM focuses on the decision boundary while ignoring other samples. Then, we train RBFNNs using undersampling results of the DSUS, the RUS, and the LA-SVM. To show the effectiveness and necessity of the DSUS, the two components of the DSUS: sample clustering ($\text{cluster}_{\text{only}}$) and sample selection by SM (SM_{only}) are also compared. In $\text{cluster}_{\text{only}}$, we randomly select k samples located closest to centers of clusters in the same way of Step 3 in Algorithm 1 as training samples and repeat until samples in the minority class being used up. For SM_{only}, we follow Algorithm 1, but select samples using the SM for all samples in majority class by passing Step 3 in Algorithm 1. The 50% of samples in each dataset are selected as candidate set for RBFNN training while the other 50% samples left are used as testing samples and will not participate in any training. All methods perform undersampling on the 50% candidate sets. Random selection of candidate and testing datasets are repeated ten times. Table I shows the averages and standard deviations of F-measure (FM) achieved by the five methods on these four datasets in ten independent runs. In Table I, a cell marked with the $\bullet$, the $\star$, the $*$, and the $\circ$ indicates that the DSUS outperforms this method for the corresponding dataset by 99%, 95%, 90%, and 80% confidence, respectively. FM is

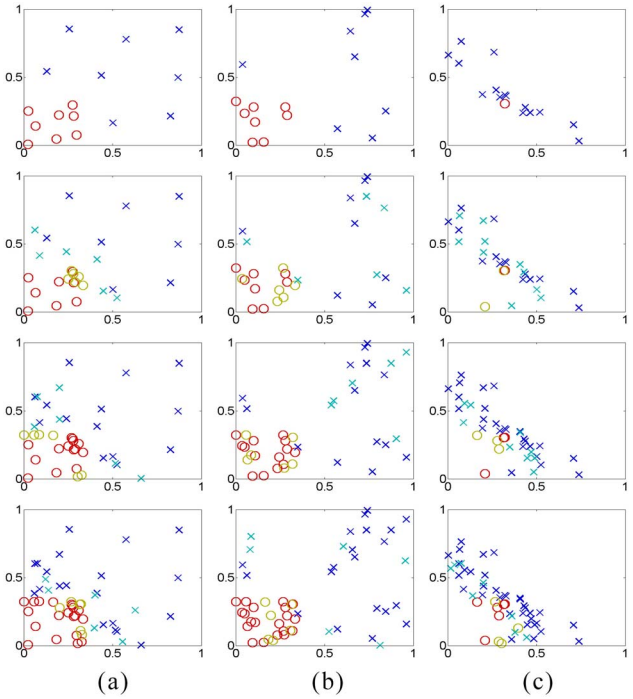

Fig. 3. Samples selected for the uniform dataset. (a) DSUS. (b) RUS. (c) LA-SVM.

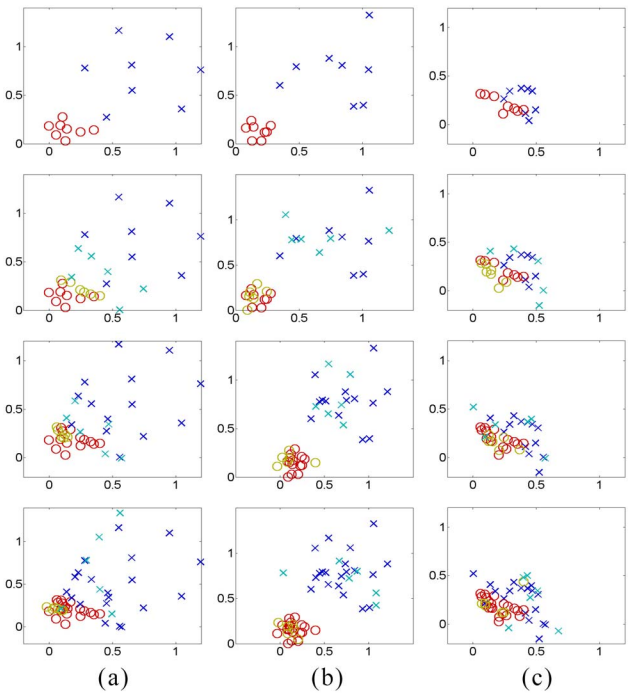

Fig. 4. Samples selected for the Gaussian-1 dataset. (a) DSUS. (b) RUS. (c) LA-SVM.

computed as follows:

$$FM = \frac{2PR}{P+R} \tag{6}$$

where P and R denote the precision rate and the recall rate of RBFNNs on testing sample sets, respectively. When the problem is too simple, e.g., the uniform dataset, the DSUS does not show significant improvement over other methods, except outperforming the RUS significantly. When the dataset becomes more complicated, the DSUS outperforms other methods. For both Gaussian-2 and complex datasets, the DSUS outperforms all other methods with a 99.9% significance. By comparing the results of the DSUS, the SM_{only}, and the $\text{cluster}_{\text{only}}$ in Table I, the combination of clustering and SM sample selection in the DSUS yields a statistically

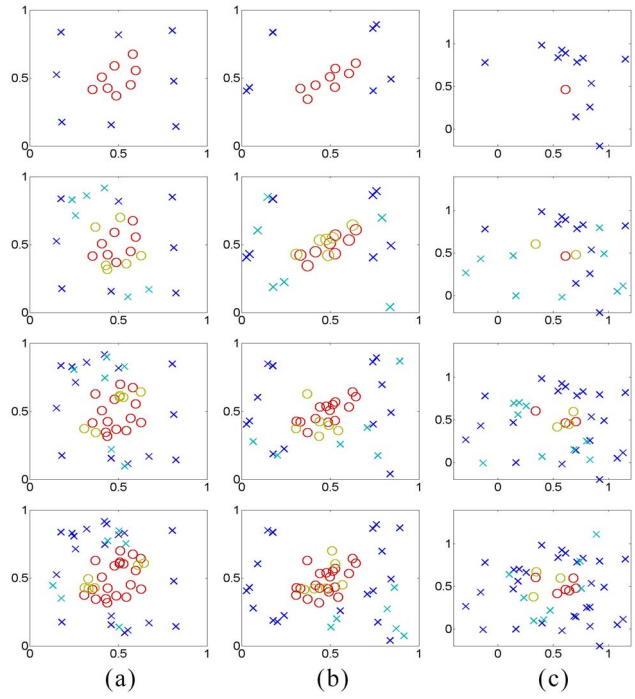

Fig. 5. Samples selected for the Gaussian-2 dataset. (a) DSUS. (b) RUS. (c) LA-SVM.

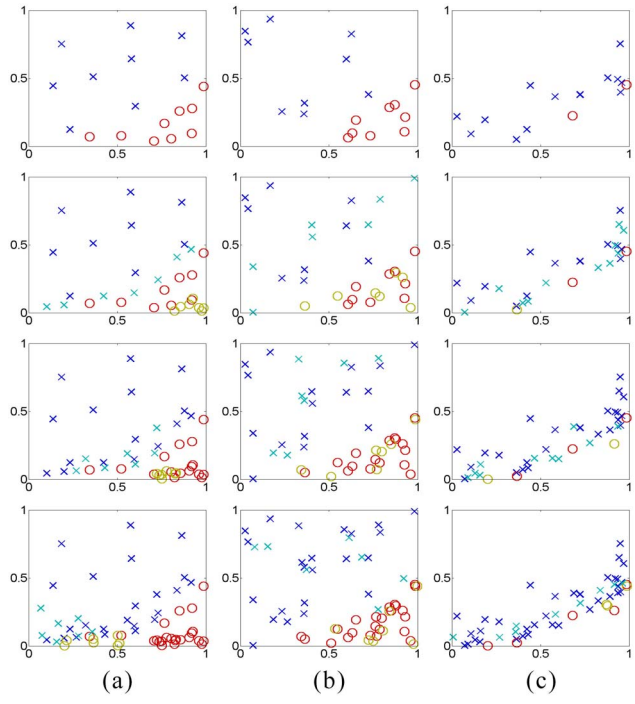

Fig. 6. Samples selected for the complex dataset. (a) DSUS. (b) RUS. (c) LA-SVM.

TABLE I
FM OF RBFNN TRAINED USING DIFFERENT UNDERSAMPLING METHODS FOR THE FOUR ARTIFICIAL DATASETS

Method	Uniform	Gaussian-1	Gaussian-2	Complex
DSUS	0.855±0.023	**0.962±0.021**	**0.812±0.025**	**0.855±0.018**
$DSUS_n$	0.846±0.022	0.961±0.007	0.729±0.047●	0.846±0.007○
$DSUS_h$	0.841±0.024○	0.955±0.014	0.794±0.022∗	0.847±0.022
$DSUS_a$	0.825±0.040⋆	0.959±0.010	0.736±0.063●	0.841±0.020∗
$Cluster_{only}$	**0.876±0.040**	0.905±0.099∗	0.703±0.081●	0.787±0.061●
SM_{only}	0.836±0.083	0.925±0.038●	0.709±0.041●	0.793±0.022●
RUS	0.778±0.556●	0.882±0.044●	0.669±0.054●	0.774±0.031●
LA-SVM	0.847±0.023	0.951±0.000●	0.784±0.013●	0.804±0.010●

significant improvement to the FMs of RBFNNs. The SM_{only} and the LA-SVM yield the second and the third best performance in Gaussian-1, Gaussian-2, and complex datasets. This shows that resampling at the decision boundary is a good method for undersampling. However, they perform worse than the DSUS because they ignore the distribution information of the majority class which may undermine the RBFNN learning about the majority class far away from the decision boundary. This shows the importance of the clustering procedure in the DSUS in providing a diversified undersampling of the majority class.

We also perform experiments on these four datasets using $DSUS_n$, $DSUS_h$, and $DSUS_a$. The $DSUS_n$, $DSUS_h$, and $DSUS_a$ replace the k value by the floor of the square root of the number of samples in the majority class, the floor of the square root of half of the total number of samples in both classes and the floor of the square root of the total number of samples in both classes respectively, i.e., $k = \lfloor\sqrt{N_n}\rfloor$, $k = \lfloor\sqrt{N_p + N_n/2}\rfloor$, and $k = \lfloor\sqrt{N_p + N_n}\rfloor$. These three choices do not provide better results in comparison to the choice of $k = \lfloor\sqrt{N_p}\rfloor$ while the overall differences are statistically insignificant.

The DSUS will not select outliers in the majority class since the clustering is done prior to the sample selection. However, outliers in the minority class may be selected by the DSUS. Therefore, the DSUS may not work well if the outlier issue is a significant one in an imbalance problem.

B. Undersampling for Different Overlapping Ratios

Samples from two Gaussian distributions are sampled as shown in Fig. 7(a). The majority class consists of 900 samples marked by blue crosses and the minority consists of 100 samples marked by red circles in Fig. 7. Fig. 7 shows five scenarios as a result of moving the minority class toward the center of the majority class to create five datasets with different overlapping ratios: (a) 20%, (b) 40%, (c) 60%, (d) 80%, and (e) 100% samples of the minority class. Then, we perform undersampling learning using the DSUS, the RUS, and the LA-SVM for these five datasets.

Table II shows averages and standard deviations of FMs of the three methods with ten independent runs. Resampling at the decision boundary is sensitive to overlapping and the LA-SVM has a statistically significant performance worse than the DSUS. The DSUS also outperforms the RUS with a 99.9% significance for 20%, 40%, 60%, and 80% overlapping. However, when samples in the minority class are 100% overlapping with samples in the majority class,

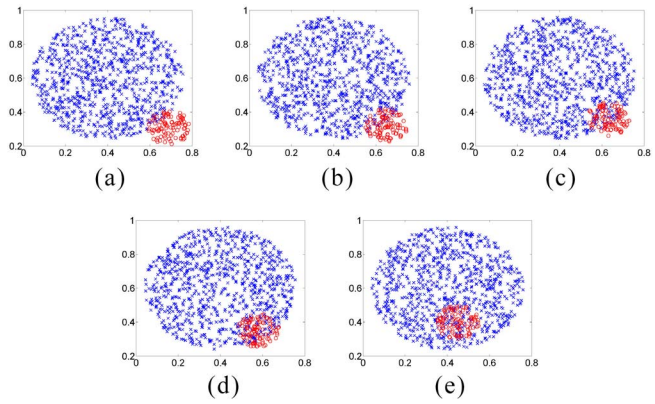

Fig. 7. Artificial overlap datasets. (a) 20% overlap. (b) 40% overlap. (c) 60% overlap. (d) 80% overlap. (e) 100% overlap.

TABLE II
AVERAGE FMS OF DSUS, RUS, AND LA-SVM FOR DIFFERENT OVERLAPPING RATIOS

Overlap	DSUS	RUS	LA-SVM
20%	**0.864±0.030**	0.769±0.026●	0.810±0.019●
40%	**0.794±0.033**	0.739±0.037●	0.713±0.023●
60%	**0.783±0.030**	0.678±0.028●	0.677±0.036●
80%	**0.715±0.032**	0.678±0.028●	0.605±0.029●
100%	0.523±0.012	**0.536±0.026**	0.444±0.024●

random method (RUS) outperforms both the DSUS and the LA-SVM. When the overlapping ratio is at least 40%, LA-SVM performs worse than the RUS which shows that resampling at the decision boundary is sensitive to overlapping between classes. The situation of a 100% overlapping is different from the one in the Gaussian-2 in Fig. 2(c). Samples in the minority class of the Gaussian-2 are located within the majority class without a significant overlapping. So, a clear decision boundary can be found for the Gaussian-2. In contrast, when there is a 100% overlapping between two classes, no method can separate them well and both the DSUS and the LA-SVM can not select any better sample than the RUS.

In summary, the DSUS is able to select useful samples from the two classes to train RBFNN yielding good FMs, i.e., good precision and recall, except the case when there is a very high percentage of (e.g., 100%) overlapping between two classes.

C. Performance for Datasets With Unclear Cluster or Decision Boundary

The DSUS consists of two major components: 1) clustering and 2) the SM. The SM relies heavily on decision boundaries created by the classifier. If the decision boundary between the majority and the minority class is not clear, the SM may be misleading. In this section, we test the DSUS on four different datasets: BC_CW, BC_CNW, BC_CW, and BNC_CNW. The decision boundary between the majority and the minority class of the dataset BC_CW is clear and the clustering works well to find representative centers [Fig. 8(a)]. Fig. 8(b) shows the dataset with a clear decision boundary but the clustering can not find representative centers for data, i.e., BC_CNW. BC_CW refers to the dataset shown in Fig. 8(c), where the decision boundary is not clear between the two classes while the clustering can find representative centers. Fig. 8(d) shows

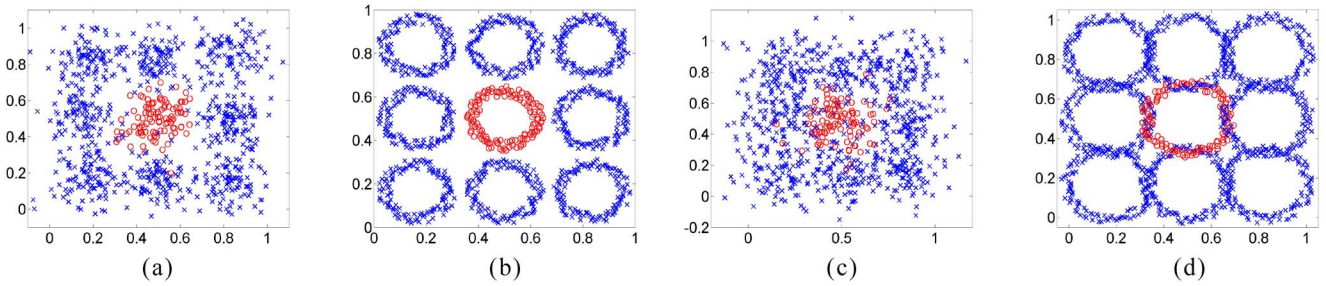

Fig. 8. Artificial datasets. (a) Boundary clear and cluster work. (b) Boundary clear and cluster not work. (c) Boundary not clear and cluster work. (d) Boundary not clear and cluster not work.

TABLE III
AVERAGE FMS OF DSUS, RUS, AND LA-SVM FOR SPECIAL DATASETS WHICH CLUSTERING MAY NOT BE MEANINGFUL

Method	BC_CW	BC_CNW	BC_CW	BNC_CNW
DSUS	**0.812±0.025**	**0.757±0.039**	**0.714±0.038**	**0.674±0.031**
RUS	0.669±0.054•	0.668±0.034•	0.594±0.034•	0.544±0.047•
LA-SVM	0.784±0.013•	0.724±0.015•	0.561±0.030•	0.609±0.012•

the dataset where decision boundary is not clear and the clustering can not find representative centers. The ring-shaped data distribution forces the clustering method to select a sample on the ring closest to the center which is not a representative center and hence a heavy bias.

Table III shows averages and standard deviations of FMs for the three methods under comparisons. Experimental results show that performances of all methods drop significantly when decision boundary is not clear. The failure of the clustering does not affect performances of the RUS and the LA-SVM much because they do not make use of clustering information. The DSUS depends on both clustering and decision boundary for the sensitivity calculation. One could also find that method depending on decision boundary only, i.e., the LA-SVM, performs much worse when decision boundary is not clear. This shows that the combination of clustering and the SM is effective and important to improve imbalance classification performance. An alternative method to further improve the DSUS in dealing with such complex datasets is to project them onto another higher dimensional space using kernel methods to reduce the degree of overlapping.

D. Experiments on 14 UCI Datasets

In this section, we compare the DSUS with representative methods in random-based, resampling at the decision boundary, and ensemble methods. The 14 UCI datasets with different numbers of features, numbers of samples, and imbalance ratios are shown in Table IV, where n, N_p, N_n, and IR denote the number of features, the number of samples in the minority class, the number of samples in the majority class, and the imbalance ratio, respectively. When a dataset consists of more than two classes, one of those classes is selected as the minority class (MC in Table IV) and samples in all other classes are treated as the majority class. The proposed method can be easily extended to multiclass problems. The third column of Table IV also shows the feature types where R, I, and C denote real values, integers, and categorical data (including binary data), respectively.

TABLE IV
STATISTICS OF THE 14 UCI DATASET

Dataset	$N_p + N_n$	n	N_p / N_n	MC	IR
pima	768	8 (6R,2I)	268/500	2	1:1.866
breast_w	699	10 (10I)	241/458	2	1:1.900
waveform	5000	40 (40R)	1653/3347	2	1:2.025
breast	286	9 (9C)	85/201	2	1:2.365
yeast	1484	8 (7R,1I)	429/1055	3	1:2.460
post_op	90	8 (7C,1I)	24/66	2	1:2.750
haberman	306	3 (3I)	81/225	1	1:2.778
cmc	1473	9 (7C,2I)	333/1140	2	1:3.423
hepatitis	155	19 (19C)	32/123	1	1:3.844
ecoli	336	7 (5R,2C)	52/284	3	1:5.462
newthyroid	215	5 (5R)	30/185	3	1:6.167
dermatology	366	34 (33C,1I)	49/317	4	1:6.469
optdigits	5620	64 (64I)	554/5066	1	1:9.144
abalone	4177	8 (1C,7I)	391/3786	7	1:9.683

The proposed DSUS will be compared with random-based (RUS, ROS) resampling methods, ensemble-based undersampling methods (IRUS, EE, BC), and a resampling at the decision boundary method (LA-SVM) introduced in Section II. For the EE, the BC, and the LA-SVM, we follow the original setup in their papers. The RUS, the ROS, the IRUS, and the DSUS use the RBFNN described in Section III-A as classifier while EE and BC use decision tree as described in [17]. Experimental results are evaluated by three metrics: 1) AUC; 2) FM; and 3) G-Mean. The AUC measures the area under the receiver operating characteristic curve. The G-Mean is computed as follows:

$$G - Mean = \sqrt{TPR \times TNR} \tag{7}$$

where TPR and TNR denote the true positive rate and the true negative rate, respectively. For each dataset, 30 independent runs are performed. The averages and the standard deviations of AUCs, FMs, and G-Means are presented in Tables V–VII, respectively. All input features are scaled to $[0, 1]$. For each independent run, 50% of samples are randomly selected as training samples and the other 50% samples left are used as testing samples. The • indicates that the proposed DSUS outperforms the corresponding method in the corresponding dataset with a statistical significance of 99% confidence. In the same way, the $\star$, the $*$, and the $\circ$ marks denote 95%, 90%, and 80% confidences, respectively, that the DSUS outperforms the corresponding method in the corresponding dataset. For AUC, FM, or G-Mean, the method yielding the highest score for a dataset is marked by a bolded font in Table V.

TABLE V
AVERAGE, STANDARD DEVIATION, AND t-TEST OF AUC OF DIFFERENT METHODS ON 14 UCI DATASETS

Dataset	ROS	RUS	IRUS	EE	BC	LA-SVM	DSUS
pima	0.760±0.029•	0.793±0.025•	0.812±0.015•	0.782±0.019•	0.795±0.014•	0.820±0.015○	**0.826±0.015**
breast_w	0.982±0.005•	0.980±0.022•	0.990±0.003•	0.985±0.006•	0.986±0.005•	**0.994±0.002**	0.992±0.003
waveform	0.959±0.003•	0.946±0.004•	0.955±0.003•	0.944±0.004•	0.951±0.004•	0.962±0.003•	**0.965±0.003**
breast	0.645±0.050○	0.645±0.057○	0.658±0.050	0.643±0.038∗	0.638±0.039⋆	0.640±0.028⋆	**0.663±0.047**
yeast	0.743±0.017•	0.735±0.022•	0.777±0.014	0.758±0.016•	0.762±0.018⋆	**0.785±0.012**	0.780±0.015
post_op	0.416±0.074∗	0.442±0.077	0.448±0.058	0.389±0.079•	0.371±0.072•	0.398±0.067•	**0.451±0.069**
haberman	0.640±0.051⋆	0.628±0.037•	**0.685±0.043**	0.659±0.045	0.641±0.043⋆	0.662±0.028	0.668±0.049
cmc	0.689±0.018•	0.679±0.029•	0.720±0.018	0.701±0.014•	0.679±0.017•	0.674±0.020•	**0.721±0.020**
hepatitis	0.809±0.046•	0.842±0.045	**0.847±0.039**	**0.847±0.038**	0.835±0.037	0.800±0.050•	0.842±0.047
ecoli	0.937±0.032○	0.943±0.029	0.926±0.031•	0.939±0.027○	0.939±0.032○	0.948±0.031	**0.949±0.027**
newthyroid	0.977±0.034•	0.990±0.011•	0.997±0.003	0.983±0.016•	0.976±0.022•	0.997±0.005	**0.998±0.005**
dermatology	0.986±0.011•	0.988±0.007•	0.985±0.005•	0.961±0.127•	0.987±0.013∗	**0.996±0.002**	0.992±0.006
optdigits	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	0.967±0.178	0.968±0.175	**1.000±0.000**	**1.000±0.000**
abalone	0.849±0.010∗	0.850±0.010○	**0.854±0.008**	0.835±0.009•	0.814±0.009•	0.848±0.007•	**0.854±0.010**

TABLE VI
AVERAGE, STANDARD DEVIATION, AND t-TEST OF FM OF DIFFERENT METHODS ON 14 UCI DATASETS

Dataset	ROS	RUS	IRUS	EE	BC	LA-SVM	DSUS
pima	0.582±0.036•	0.637±0.030•	0.625±0.027•	0.636±0.019•	0.638±0.022•	0.638±0.013•	**0.664±0.019**
breast_w	0.929±0.017•	0.918±0.022•	0.900±0.020•	0.941±0.012•	0.946±0.011•	0.952±0.010○	**0.956±0.009**
waveform	0.847±0.007•	0.791±0.008•	0.750±0.014•	0.817±0.008•	0.827±0.008•	0.848±0.005•	**0.858±0.008**
breast	0.449±0.059•	0.469±0.063	0.469±0.072	0.477±0.044	0.451±0.046•	0.479±0.028	**0.487±0.050**
yeast	0.531±0.023•	0.525±0.029•	0.563±0.029•	0.566±0.021•	0.572±0.018○	0.569±0.008•	**0.578±0.016**
post_op	0.230±0.080•	0.302±0.108⋆	0.312±0.080∗	0.275±0.082•	0.253±0.083•	0.240±0.089•	**0.349±0.073**
haberman	0.418±0.066•	0.420±0.042•	**0.474±0.063**	0.456±0.044	0.441±0.041○	0.423±0.060•	0.461±0.062
cmc	0.445±0.023•	0.429±0.029•	0.467±0.024	0.455±0.017•	0.432±0.021•	0.410±0.021•	**0.472±0.017**
hepatitis	0.548±0.076•	0.570±0.073○	0.586±0.067	0.566±0.061∗	0.565±0.057⋆	0.518±0.070•	**0.600±0.076**
ecoli	0.760±0.080⋆	0.726±0.061•	0.746±0.053•	0.735±0.067•	0.739±0.064•	0.764±0.042•	**0.800±0.061**
newthyroid	0.842±0.067•	0.859±0.063•	0.879±0.062•	0.812±0.076•	0.806±0.082•	0.899±0.056⋆	**0.929±0.054**
dermatology	0.747±0.092•	0.763±0.077•	0.806±0.038•	0.752±0.067•	0.837±0.071	0.855±0.021	**0.857±0.062**
optdigits	0.989±0.005∗	0.944±0.017•	0.920±0.018•	0.982±0.012•	0.987±0.011∗	0.990±0.003	**0.991±0.004**
abalone	0.381±0.010	0.364±0.013•	0.407±0.010	0.372±0.009•	0.365±0.014•	0.377±0.006•	**0.382±0.009**

TABLE VII
AVERAGE, STANDARD DEVIATION, AND t-TEST OF G-MEAN OF DIFFERENT METHODS ON 14 UCI DATASETS

Dataset	ROS	RUS	IRUS	EE	BC	LA-SVM	DSUS
pima	0.672±0.029•	0.716±0.025•	0.703±0.023•	0.714±0.016•	0.717±0.018•	0.655±0.027•	**0.739±0.016**
breast_w	0.946±0.017•	0.935±0.020•	0.912±0.018•	0.960±0.010•	0.964±0.009•	0.970±0.009○	**0.973±0.007**
waveform	0.893±0.006•	0.860±0.007•	0.791±0.012•	0.876±0.005•	0.883±0.006•	**0.905±0.003**	0.902±0.006
breast	0.586±0.040•	0.598±0.054○	0.603±0.062	0.604±0.035	0.584±0.037•	0.479±0.028•	**0.615±0.043**
yeast	0.660±0.020•	0.654±0.024•	0.686±0.025•	0.692±0.018○	0.697±0.016	0.663±0.012•	**0.699±0.016**
post_op	0.393±0.086•	0.444±0.102	0.427±0.066○	0.402±0.085•	0.396±0.083•	0.390±0.089•	**0.456±0.079**
haberman	0.578±0.058•	0.581±0.038•	0.613±0.058	0.610±0.038	0.600±0.037	0.575±0.065•	**0.615±0.056**
cmc	0.634±0.022•	0.618±0.027•	0.655±0.022○	0.648±0.016•	0.627±0.019•	0.598±0.019•	**0.663±0.016**
hepatitis	0.716±0.054•	0.755±0.058	**0.763±0.049**	0.757±0.051	0.750±0.049	0.691±0.058•	0.751±0.062
ecoli	0.903±0.024	0.895±0.028	0.883±0.047	0.885±0.042	0.889±0.037	**0.912±0.03**	0.895±0.049
newthyroid	0.926±0.058•	0.931±0.047•	0.886±0.055•	0.942±0.029•	0.927±0.048•	0.904±0.049•	**0.971±0.037**
dermatology	0.933±0.029	0.934±0.022	0.926±0.018∗	0.928±0.036	0.937±0.043	**0.955±0.021**	0.938±0.035
optdigits	0.996±0.002	0.991±0.003•	0.923±0.016•	0.993±0.004•	0.993±0.004•	0.992±0.002•	**0.996±0.002**
abalone	0.782±0.014∗	0.766±0.013•	**0.790±0.014**	0.779±0.010•	0.745±0.014•	0.781±0.007•	0.788±0.011

As shown in Tables V–VII, the DSUS outperforms other methods with a 99% statistical significance in 153 out of 252 (60.71%) experiments of different datasets and resampling methods. For six out of 14 UCI datasets, pima, breast, post_op, cmc, newthyroid and optdigits, the DSUS yields the best results in term of all AUC, FM, and G-Mean. For the FM, the DSUS performs the best except for the Haberman dataset. These show that the DSUS performs well for different feature types, imbalance ratios, numbers of feature, and numbers of sample. The LA-SVM outperforms the DSUS without a statistical significance in some cases while the DSUS outperforms the LA-SVM significantly in most cases. The newest IRUS uses ensemble classifiers with different sets of RUS yields better results than the DSUS in some cases, but none of them has any statistical significance. Ensemble methods (EE, BC, and IRUS) perform worse than the DSUS while they require more computational complexity for building ensemble of classifiers. This shows that the DSUS is more efficient and effective in selecting useful samples from both the majority and minority classes.

E. Cases That the DSUS Works and Does Not Work

In our experiments, the DSUS performs well in most cases in comparison to random-based, ensemble-based, and decision boundary-based undersampling methods. However, the DSUS performs worse than RUS when there is a significant

overlapping (e.g., 100%) between the majority and minority classes. Moreover, when the decision boundary between two classes is not clear, the SM can not allocate informative samples easily because the classifier will be sensitive owing to complex decision boundary. On the other hand, the failure of finding representative samples for clusters using clustering method degrades the performance of the DSUS. Finally, if the minority class suffers from a serious outlier problem, the DSUS may select outliers as training samples which may degrade the performance of the DSUS.

V. Conclusion

In this paper, we proposed the DSUS which preserves the distribution information of the majority class and select informative samples from both classes. Experimental results show that the DSUS is robust to different imbalance ratios and up to 80% overlapping. One future work is to incorporate both undersampling and oversampling with the SM to create ensemble of classifiers for imbalance problems. In this paper, we treat all feature types the same after scaling them into [0, 1] during the calculation of the SM for sample selection. A better sample selection result may be expected by proposing a particular SM calculation method for different feature types. Moreover, when data changes across time, incremental learning is needed. Very few works have been done in dealing with imbalance problems in incremental learning which is an important issue for many web-based real-world applications.

References

[1] J. C. Fernandez Caballero, F. J. Martínez, C. Hervás, and P. A. Gutiérrez, "Sensitivity versus accuracy in multiclass problems using memetic Pareto evolutionary neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 750–770, May 2010.

[2] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.

[3] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[4] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[5] V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6585–6608, 2012.

[6] M. Gao, X. Hong, S. Chen, and C. J. Harris, "A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems," *Neurocomputing*, vol. 74, no. 17, pp. 3456–3466, 2011.

[7] V. García, J. S. Sánchez, and R. A. Mollineda, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowl.-Based Syst.*, vol. 25, no. 1, pp. 13–21, 2012.

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

[9] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*. Berlin, Germany: Springer, 2005, pp. 878–887.

[10] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Advances in Knowledge Discovery and Data Mining*. Berlin, Germany: Springer, 2009, pp. 475–482.

[11] T. Maciejewski and J. Stefanowski, "Local neighbourhood extension of SMOTE for mining imbalanced data," in *Proc. 2011 IEEE Symp. Comput. Intell. Data Min. (CIDM)*, Paris, France, pp. 104–111.

[12] 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," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, 2012.

[13] A. Fernández, M. J. Del Jesus, and F. Herrera, "Multi-class imbalanced data-sets with linguistic fuzzy rule based classification systems based on pairwise learning," in *Computational Intelligence for Knowledge-Based Systems Design*. Berlin, Germany: Springer, 2010, pp. 89–98.

[14] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognit.*, vol. 45, no. 10, pp. 3738–3750, 2012.

[15] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[16] S. Hu, Y. Liang, L. Ma, and Y. He, "MSMOTE: Improving classification performance when training data is imbalanced," in *Proc. 2009 2nd Int. Workshop IEEE Comput. Sci. Eng. (WCSE)*, vol. 2. Qingdao, China, 2009, pp. 13–17.

[17] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[18] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Comparing boosting and bagging techniques with noisy and imbalanced data," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 3, pp. 552–568, May 2011.

[19] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 281–288, Feb. 2009.

[20] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, Sep. 2005.

[21] H.-N. Qu, G.-Z. Li, and W.-S. Xu, "An asymmetric classifier based on partial least squares," *Pattern Recognit.*, vol. 43, no. 10, pp. 3448–3457, 2010.

[22] C.-Y. Yang, J.-S. Yang, and J.-J. Wang, "Margin calibration in SVM class-imbalanced learning," *Neurocomputing*, vol. 73, no. 1, pp. 397–411, 2009.

[23] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, Jan. 2006.

[24] S. S. Haykin, *Neural Networks and Learning Machines*, vol. 3. New York, NY, USA: Prentice Hall, 2009.

[25] D. S. Yeung, W. W. Y. Ng, D. Wang, E. C. Tsang, and X.-Z. Wang, "Localized generalization error model and its application to architecture selection for radial basis function neural network," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1294–1305, Sep. 2007.

[26] B. Sun, W. W. Y. Ng, D. S. Yeung, and P. P. K. Chan, "Hyper-parameter selection for sparse LS-SVM via minimization of its localized generalization error," *Int. J. Wavelets Multiresolut. Inf. Process.*, vol. 11, no. 3, 2013, Art. ID 1350030.

[27] D. S. Yeung, J. Li, W. W. Y. Ng, and P. P. K. Chan, "MLPNN training via a multiobjective optimization of training error and stochastic sensitivity," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

Wing W. Y. Ng (S'01–M'06) received the B.Sc. and Ph.D. degrees from Hong Kong Polytechnic University, Hong Kong, in 2001 and 2006, respectively.

He joined the Department of Computer Science and Technology, Shenzhen Graduate School, Harbin Institute of Technology, Harbin, China, as an Assistant Professor from 2006 to 2008. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include large-scale image retrieval, machine learning in big data and nonstationary environments, and business intelligence.

Dr. Ng is currently an Associate Editor of the *International Journal of Machine Learning and Cybernetics*. He is the Principle Investigator of two China National Nature Science Foundation projects and the Program for New Century Excellent Talents in University from the China Ministry of Education. He served as a Board of Governor of the IEEE Systems, Man and Cybernetics Society from 2011 to 2013.

Junjie Hu (S'13) received the B.Eng. degree in computer science and technology from South China University of Technology, Guangzhou, China. He is currently pursuing the postgraduate degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

His current research interests include scalable machine learning for big data analytics, especially online learning with imbalanced streaming data, kernel classifier, online regret analysis, online AUC optimization, top-K ranking, social recommender systems, similarity measurement via kernel method, machine learning with imbalanced data, active learning, RBFNN, L-GEM, and clustering. He has published several academic papers in conferences, including AAAI, International Conference on Machine Learning and Cybernetics (ICMLC), International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), and NIPS Big Learning Workshop. He has also served as a Student Assistant in 2012 ICMLC, 2013 ICWAPR, and 2013 ACM RecSys.

Mr. Hu was the recipient of the National Scholarship twice during 2010 and 2012 from the Ministry of Education of China, the 2013 IBM Outstanding Student Scholarship from the IBM, and the 2013 Outstanding Undergraduate Student Award from the China Computer Federation.

Daniel S. Yeung (M'89–SM'99–F'04) is currently a Chair Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He has been a faculty member of Hong Kong Polytechnic University, Hong Kong and Rochester Institute of Technology, Rochester, NY, USA. He has also worked for TRW Inc., General Electric Corporation R&D Centre and Computer Consoles Inc. in USA.

Prof. Yeung served as the President of the IEEE Systems, Man and Cybernetics Society from 2008 to 2009. He is a fellow of the IEEE.

Shaohua Yin (S'13) received the B.S. degree in digital media technology from Jiangnan University, Wuxi, China, in 2011. He is currently pursuing the postgraduate degree from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

His current research interests include learning from imbalanced samples, content-based information retrieval, and metric learning.

Fabio Roli (F'12) received the M.S. (Hons.) and Ph.D. degrees in electronic engineering from the University of Genoa, Genoa, Italy.

From 1988 to 1994, he was at the University of Genoa as a Research Group Member in image processing and understanding. He was an Adjunct Professor at the University of Trento, Trento, Italy, from 1993 to 1994. In 1995, he joined the Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy, where he is currently a Professor of Computer Engineering and a Director of the Research Laboratory on Pattern Recognition and Applications. His current research interests include the design of pattern recognition systems and their applications to biometric personal identification, multimedia text categorization, and computer security.

Prof. Roli is a Governing Board Member of the International Association for Pattern Recognition and the IEEE Systems, Man and Cybernetics Society. He is a fellow of the International Association for Pattern Recognition. He is a fellow of the IEEE, and a fellow of the International Association for Pattern Recognition.