Pattern Recognition 71 (2017) 306–319

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

GIR-based ensemble sampling approaches for imbalanced learning

Bo Tang[a], Haibo He[b,*]

[a] Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, 39762, USA
[b] Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, 02881, USA

ARTICLE INFO

Article history:
Received 13 September 2016
Revised 1 May 2017
Accepted 11 June 2017
Available online 13 June 2017

Keywords:
Imbalanced learning
Generalized imbalance ratio
Undersampling and oversampling
Adaptive learning
Boosting and bagging

ABSTRACT

This paper presents two adaptive ensemble sampling approaches for imbalanced learning: one is the undersampling-based approach, and the other one is the oversampling-based approach, with the objectives of bias reduction and adaptive learning. Both of these two approaches are based on a novel class imbalance metric, termed generalized imbalance ratio (GIR), instead of the conventional sample size ratio. Specifically, these two sampling-based approaches adaptively split the imbalanced learning problem into multiple balanced learning subproblems in a probabilistic way, which forces the classifiers trained in the subproblems focus on those difficult to learn samples. In each subproblem, several weak classifiers are trained in a boosting manner. A final stronger classifier is further built by combining all these weak classifiers in a bagging manner. Extensive experiments are conducted on real-life UCI imbalanced data sets to evaluate the performance of the proposed methods. The superior performance demonstrates the effectiveness of the proposed methods and indicates wide potential applications in data mining.

1. Introduction

In recent years, imbalanced learning has drawn a significant interest in numerous scientific and engineering areas [1], such as marketing, data mining, government funding agencies, and biomedical science. The "imbalance" indicates that the class distribution is uneven for a given imbalanced data set. Most standard classification algorithms that have been proposed for balanced data sets do not consider the issues existed in imbalanced data sets, and they always fail to properly represent the class distribution, leading to a poor learning performance.

In an imbalanced data set, one class data (majority) usually dominates the other one (minority). It has been known that the classifiers trained by traditional learning algorithms would be very weak for the minority against the majority [1]. However, in practice, the misclassification of the minority is usually more costly than that of the majority. One well-known example is the cancerous patient predication problem, in which the number of "Healthy" (negative) data samples is far more than the number of "Cancerous" (positive) data samples, such as the real-life MAMMOGRAPHY data set [2] in which there are 10,923 Healthy samples and only 260 Cancerous data samples. Of course, we wish the rate of misclassifying the cancerous patient to be healthy is as low as possible, because it is always much more costly than classifying a healthy patient to be cancerous in medical industry. Hence, for many imbalanced learning problems, we are interested in enhancing the discrimination of minority with the minimum cost of misclassification of majority.

The fundamental difficulty of a traditional learning algorithm for imbalanced data sets is that it hardly represents the distribution of the minority because of the domination of the majority. For an extremely imbalanced data set, the traditional classifiers may provide imbalanced classification performance, with close to a 100 percent accuracy for the majority and with a near 0 percent accuracy for the minority. To solve this deficiency, many complementary techniques or modifications of the original learning algorithms have been proposed in last decades. Among them, the sampling (oversampling and undersampling) approaches, such as synthetic minority over-sampling technique (SMOTE) [3,4], adaptive synthetic sampling approaches (ADASYN) [5], and ensemble-based undersampling approaches [6], are of great interest because of their effectiveness, simplicity and easy-incorporation into the existing learning algorithms.

One of critical issues in these sampling-based approaches is how to choose the best balance degree in the newly formed data set. Most of existing approaches use sample size ratio to measure class imbalance between the majority and the minority. Typically, a fully balanced data set in size (i.e., both majority and minority have the same sample size) is formed. However, it is possible that the class imbalance still exists even for a data set whose sample size is balanced. To address this issue, we introduce a new mea-

[*] Corresponding author.
E-mail addresses: he@ele.uri.edu, hhbhust@gmail.com (H. He).

http://dx.doi.org/10.1016/j.patcog.2017.06.019
0031-3203/

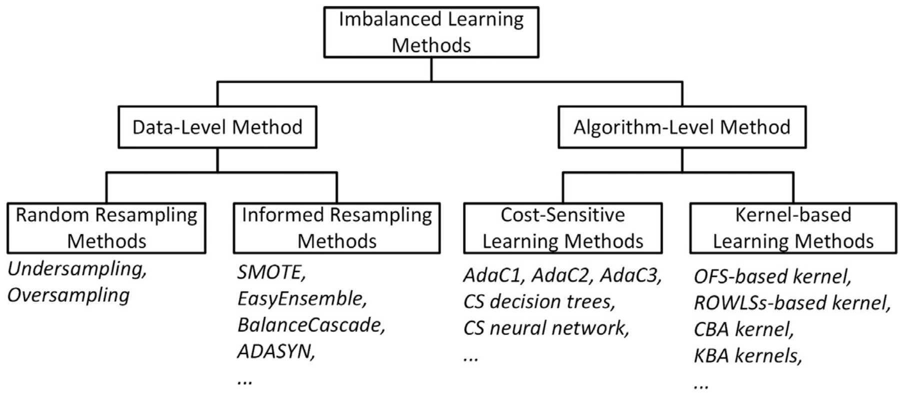
Imbalanced Learning Methods
Data-Level Method
Algorithm-Level Method
Random Resampling Methods
Informed Resampling Methods
Cost-Sensitive Learning Methods
Kernel-based Learning Methods
Undersampling, Oversampling
SMOTE, EasyEnsemble, BalanceCascade, ADASYN, …
AdaC1, AdaC2, AdaC3, CS decision trees, CS neural network, …
OFS-based kernel, ROWLSs-based kernel, CBA kernel, KBA kernels, …

Fig. 1. The summary of imbalanced learning methods.

surement of class imbalance, termed generalized imbalance ratio (GIR), based on the intra-class coherence metric which is a global class-wise statistic of class distribution. We derive the asymptotic properties of the GIR measure when the ratio of the sample sizes goes to a finite limit. The theoretic properties illustrate that the GIR measure could be equivalent to the sample size ratio under certain conditions. Another important issue in sampling-based approaches is that the classifier trained in the new balanced data set is usually weak and biased, since some useful majority samples may be discarded in undersampling-based approaches and some irrelevant minority samples may be generated in oversampling-based approaches. Motivated by the success of ensemble learning approaches, we adaptively split the imbalanced learning problem into several balanced subproblems using either undersampling or oversampling schemes, where a density distribution that is based on point-wise statistics is used for adaptive sampling to force the classifiers in each subproblem focus on those samples that are difficult to learn. Lastly, a stronger classifier is built for final classification. We name it GIR-based ensemble sampling approach, and develop two specific sampling approaches: GIR-basd ensemble undersampling (GIREnUS) approach and GIR-based ensemble oversampling (GIREnOS) approach. We conduct extensive experiments on several UCI data sets to evaluate the performance of these two GIR-based approaches. The superior performance demonstrates their effectiveness and indicates wide applications on data mining. In summary, our contribution in this paper is threefold:

- We propose a novel metric, named generalized imbalance ratio, which measures class distribution imbalance between two classes, based on their intra-class coherence. Theoretical properties of the proposed distribution imbalance metric are also studied.
- We propose to adaptively split the imbalanced learning problem into multiple balanced learning subproblems in a probabilistic way, which forces the classifiers trained in the subproblems focus on those difficult to learn samples.
- We develop two adaptive sampling-based (oversampling and undersampling) approaches based on the proposed GIR measure, named *GIREnUS* and *GIREnOS*, for imbalance learning.

The rest of this paper is organized as follows: In Section 2, we review related work on imbalanced learning. In Section 3, we give the motivation of our new generalized imbalance ratio with global class-wise statistics to measure the class imbalance for a given data set. In Section 4, we develop GIR-based ensemble undersampling and oversampling approaches for imbalanced learning. In Section 5, we present our experimental settings, results and analysis. Finally, we give our discussions and conclusions in Section 6.

2. Related work

The over-fitting for one class and the under-fitting for the other class are two major issues in imbalanced learning when traditional classification algorithms are applied to imbalanced data sets, leading to classification performance degradation. A number of methods have been proposed to address these two issues, which can be categorized into two major groups: the data-level method and the algorithm-level method. While the data-level method aims to balance the original data set before training the classifiers, the algorithm-level method attempts to modify or extend existing classification algorithms to be suitable for imbalanced learning. We summarize most of existing imbalanced learning algorithms in Fig. 1, and describe each type of method in the rest of this section.

2.1. Data-level methods

Undersampling (removing a subset of majority samples) and oversampling (creating a subset of synthetic minority samples) are two commonly used data-level methods. Previous studies have shown that randomly removing some majority samples, termed *random undersampling*, or randomly repeating some minority samples, termed *random oversampling*, can significantly improve the performance of imbalanced learning [7–9]. Several more complex sampling approaches, named *informed undersampling* or *informed oversampling* [1], have been proposed to avoid information loss in random sampling approaches and further to enhance the learning of classifiers on imbalanced data sets. For example, the most popular oversampling approaches, SMOTE [3] and its variations (e.g., borderline-SMOTE [4] and modified synthetic minority oversampling technique (MSMOTE) [10]), generate artificial minority samples based on the nearest neighbors in feature space, which have shown that the generated synthetic samples are more informative than the repetitive samples in random oversampling method. ADASYN [5] and its extension KernelADASYN [11] weight the minority samples with their local distributions, which is able to shift decision boundaries toward difficult samples and to reduce the bias of classifiers. JOUS-Boost [12,13] combines jittering of data with oversampling or undersampling schemes to reduce misclassification costs. An extreme data imbalance case, i.e., infinitely imbalanced case, is considered for logistic regression [14], in which one class has a finite sample size and the sample size of the other class grows without bound.

Among those informed sampling approaches, the idea of ensemble (bagging [15] and boosting [16]) is incorporated into the sampling approaches and gains a great success in imbalanced learning. In boosting-based ensemble methods, the AdaBoost procedure [17] is widely applied. For example, the AdaBoost algorithm is combined with SMOTE and MSMOTE, leading to SMOTEBoost [18] and MSMOTEBoost [10], respectively. Compared to the boosting-based ensemble methods, the bagging-based ensemble methods are much more simple, since a bagging algorithm does not require to modify the sampling distribution at each classifier's training. The majority voting rule, a commonly used bagging algorithm [19], can be combined with most sampling approaches, leading to simple but effective imbalanced learning methods, such as UnderBagg (random undersampling + bagging) [20] and OverBagg (random oversampling + bagging) [21]. It has also been shown that both boosting and bagging can be applied together for imbalanced learning. For example, the methods of EasyEnsemble and Balance-Cascade [6] use bagging as the ensemble method and train each classifier with boosting.

2.2. Algorithm-level methods

The cost-sensitive learning approach which specifies the costs for the misclassification of a particular class sample [22,23] can be naturally applied to imbalanced learning. A general cost-sensitive learning framework, named Metacost, is proposed in [24]. Motivated by the success of the boosting algorithm, several cost-sensitive learning methods with the AdaBoost have been proposed, such as AdaC1, AdaC2, and AdaC3 [25]. The cost-sensitive learning strategy can be also incorporated into most existing classification algorithms to improve their learning capacity for imbalanced learning, such as cost-sensitive neural network [26], cost-sensitive decision tree [22,27], cost-sensitive Bayesian classifiers [28,29], etc. The Kernel-based approach is the other algorithm-level method, which modifies the kernels to enhance the learning for the minority samples. Several of these approaches are based on support vector machine (SVM), incorporating various sampling and boosting methods [30–33]. For example, In [30], Wu and Chang propose a kernel-boundary alignment (KBA) algorithm in which the kernel matrix is modified considering the imbalanced data distribution. Because of the limited space of paper, we refer interested readers to the comprehensive survey paper [1] for more details.

For both data-level and algorithm-level imbalance learning methods, many data complexity metrics [34] have also been applied to imbalance learning problems [35–37,37,38]. For example, in [35], three types of data complexity measures, including overlaps in feature values from different classes, separability of classes, and geometry, topology and density of manifolds, have been integrated into both SMOTE-based oversampling and evolutionary undersampling approaches. In [37], both diversity and separable metrics are considered for oversampling in diversity and separable metrics in oversampling technique (DSMOTE). In algorithm-level approaches, the classification error information, as a data complexity measure, can be used to adjust the costs of each class in cost-sensitive learning algorithms, as it can measure the difficulty of each class for discrimination [38].

3. GIR: generalized imbalance ratio

3.1. Motivation

For a given imbalanced data set, it is necessary to first measure the imbalance degree between the majority and the minority. Most of previous work in imbalanced learning use the sample size ratio as the measurement of class imbalance. The sample size ratio is actually the measurement of sample size imbalance, which may not represent the class imbalance on data distribution. The previous empirical study [39] has shown that the testing accuracy for the minority class increases with more representative minority samples when the sample size ratio is fixed, indicating that the minority class distribution could be well learned regardless of the uneven sample size ratio.

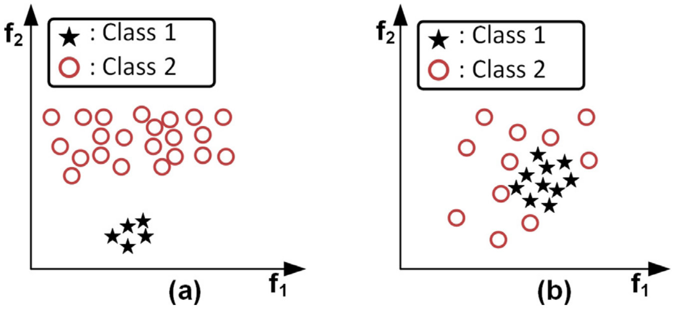

Fig. 2. Illustration of drawbacks of using sample size ratio as class imbalance measurement: (a). class-balanced data set with an imbalanced sample size ratio; (b). class-imbalanced data set with a balanced sample size ratio.

We illustrate the issue of using sample size ratio as class imbalance measurement with two examples in Fig. 2. In the first example shown in Fig. 2(a), the data set has an imbalanced sample size, where the sample size of class 1 (minority) is less than that of class 2 (majority) and the sample size ratio between the majority and the minority is $20:5=4$. However, for this case, it is shown that the classification boundary would be strong enough for these two classes and any simple classifier could discriminate them, since there is no overlapping between two classes. In other words, the distributions of both majority and minority are well represented by their samples, and it is better to consider such data set as a class balanced data set. In the second example shown in Fig. 2(b), two classes have the same sample size (balanced sample size ratio), but class 1 data samples are more concentrated than class 2 data samples. There would be less "support" class 2 data samples along the classification boundary, which may result in a poor classification performance for class 2.

Therefore, it is inappropriate to use sample size ratio to measure the class imbalance for a given data set. For a data set with an imbalanced sample size (i.e., a large sample size ratio), class imbalance may not exist, such as the example in Fig. 2(a). For a data set with a balanced sample size ratio (i.e., each class has the same sample size), class imbalance may still exist, and the existing imbalanced learning methods can be still applied to improve the classifier's learning capacity for the class that is under-represented and has weak decision boundaries. For example, for the data set shown in Fig. 2(b), if the misclassification of class 2 data is more expensive than that of class 1 data (i.e., cost-sensitive issue), generating synthetic class 2 data along the classification boundary could help to strengthen a classifier's discrimination for class 2 against class 1. For this reason, instead of using the term of "majority" and the term of "minority" in conventional imbalanced learning approaches, we use the term of "positive" to denote the class that has a dominated distribution, and "negative" to denote the other class in our paper. We next introduce a metric of intra-class coherence to evaluate how well the class is represented by the data, and propose a new class imbalance measurement as the replacement of the sample size ratio for imbalanced learning.

3.2. GIR definition

Given the training data set $\mathcal{X}=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_N,y_N)\}$, where $\mathbf{x}_i\in\mathbb{R}^d$ is the d-dimensional data sample and $y_i\in\{+1,-1\}$ is the corresponding class label denoting the positive and negative class, respectively, we denote the set of positive samples by $\mathcal{P}$ with the size of N_+ and the set of negative samples by $\mathcal{N}$ with the size of N_-, and we have $N_+ + N_- = N$. In conventional imbalanced learning methods, the number of negative samples is usually much more than that of positive samples in imbalanced data sets, i.e., $N_- \gg N_+$. This, however, may be not true with our new class imbalance measurement. The number of negative samples could be equal to or less than that of positive samples as long as the negative class has a dominated distribution over the positive class, which is indicated by the metric of intra-class coherence.

For each class, we measure the *intra-class coherence* with a generalized class-wise statistic, which is originally proposed in extended nearest neighbors (ENN) method [40], based on k-nearest neighbors. The generalized class-wise statistic for the majority class is given by:

$$\begin{aligned} T_+ &= \frac{1}{N_+}\sum_{\mathbf{x}\in\mathcal{P}}\frac{1}{k}\sum_{r=1}^{k} I_r(\mathbf{x},\mathcal{X}) \\ &= \frac{1}{N_+}\sum_{\mathbf{x}\in\mathcal{P}} t_k(\mathbf{x}) \end{aligned} \quad (1)$$

where k is the total number of nearest neighbors to be considered, and $I_r(\mathbf{x},\mathcal{X})$ is the indicator function indicating whether the data sample $\mathbf{x}$ and its rth nearest neighbor in $\mathcal{X}$, denoted by $NN_r(\mathbf{x},\mathcal{X})$, are from the same class or not, i.e., $I_r(\mathbf{x},\mathcal{X}) = 1$ if both $\mathbf{x}$ and $NN_r(\mathbf{x},\mathcal{X})$ belong the same class, otherwise $I_r(\mathbf{x},\mathcal{X}) = 0$. Note that $t_k(\mathbf{x})$ is a point-wise statistic for the sample $\mathbf{x}$ which evaluates how many samples in its k nearest neighbors come from its own class. Similarly, the generalized class-wise statistic for the minority class is given by:

$$T_- = \frac{1}{N_-}\sum_{\mathbf{x}\in\mathcal{N}} t_k(\mathbf{x}) \quad (2)$$

It can be easily shown that $T_+ \in [0,\ 1]$ and $T_- \in [0,\ 1]$. The two generalized class-wise statistics T_+ and T_- measure the coherence of the positive class and the negative class, respectively, and thus the generalized class-wise statistic is also called the metric of intra-class coherence, indicating the degree to which the nearest neighbors of samples in one class are dominated by the samples from the other class. For example, a large value of T_+ indicates that the positive samples are much more concentrated and their nearest neighbors are dominated by the positive samples, whereas a small value of T_+ means that the neighbors of positive samples are dominated by the negative samples. Especially, when $T_+ = 1$, all nearest neighbors of positive samples are positive, and when $T_+ = 0$, all nearest neighbors of positive samples are negative. This is also true for T_- with negative samples. A formal mathematical definition and discussion of the generalized class-wise statistic can be seen in [40].

For a given imbalanced data set, unlike most existing imbalance learning algorithms which consider the positive class has less data samples than the negative class, we consider that the class with smaller intra-class coherence is positive class and the class with larger intra-class coherence is negative class. In other words, we always have $T_+ \le T_-$. To measure the class imbalance between the positive class and the negative class, we define a new measurement, *generalized imbalance ratio* (GIR) denoted by ΔT, as follows:

$$\Delta T = T_- - T_+ \quad (3)$$

which is the difference between T_- and T_+, and we always have $0 \le \Delta T \le 1$. The new class imbalance measurement offers many advantages: first, it is independent to the class sample size, which is able to avoid the issue of the ignorance of representative minority samples in traditional definition of sample size ratio; second, the GIR measure is bounded between 0 and 1, while the sample size ratio is not upper bounded; last but not least, the difference between T_- and T_+ can somehow reflect a classifier's capacity to discriminate negative and positive samples from their nature distributions.

3.3. Theoretical properties

The proposed GIR measure is not only quite conceptually simple but also possesses nice theoretical properties. In this subsection, we discuss these theoretical properties of the proposed GIR measure in the context of two imbalanced data sets. Without loss of generality, we first denote the positive data set by $\mathcal{P} := \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_+}\}$ which are sampled from an unknown distribution F, and the negative data set by $\mathcal{N} := \{\mathbf{x}_{N_+ +1}, \mathbf{x}_{N_+ +2}, \cdots, \mathbf{x}_N\}$ which are sampled from an unknown distribution G. We then have $\mathcal{X} = \mathcal{P}\cup\mathcal{N}$ and $N = N_+ + N_-$. The distributions of F and G are assumed to be absolutely continuous with respect to Lebesgue measure, and their corresponding densities are denoted by $f = f(\mathbf{x})$ and $g = g(\mathbf{x})$, respectively. Motivated by the theoretical results in [41,42] for two-sample tests, we obtain the following theorem that gives the asymptotic mean of our GIR measure with any general distributions of F and G.

Theorem 1. *If $N_+, N_- \to \infty$ with $N_+/N = \lambda_+$ and $N_-/N = \lambda_-$, where both λ_+ and λ_- are finite values, then the expectation of the GIR measure ΔT can be given by*

$$\lim_{N\to\infty}\mathbb{E}(\Delta T) = \int \frac{\lambda_- g^2 - \lambda_+ f^2}{\lambda_- g + \lambda_+ f} d\mathbf{x} \quad (4)$$

Proof. We only consider the case $k = 1$ in our proof, and the situation for $k > 1$ follows similarly. We have

$$\begin{aligned} \mathbb{E}(\Delta T) &= \mathbb{E}(T_-) - \mathbb{E}(T_+) \\ &= \mathbb{P}(I_1(\mathbf{x}_{N_+ +1},\mathcal{X}) = 1) - \mathbb{P}(I_1(\mathbf{x}_1,\mathcal{X}) = 1) \end{aligned} \quad (5)$$

where $I_1(\mathbf{x},\mathcal{X})$ equals one if both $\mathbf{x}$ and its nearest neighbor $NN_1(\mathbf{x},\mathcal{X})$ belong the same class, otherwise $I_1(\mathbf{x},\mathcal{X}) = 0$. We also have

$$\begin{aligned} &\mathbb{P}(I_1(\mathbf{x}_1,\mathcal{X}) = 1) = (N_+ - 1)\mathbb{P}(NN_1(\mathbf{x}_1,\mathcal{X}) = \mathbf{x}_2) \\ &\quad = (N_+ - 1)\int_{\mathbb{R}^d} f(\mathbf{x}_1)\int_{\mathbb{R}^d} f(\mathbf{x}_2)\left(1 - \int_S f(\mathbf{x})d\mathbf{x}\right)^{N_+ - 2} \\ &\quad\times\left(1 - \int_S g(\mathbf{x})d\mathbf{x}\right)^{N_-} d\mathbf{x}_2 d\mathbf{x}_1 \end{aligned} \quad (6)$$

where S is the sphere centered at $\mathbf{x}_1$ with the radius of $\|\mathbf{x}_2 - \mathbf{x}_1\|$. If we make a variable change $\omega = n^{1/d}(\mathbf{x}_2 - \mathbf{x}_1)$ and apply the Lebesgue Differentiation Theorem and the Dominated Convergence Theorem (see Theorem 3.4 in [41] or Theorem 2 in [42]), we have

$$\begin{aligned} &\lim_{N\to\infty}\mathbb{P}(I_1(\mathbf{x}_1,\mathcal{X}) = 1) \\ &\quad = \int_{\mathbb{R}^d} f^2(\mathbf{x}_1)\int_{\mathbb{R}^d}\exp\left\{-[f(\mathbf{x}_1) + \frac{\lambda_-}{\lambda_+}g(\mathbf{x}_1)]K_d\|\omega\|^d\right\}d\omega d\mathbf{x}_1 \end{aligned} \quad (7)$$

where K_d is the volume of the unit ball in $\mathbb{R}^d$. Putting $\rho = \|\omega\|^d$ in Eq. (7) produces

$$\begin{aligned} &\lim_{N\to\infty}\mathbb{P}(I_1(\mathbf{x}_1,\mathcal{X}) = 1) \\ &\quad = \int_{\mathbb{R}^d} f^2(\mathbf{x}_1)\int_0^{+\infty}\exp\left\{-[(f(\mathbf{x}_1) + \frac{\lambda_-}{\lambda_+}g(\mathbf{x}_1))K_d\rho]\right\} \\ &\quad\times K_d d\rho d\mathbf{x}_1 \end{aligned} \quad (8)$$

where we apply the transformation of spherical coordinate. Hence, we have

$$\lim_{N\to\infty}\mathbb{P}(I_1(\mathbf{x}_1,\mathcal{X}) = 1) = \lambda_+\mathbb{E}_f\left[\frac{f}{\lambda_+ f + \lambda_- g}\right] \quad (9)$$

where $\mathbb{E}_f$ is the expectation with respect to the distribution of F.

Similarly, one can obtain

$$\lim_{N\to\infty}\mathbb{P}(I_1(\mathbf{x}_{N_+ +1},\mathcal{X}) = 1) = \lambda_-\mathbb{E}_g\left[\frac{g}{\lambda_+ f + \lambda_- g}\right] \quad (10)$$

Applying Eqs. (9) and (10) into Eq. (6), we have

$$\begin{aligned}\mathbb{E}(\Delta T) &= \lambda_-\mathbb{E}_g\left[\frac{g}{\lambda_+ f+\lambda_- g}\right] - \lambda_+\mathbb{E}_f\left[\frac{f}{\lambda_+ f+\lambda_- g}\right]\\ &= \int \frac{\lambda_- g^2 - \lambda_+ f^2}{\lambda_- g+\lambda_+ f}d\mathbf{x} \end{aligned}\tag{11}$$

For a general $k >= 1$, the expression given by Eq. (11) remains unchanged. □

From Theorem 1, it can be shown that the expectation of the GIR measure tends to zero when the two data sets have balanced sample size ratio, i.e., $\lambda_+ = \lambda_-$, as given in Proposition 1. In other words, when two classes have the same and infinite number of data samples, our proposed GIR measure is equivalent to the measure of sample size ratio.

Proposition 1. *If $\lambda_+ = \lambda_-$, the asymptotic mean of the GIR measure ΔT tends to 0, i.e.,*

$$\lim_{N\to\infty}\mathbb{E}(\Delta T) = 0 \tag{12}$$

Proof. The proof is straightforward using Theorem 1. When $\lambda_+ = \lambda_-$, we have

$$\begin{aligned}\lim_{N\to\infty}\mathbb{E}(\Delta T) &= \int \frac{\lambda_- g^2 - \lambda_+ f^2}{\lambda_- g+\lambda_+ f}d\mathbf{x}\\ &= \int \frac{g^2 - f^2}{g+f}d\mathbf{x}\\ &= \int g d\mathbf{x} - \int f d\mathbf{x}\\ &= 1-1=0\end{aligned}\tag{13}$$

□

4. GIR-based ensemble sampling approaches

In this section, we present two GIR-based ensemble sampling approaches: GIR-based ensemble undersampling (GIREnUS) approach and GIR-based ensemble oversampling (GIREnOS) approach. The key idea of these two sampling schemes is to build a strong classifier with several weak classifiers in an ensemble manner which is the combination of boosting and bagging strategies. To train weak classifiers, balanced data sets are constructed adaptively with small GIRs (e.g., $\Delta T < 0$), instead of a balanced sample size in conventional sampling approaches. To begin with, we note that the formation of new balanced data sets and the calculation of GIR in these two approaches are based on the KNN graph of the given original data set. Hence, we introduce a preprocessing stage for both approaches to build the KNN graph denoted by $\mathcal{G}_{\text{KNN}}(\mathcal{X})$ in which the training data samples are vertices and the distance of the sample to its nearest neighbors are the edges.

4.1. GIREnUs: GRI-based ensemble undersampling approach

Undersampling scheme is an effective approach for imbalance learning, which aims to remove a subset of negative samples from the original data set. Since the undersampling approach may throw away many useful positive samples and the classifier trained in the subset of data is usually weak, it is necessary to build a strong classifier in an ensemble manner by combining multiple weak classifiers.

We summarize our GRIEnUS approach in Algorithm 1 . It is an ensemble and adaptive method based on the GIR metric. For a given imbalanced data set, we first calculate the point-wise statistic $t_k(\mathbf{x})$ using the KNN graph $\mathcal{G}_{\text{KNN}}(\mathcal{X})$:

$$t_k(\mathbf{x}) = \frac{1}{k}\sum_{r=1}^{k} I_r(\mathbf{x},\mathcal{X}) \tag{14}$$

Notice that both T_+ and T_- can be also obtained according to Eqs. (1) and (2). The point-wise statistic $t_k(\mathbf{x})$ also reflects the difficulty in learning, e.g., if the nearest neighbors classification rule is used. Given $t_k(\mathbf{x})$ for each negative sample, we calculate the density distribution $p_-(\mathbf{x})$ for negative class:

$$p_-(\mathbf{x}) = \frac{t_k(\mathbf{x})+1}{\sum_{\mathbf{x}\in\mathcal{N}} t_k(\mathbf{x}) + N_-} \tag{15}$$

We consider $p_-(\mathbf{x})$ as the density distribution which measures the distribution of weights for different negative samples in terms of their level of easiness in learning. Using the density distribution $p_-(\mathbf{x})$ as the sampling probability, we further form U balanced data sets by removing those easy to learn negative samples. By doing so, the classifier trained in the new data set more focuses on those samples that are difficult to learn. We denote these U balanced data sets as $\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_U$, and each balanced data set can be considered as a balanced learning subproblem. Motivated by the success of EasyEnsemble [6], we apply the scheme of “ensemble of ensembles” to construct a strong classification rule. Specifically, given a new balanced data set $\mathcal{P}\cup\mathcal{N}_i$, s_i weak classifiers h_{ij}, $j = 1, 2, \cdots, s_i$ are then trained with the AdaBoost learning scheme [16]. For these learned classifiers from U balanced data sets, a

Algorithm 1: GIREnUS: GIR-based ensemble undersampling approach for imbalanced learning.

```
input : $\mathcal{X}$: Data set $\mathcal{X} = \mathcal{P}\cup\mathcal{N}$ with $N$ samples, consisting of
        $N_+$ positive samples in $\mathcal{P}$ and $N_-$ negative samples
        in $\mathcal{N}$
Procedure:
1. Calculate the point-wise statistic $t_k(\mathbf{x})$ for all samples in $\mathcal{X}$
using Eq. (14), and $T_+$ and $T_-$ using Eq. (1) and (2),
respectively;
2. Calculate the sampling probability $p_-(\mathbf{x})$ in Eq. (15) for
negative samples;
for $i$ = $1 : U$ do
    3. $\Delta T = T_- - T_+$ and $\mathcal{N}_i = \mathcal{N}$;
    repeat
        a. Sample a negative sample $\mathbf{x}_s$ with the sampling
        probability distribution $p_-(\mathbf{x})$, and remove $\mathbf{x}_s$ from the
        data set:
            $\mathcal{N}_i = \{\mathbf{x} : \mathbf{x}\in\mathcal{N}_i, \mathbf{x}\neq\mathbf{x}_s\}$        (16)
        b. Update $T_+$ and $T_-$ using $\mathcal{P}$ and $\mathcal{N}_i$;
        c. Calculate $\Delta T = T_- - T_+$;
    until $\Delta T \leq 0$;
    4. Learn an ensemble classifier $H_i$ using $\mathcal{P}$ and $\mathcal{N}_i$ which
    combines $s_i$ weak classifiers $h_{ij}$ with the weights $\beta_{ij}$:
    for $j$ = $1 : L$ do
        a. Train a weak classifier $h_{ij}$ with the sampling
        distribution $D_j(\mathbf{x})$ which is initially a uniform
        distribution;
        b. Calculate the error rate $\varepsilon_j$. If $\varepsilon_j > 0.5$, then set $s_i = j$
        and abort loop;
        c. Set $\beta_{ij} = \varepsilon_j/(1-\varepsilon_j)$;
        d. Update $D_j(\mathbf{x})$: $D_{j+1}(\mathbf{x}) = D_j(\mathbf{x})\times\beta_{ij}$ if $\mathbf{x}$ is correctly
        classified by $h_{ij}$;
        e. Normalize $D_{j+1}$ so that it is a distribution;
    end
end
5. Construct the final classifier $H$:
    $H = \text{argmax}_{y\in\{+1,-1\}} \sum_{i=1}^{U}\sum_{j=1}^{s_i} \log\frac{1}{\beta_{ij}}$        (17)
```

bagging-like strategy is used to build a single ensemble classifier *H* for final classification.

4.2. GIREnOs: GIR-based ensemble oversampling approach

Unlike the undersampling scheme, the oversampling scheme attempts to generate new positive samples to enhance the learning of classifiers for positive samples. Many synthetic sample generation methods have been proposed in past decades, such as random oversampling, SMOTE, SMOTEBoost, AdaSyn, etc. Motivated by the success of SMOTE and AdaSyn, we develop an GIR-based ensemble oversampling (GIREnOS) approach. Similar to the GIREnUS, the GIREnOS is an ensemble and adaptive method based on the GIR metric. We first calculate the point-wise statistic $t_k(\mathbf{x})$ for all positive and negative samples, and the class-wise generalized statistics T_+ and T_- using $t_k(\mathbf{x})$. A density distribution $p_+(\mathbf{x})$ for positive class is obtained as follows:

$$p_+(\mathbf{x}) = 1 - \frac{t_k(\mathbf{x}) + 1}{\sum_{\mathbf{x}\in\mathcal{P}} t_k(\mathbf{x}) + N_+} \tag{18}$$

Unlike $p_-(\mathbf{x})$ in GIREnUS, the density distribution $p_+(\mathbf{x})$ is able to measure the level of difficulty of positive samples for learning. By doing so, the classifier trained in the new data set more focuses on those positive samples that are difficult to learn.

We summarize the GIREnOS approach in Algorithm 2. With the sampling probability $p_+(\mathbf{x})$, it forms U balanced subsets, each of which consists of all the original positive and negative samples and some additional synthetic positive samples. These additional synthetic positive samples are generated and added into the original data set to form a balanced data set in which we measure the balance using the GIR metric. For synthetic sample generation, it first randomly selects a referenced positive sample with the distribution of $p_+(\mathbf{x})$, denoted by $\mathbf{x}_r$. Then, given the reference sample $\mathbf{x}_r$, a new positive sample is generated using the SMOTE algorithm which includes two stages: (i) randomly choose one positive sample $\mathbf{x}_z$ from the K nearest neighbors of $\mathbf{x}_r$; (ii) generate a synthetic data sample $\mathbf{x}$ as follows:

$$\mathbf{x}_p = \mathbf{x}_r + (\mathbf{x}_z - \mathbf{x}_r) \circ \lambda \tag{21}$$

where $\boldsymbol{\lambda}$ is a random vector, each of element ranges from 0 to 1, and the symbol "$\circ$" denotes element-wise multiplication of two vectors. The procedure continues until a balanced data set is formed. For the new balanced data set $\mathcal{P}_i \cup \mathcal{N}$, s_i weak classifiers h_{ij}, $j = 1, 2, \cdots, s_i$, are trained using AdaBoost learning scheme. We repeat the construction of balanced data set and the training of multiple weak classifiers U times, and construct a single final classifier H using a bagging-like manner as given in Eq. (20).

4.3. Discussion

As shown in Algorithms 1 and 2, the proposed GIREnUS and GIREnOS approaches are based on the general adaptive ensemble learning framework with the undersampling and oversampling scheme, respectively. We note that the significance of this approach has three folds: first, using the density distribution $p_-(\mathbf{x})$ as the sampling distribution, the learning algorithms in these subproblems focus on the samples that are difficult to learn, therefore improving learning performance. Notice that this adaptive learning mechanism is similar to the our previous AdaSYN approach in which a density distribution is constructed to determine the number of new synthetic positive samples to be generated [5,11]. However, unlike the AdaSYN, we generate new positive samples or remove negative samples in probabilistic manners. Second, the balance of new constructed data sets is measured by the new metric of GIR, instead of the sample size ratio. Theoretical properties show the consistency of the GIR measure and illustrate the asymptotic equivalence to the sample size ratio under the conditions of either balanced sample size ratio ($\lambda_+ = \lambda_-$) or two well-mixed distributions ($F = G$). The idea behind our two GIR-based sampling schemes is to ensure the newly formed data set has a balanced class distribution in each subproblem and the bias issue could be avoid in the new data set. Third, it adopts the scheme of "ensemble of ensembles": while the boosting strategy is used in each subproblem, the bagging strategy is used to build a single classifier by combining T ensembles. It has been known that the boosting scheme aims to reduce the bias, while the bagging scheme mainly reduces the variance by averaging. The combination of these two different types of ensemble schemes helps the final classifier to achieve stronger generalization.

Algorithm 2: GIREnOS: GIR-based ensemble oversampling approach for imbalanced learning.

```
input : $\mathcal{X}$: Data set $\mathcal{X} = \mathcal{P} \cup \mathcal{N}$ with $N$ samples, consisting of
        $N_+$ positive samples in $\mathcal{P}$ and $N_-$ negative samples
        in $\mathcal{N}$;
Procedure:
1. Calculate the point-wise statistic $t_k(\mathbf{x})$ for all samples in $\mathcal{X}$
using Eq. (14), and $T_+$ and $T_-$ using Eq. (1) and (2),
respectively;
2. Calculate the sampling probability $p_+(\mathbf{x})$ in Eq. (18) for
positive samples;
for $i$ = 1 : $U$ do
    3. $\Delta T = T_- - T_+$ and $\mathcal{P}_i = \mathcal{P}$.
    repeat
        a. Sample a positive sample $\mathbf{x}_r$ with the sampling
        probability distribution $p_+(\mathbf{x})$, and generate a new
        positive sample $\mathbf{x}_p$ based on $\mathbf{x}_r$ using Eq. (21);
        b. Add the new positive sample $\mathbf{x}_p$ in $\mathcal{P}_i$:
        $\mathcal{P}_i = \{\mathcal{P}_i \cup \mathbf{x}_p\}$    (19)
        c. Update $T_+$ and $T_-$ using $\mathcal{P}_i$ and $\mathcal{N}$;
        d. Calculate $\Delta T = T_- - T_+$;
    until $\Delta T \le 0$;
    4. Learn an ensemble classifier $H_i$ using $\mathcal{P}_i$ and $\mathcal{N}$ which
    combines $s_i$ weak classifiers $h_{ij}$ with the weights $\beta_{ij}$:
    for $j$ = 1 : $L$ do
        a. Train a weak classifier $h_{ij}$ with the sampling
        distribution $D_j(\mathbf{x})$ which is initially a uniform
        distribution;
        b. Calculate the error rate $\varepsilon_j$. If $\varepsilon_j > 0.5$, then set $s_i = j$
        and abort loop;
        c. Set $\beta_{ij} = \varepsilon_j/(1 - \varepsilon_j)$;
        d. Update $D_j(\mathbf{x})$: $D_{j+1}(\mathbf{x}) = D_j(\mathbf{x}) \times \beta_{ij}$ if $\mathbf{x}$ is correctly
        classified by $h_{ij}$;
        e. Normalize $D_{j+1}$ so that it is a distribution;
    end
end
5. Construct the final classifier $H$:
$H = \mathrm{argmax}_{y\in\{+1,-1\}} \sum_{i=1}^{U} \sum_{j=1}^{s_i} \log \frac{1}{\beta_{ij}}$    (20)
```

5. Experiments and result analysis

5.1. Performance evaluation metrics

It is known that the overall accuracy is not a good performance evaluation metric for imbalanced learning, since the size of positive data is much less than the size of negative data and the goal of imbalanced learning is to improve the prediction performance

Table 1
Confusion matrix.

	Predicted positive	Predicted negative
Actual positive	TP (True positives)	FN (False negatives)
Actual negative	FP (False positives)	TN (True negatives)

on positive data without hurting the performance on negative data too much. In our experiments, we use the comprehensive metrics of F-Measure, G-Mean and AUC to evaluate the performance of classifiers. The F-Measure and G-Mean metrics are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F-Measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$\text{G-Mean} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \times \frac{\text{TN}}{\text{TN} + \text{FP}}}$$

where TP (true positive) denotes the number of positive samples that are classified correctly, TN (true negative) denotes the number of negative samples that are classified correctly, FP (false positive) denotes the number of negative samples that are misclassified as positive, and FN (false negative) denotes the number of positive samples that are misclassified as negative. The relationship among these four metrics is illustrated in the confusion matrix shown in Table 1. It is also known that the metric of precision measures the exactness of a classifier, while the metric of recall measures the completeness of a classifier [1]. Hence, F-Measure and G-Mean metrics are two comprehensive performance assessments in imbalanced learning since both are the combination of precision and recall. AUC (Area Under the ROC Curve) is also a reliable performance evaluation metric for imbalanced learning, since the ROC curve reflects the classifier's performance with all possible thresholds using the pairs of false positive rate and true positive rate. AUC is the area under the ROC curve, which integrates the performance of a classification method over all possible false positive rates.

5.2. Experimental settings

In the experiments, we compare the performance of various classifiers on 15 UCI real-life data sets [43] shown in Table 2. These data sets have varying sizes, features, and class-imbalanced ratios. Some of these data sets are two-class, while the others are multi-class data sets. Following the previous work in literature [1,44], we modify or re-arrange class labels of multi-class data sets to two-class data sets. For example, in *Movement* data set, the class "1" is considered as the positive class and the others are considered as the negative class. In *Seeds* data set, the class "Kama" is considered as the positive class and the classes of "Rosa" and "Canadian" are considered as the negative class. We summarize the characteristics of these 15 UCI real-life data sets and their corresponding modification descriptions in Table 2.

For every data set shown in Table 2, we perform 10-fold cross validation in which the whole data set is randomly split into 10 folds. Each of these 10 folds is tested once, while the remaining 9 folds are used for training. Considering the randomness in the sampling process, we trained all compared classification methods 10 times within each fold and the results (F-Measure, G-Mean, and AUC) are averaged over these 10 runs. We further repeat this 10-fold cross validation for 10 times, and obtain final results which are the averages over these 10 cross validations.

We compare our proposed GIREnUS and GIREnOS approaches with 12 approaches which have been commonly used for imbalanced learning. Following the experimental settings in [6], we summarize these 12 approaches as follows:

1. CART (abbreviated as *ORI*): Classification And Regression Tree [45] uses the original data set without sampling for training the classifier.
2. CART with Bagging (abbreviated as *BAGG*): Bagging [15] is used as the ensemble method, and CART is used as the base classifier. The number of iteration for bagging is 40. Similar to *ORI*, the original data set without sampling is used for training the classifier.
3. CART with AdaBoost (abbreviated as *ADA*): AdaBoost.M1 [17] is used as the ensemble method, and CART is used as the base classifier. The number of iteration for boosting is 40. Similar to *ORI* and *BAGG*, the original data set without sampling is used for training the classifier.
4. Random Forest (abbreviated as *RF*): Random Forest [46] uses feature bagging to form random classification trees and build a final classifier with the majority voting. At each split, $\sqrt{d}$ features are used. The number of iteration for bagging is 40.
5. Undersampling + Bagging (abbreviated as *USBAGG*): *USBAGG* applies a uniform distribution to remove negative samples for training each CART classifier. The final classifier is formed with the majority voting. The number of iteration for bagging is 40.

Table 2
Characteristics of 15 UCI data sets.

Data sets	# Features	# Positives	# Negatives	SSR*	GIR**
Pima-indians-diabetes	8	268 (class "0")	500 (class "1")	1.87	0.22
Ionosphere	34	126 (class "g")	225 (class "b")	1.79	0.39
Haberman	3	81 (class "1")	225 (class "2")	2.78	0.47
Parkinsons	22	48 (class "1")	147 (class "0")	3.06	0.29
Vertebral	6	100 (class "AB")	210 (class "NO")	2.10	0.14
Breastcancer	9	241 (class "R")	452 (class "N")	1.90	0.05
Breasttissue	9	21 (class "carcinoma")	85 (others)	4.05	0.30
Movement	90	24 (class "1")	336 (others)	14.00	0.44
Spect	22	55 (class "1")	212 (class "0")	3.85	0.80
Glass	9	51 (class "1", "2" and "3")	163 (others)	3.20	0.19
ILPD	8	268 (class "positive")	500 (class "negative")	1.87	0.26
Seeds	9	241 (class "Kama")	452 (others)	2.00	0.123
Sonar	60	97 (class "R")	111 (class "M")	1.14	0.155
Mfeat-mor	6	200 (class "10")	1800 (others)	9.00	0.69
Mfeat-zer	47	200 (class "10")	1800 (others)	9.00	0.51

SSR* = Sample Size Ratio. GIR** = Generalized Imbalance Ratio

Table 3
F-Measure of all compared methods. The F-Measure average is followed by the standard deviation. The method with the best F-Measure average is highlighted with **Bold** value.

Dataset	*Pima*	*Ionosphere*	*Haberman*	*Parkinsons*	*Vertebral*
ORI	0.586 ± 0.08	0.826 ± 0.08	0.331 ± 0.16	0.694 ± 0.14	0.704 ± 0.10
BAGG	0.629 ± 0.05	0.885 ± 0.06	0.299 ± 0.14	0.769 ± 0.14	0.734 ± 0.10
ADA	0.611 ± 0.05	0.908 ± 0.06	0.337 ± 0.13	0.840 ± 0.10	0.725 ± 0.09
RF	0.654 ± 0.04	0.889 ± 0.06	0.429 ± 0.11	0.748 ± 0.11	0.723 ± 0.08
USBAGG	0.662 ± 0.05	0.852 ± 0.06	0.471 ± 0.11	0.740 ± 0.12	0.748 ± 0.08
OSBAGG	0.613 ± 0.06	0.850 ± 0.08	0.342 ± 0.14	0.784 ± 0.13	0.696 ± 0.11
SMOTEADA	0.618 ± 0.05	0.899 ± 0.05	0.336 ± 0.11	0.843 ± 0.09	0.715 ± 0.09
RFUS	0.655 ± 0.04	0.888 ± 0.06	0.430 ± 0.12	0.745 ± 0.11	0.724 ± 0.08
RFOS	0.580 ± 0.05	0.890 ± 0.06	0.320 ± 0.13	0.795 ± 0.13	0.678 ± 0.11
EASY	0.649 ± 0.05	0.891 ± 0.06	0.435 ± 0.10	0.756 ± 0.10	0.739 ± 0.08
ADASYN	0.635 ± 0.05	0.847 ± 0.06	0.354 ± 0.13	0.796 ± 0.12	0.712 ± 0.10
RBBagg	0.665 ± 0.05	0.873 ± 0.06	0.430 ± 0.12	0.777 ± 0.11	0.751 ± 0.08
GIREnUS	**0.686 ± 0.05**	0.912 ± 0.05	**0.513 ± 0.09**	0.811 ± 0.09	**0.779 ± 0.07**
GIREnOS	0.666 ± 0.05	**0.922 ± 0.05**	0.407 ± 0.13	**0.887 ± 0.08**	0.765 ± 0.09
Dataset	*Breastcancer*	*Movement*	*Breasttissue*	*Glass*	*ILPD*
ORI	0.913 ± 0.04	0.476 ± 0.33	0.806 ± 0.22	0.824 ± 0.13	0.386 ± 0.09
BAGG	0.940 ± 0.03	0.464 ± 0.33	0.784 ± 0.22	0.854 ± 0.11	0.348 ± 0.09
ADA	0.948 ± 0.03	0.791 ± 0.23	0.586 ± 0.28	0.863 ± 0.12	0.435 ± 0.08
RF	0.959 ± 0.03	0.688 ± 0.16	0.861 ± 0.14	0.895 ± 0.08	0.549 ± 0.05
USBAGG	0.936 ± 0.03	0.567 ± 0.18	0.813 ± 0.20	0.875 ± 0.09	0.537 ± 0.05
OSBAGG	0.921 ± 0.03	0.634 ± 0.29	0.787 ± 0.21	0.841 ± 0.11	0.381 ± 0.09
SMOTEADA	0.946 ± 0.03	0.829 ± 0.19	0.752 ± 0.25	0.869 ± 0.09	0.460 ± 0.07
RFUS	0.958 ± 0.03	0.686 ± 0.17	0.865 ± 0.14	0.898 ± 0.08	0.549 ± 0.05
RFOS	0.960 ± 0.03	0.629 ± 0.31	0.853 ± 0.18	**0.900 ± 0.09**	0.373 ± 0.09
EASY	0.951 ± 0.03	0.494 ± 0.14	0.827 ± 0.15	0.867 ± 0.09	0.534 ± 0.05
ADASYN	0.929 ± 0.03	0.813 ± 0.20	0.830 ± 0.19	0.842 ± 0.11	0.481 ± 0.08
RBBagg	0.947 ± 0.03	0.764 ± 0.19	0.839 ± 0.19	0.875 ± 0.09	0.525 ± 0.05
GIREnUS	**0.961 ± 0.02**	0.613 ± 0.16	**0.868 ± 0.14**	0.887 ± 0.09	**0.579 ± 0.05**
GIREnOS	0.954 ± 0.03	**0.851 ± 0.19**	0.816 ± 0.18	0.899 ± 0.09	0.528 ± 0.07
Dataset	*SPECT*	*Mfeat-mor*	*Mfeat-zer*	*Wine*	*Yeast*
ORI	0.484 ± 0.21	0.265 ± 0.09	0.231 ± 0.08	0.910 ± 0.09	0.757 ± 0.06
BAGG	0.502 ± 0.21	0.237 ± 0.07	0.158 ± 0.06	0.944 ± 0.07	**0.807 ± 0.05**
ADA	0.509 ± 0.18	0.312 ± 0.07	0.184 ± 0.06	0.647 ± 0.23	0.789 ± 0.05
RF	0.540 ± 0.13	**0.644 ± 0.05**	0.550 ± 0.05	**0.985 ± 0.03**	0.752 ± 0.04
USBAGG	0.512 ± 0.13	0.632 ± 0.05	0.543 ± 0.05	0.948 ± 0.06	0.794 ± 0.04
OSBAGG	0.503 ± 0.18	0.323 ± 0.08	0.183 ± 0.07	0.915 ± 0.06	0.773 ± 0.06
SMOTEADA	0.473 ± 0.19	0.349 ± 0.08	0.306 ± 0.07	0.960 ± 0.06	0.784 ± 0.05
RFUS	0.544 ± 0.13	**0.644 ± 0.05**	0.549 ± 0.05	0.986 ± 0.02	0.749 ± 0.04
RFOS	0.553 ± 0.13	0.330 ± 0.08	0.086 ± 0.05	0.981 ± 0.03	0.728 ± 0.05
EASY	0.507 ± 0.14	0.624 ± 0.05	0.553 ± 0.05	0.951 ± 0.05	0.780 ± 0.04
ADASYN	0.438 ± 0.18	0.375 ± 0.08	0.263 ± 0.07	0.944 ± 0.06	0.781 ± 0.05
RBBagg	0.513 ± 0.16	0.573 ± 0.05	0.466 ± 0.06	0.956 ± 0.05	0.805 ± 0.04
GIREnUS	**0.573 ± 0.13**	0.637 ± 0.05	**0.583 ± 0.05**	0.968 ± 0.05	0.799 ± 0.04
GIREnOS	0.526 ± 0.19	0.374 ± 0.08	0.342 ± 0.08	0.973 ± 0.05	0.806 ± 0.04

6. Oversampling + Bagging (abbreviated as *OSBAGG*): *OSBAGG* applies a uniform distribution to replicate positive samples for training each CART classifier. Similar to other bagging-based approaches, the majority voting is used to form the final classifier. The number of iteration for bagging is 40.
7. SMOTE + AdaBoost (abbreviated as *SMOTEADA*): *SMOTEADA* [18] combines the SMOTE algorithm and the boosting procedure to create new synthetic positive samples. The oversampling distribution P_+ is then updated according to the previous classifiers' capacity. The number of iteration for boosting is 40. The number of nearest neighbors to be considered in SMOTE is 7.
8. Undersampling + Random Forest (abbreviated as *USRF*): At each subclassifier in Random Forest, the negative samples are undersampled with a uniform distribution. At each split, $\sqrt{d}$ features are used. The number of iteration for bagging is 40.
9. Oversampling + Random Forest (abbreviated as *OSRF*): At each subclassifier in Random Forest, random oversampling is used to create new synthetic positive samples. At each split in Random Forest, $\sqrt{d}$ features are used. The number of iteration for bagging is 40.
10. EasyEnsemble (abbreviated as *EASY*): *EASY* [6] applies both boosting and bagging strategies for classification. The number of iteration for bagging is 10, and the number of iteration for boosting is 4.
11. ADASYN + Bagging: (abbreviated as ASBagg): ASBagg [5] applies the bagging to the ADASYN [5]. The number of iteration for bagging is 40, and the number of nearest neighbors to be considered in ADASYN is 7.
12. Random Balance + Bagging: (abbreviated as RBBagg): RBBagg applies the bagging to the Random Balance [47], which forms a balanced data set through either oversampling or undersampling randomly. The number of iteration for bagging is 40, and the number of nearest neighbors to be considered in SMOTE for oversampling is 7.

Among these 12 methods, the first 4 methods are performed on the original data set without sampling, while the last 8 methods are sampling-based imbalanced learning methods which forms balanced data sets in size (i.e., the sample size ratio equals 1). Similar to the EASY, in our proposed GIREnUS and GIREnOS methods, the number of iteration for bagging U is set to 10, and the number of iteration for boosting L is set to 4. We note that the learning performance of the final classifier in an ensemble method would be related to the number of iterations (weak classifiers). In our experiments, while the CART-ORI method serves as the baseline method

Table 4
G-Mean of all compared methods. The G-Mean average is followed by the standard deviation. The method with the best G-Mean average is highlighted with **Bold** value.

Dataset	*Pima*	*Ionosphere*	*Haberman*	*Parkinsons*	*Vertebral*
ORI	0.673 ± 0.06	0.863 ± 0.06	0.474 ± 0.17	0.787 ± 0.11	0.777 ± 0.09
BAGG	0.705 ± 0.04	0.907 ± 0.05	0.435 ± 0.16	0.822 ± 0.12	0.797 ± 0.08
ADA	0.693 ± 0.04	0.921 ± 0.05	0.492 ± 0.12	0.881 ± 0.08	0.792 ± 0.07
RF	0.727 ± 0.04	0.913 ± 0.05	0.585 ± 0.10	0.851 ± 0.08	0.799 ± 0.07
USBAGG	0.734 ± 0.05	0.886 ± 0.05	0.621 ± 0.10	0.842 ± 0.09	0.821 ± 0.06
OSBAGG	0.694 ± 0.05	0.879 ± 0.06	0.497 ± 0.13	0.846 ± 0.10	0.771 ± 0.09
SMOTEADA	0.700 ± 0.04	0.917 ± 0.05	0.497 ± 0.11	0.894 ± 0.07	0.788 ± 0.07
RFUS	0.728 ± 0.04	0.913 ± 0.05	0.586 ± 0.11	0.849 ± 0.08	0.800 ± 0.07
RFOS	0.665 ± 0.04	0.909 ± 0.05	0.471 ± 0.13	0.835 ± 0.11	0.751 ± 0.09
EASY	0.723 ± 0.04	0.913 ± 0.05	0.586 ± 0.09	0.860 ± 0.07	0.813 ± 0.07
ADASYN	0.713 ± 0.04	0.880 ± 0.05	0.508 ± 0.13	0.856 ± 0.09	0.787 ± 0.08
RBBagg	0.736 ± 0.04	0.902 ± 0.05	0.582 ± 0.11	0.859 ± 0.08	0.820 ± 0.07
GIREnUS	**0.753 ± 0.04**	0.930 ± 0.05	**0.657 ± 0.08**	0.901 ± 0.06	**0.848 ± 0.06**
GIREnOS	0.739 ± 0.04	**0.936 ± 0.04**	0.562 ± 0.11	**0.929 ± 0.06**	0.830 ± 0.07
Dataset	*Breastcancer*	*Movement*	*Breasttissue*	*Glass*	*ILPD*
ORI	0.934 ± 0.03	0.596 ± 0.38	0.876 ± 0.20	0.878 ± 0.11	0.535 ± 0.08
BAGG	0.955 ± 0.02	0.495 ± 0.35	0.845 ± 0.21	0.896 ± 0.09	0.488 ± 0.08
ADA	0.963 ± 0.02	0.809 ± 0.23	0.622 ± 0.28	0.900 ± 0.10	0.570 ± 0.07
RF	0.974 ± 0.02	**0.884 ± 0.15**	0.933 ± 0.09	**0.950 ± 0.04**	0.674 ± 0.05
USBAGG	0.957 ± 0.02	0.848 ± 0.18	0.889 ± 0.17	0.932 ± 0.06	0.665 ± 0.05
OSBAGG	0.942 ± 0.03	0.727 ± 0.30	0.855 ± 0.19	0.894 ± 0.09	0.526 ± 0.07
SMOTEADA	0.962 ± 0.02	0.872 ± 0.18	0.825 ± 0.24	0.912 ± 0.07	0.599 ± 0.06
RFUS	0.974 ± 0.02	0.879 ± 0.16	0.936 ± 0.09	0.952 ± 0.04	0.674 ± 0.05
RFOS	0.974 ± 0.02	0.653 ± 0.31	0.896 ± 0.16	0.933 ± 0.07	0.509 ± 0.08
EASY	0.967 ± 0.02	0.842 ± 0.17	0.921 ± 0.10	0.930 ± 0.06	0.660 ± 0.05
ADASYN	0.948 ± 0.03	0.881 ± 0.18	0.895 ± 0.15	0.898 ± 0.08	0.614 ± 0.06
RBBagg	0.964 ± 0.02	0.870 ± 0.16	0.909 ± 0.16	0.926 ± 0.07	0.654 ± 0.05
GIREnUS	**0.977 ± 0.01**	0.878 ± 0.15	**0.947 ± 0.07**	0.944 ± 0.05	**0.698 ± 0.04**
GIREnOS	0.968 ± 0.02	**0.884 ± 0.17**	0.889 ± 0.15	0.939 ± 0.06	0.657 ± 0.06
Dataset	*SPECT*	*Mfeat-mor*	*Mfeat-zer*	*Wine*	*Yeast*
ORI	0.617 ± 0.21	0.487 ± 0.10	0.450 ± 0.09	0.932 ± 0.07	0.834 ± 0.04
BAGG	0.616 ± 0.22	0.450 ± 0.08	0.339 ± 0.09	0.954 ± 0.06	0.862 ± 0.04
ADA	0.671 ± 0.19	0.549 ± 0.08	0.379 ± 0.08	0.651 ± 0.23	0.849 ± 0.04
RF	0.747 ± 0.12	0.922 ± 0.02	0.848 ± 0.04	0.989 ± 0.02	0.850 ± 0.03
USBAGG	0.723 ± 0.12	0.920 ± 0.02	0.866 ± 0.03	0.958 ± 0.05	0.884 ± 0.03
OSBAGG	0.678 ± 0.20	0.564 ± 0.08	0.387 ± 0.09	0.939 ± 0.05	0.843 ± 0.04
SMOTEADA	0.631 ± 0.20	0.604 ± 0.08	0.545 ± 0.07	0.969 ± 0.05	0.856 ± 0.04
RFUS	0.751 ± 0.12	0.922 ± 0.02	0.848 ± 0.04	**0.990 ± 0.02**	0.849 ± 0.03
RFOS	0.755 ± 0.12	0.572 ± 0.09	0.219 ± 0.10	0.985 ± 0.02	0.797 ± 0.04
EASY	0.706 ± 0.13	0.912 ± 0.02	0.861 ± 0.03	0.965 ± 0.04	0.876 ± 0.03
ADASYN	0.601 ± 0.20	0.628 ± 0.08	0.499 ± 0.08	0.958 ± 0.05	0.857 ± 0.04
RBBagg	0.685 ± 0.17	0.849 ± 0.04	0.732 ± 0.05	0.965 ± 0.04	0.883 ± 0.03
GIREnUS	**0.777 ± 0.11**	**0.923 ± 0.02**	**0.890 ± 0.03**	0.978 ± 0.03	**0.891 ± 0.03**
GIREnOS	0.666 ± 0.20	0.631 ± 0.08	0.583 ± 0.08	0.979 ± 0.04	0.876 ± 0.03

which is only trained once, all other methods have 40 weak classifiers.

5.3. Experimental results and analysis

In Table 2, we present two metrics of class imbalance measurement: sample size ratio (SSR) and generalized imbalance ratio (GIR). The former one is defined by the ratio of the number of negative samples to the number of positive samples, which is conventionally used in imbalanced learning, while the latter one defined in this paper is the difference between two generalized class-wise statistics. While the sample size ratio cannot reflect the class imbalance of an imbalanced data set, the generalized imbalance ratio offers a new insight to the concept of "imbalance" for imbalanced learning. For example, according to the sample size ratio measurement, the *Movement* data set (14.00) is far more imbalanced than the *Spect* data set (3.85), while according to the generalized imbalance ratio measurement, the *Spect* (0.80) is more imbalanced than the *Movement* (0.44).

We present the F-Measure, G-Mean, and AUC of 14 compared methods on 15 UCI data sets in Tables 3–5. In these three tables, the average value with the standard deviation is given, and the best performance is highlighted with bold value for each data set. From Table 3, it can be shown that the proposed two approaches achieve the best performance for 11 out of 15 data sets in terms of F-Measure metric. From Table 4, one can see that the proposed two approaches outperform the others for 13 out of 15 data sets in terms of G-Mean metric. One interesting observation from these two tables is that the performance metric of G-Mean is related to the metric of F-Measure for all methods that are considered. For example, our GIR-based approaches achieve the best G-Mean for 13 data sets, among which 11 data sets also have the best F-Measure. The reason behind this observation is that both F-Measure and G-Mean are functions of the confusion matrix: while the F-Measure considers both recall and precision, the G-Mean considers both positive class accuracy and negative class accuracy. This indicates that both are comprehensive performance evaluation metrics. For the AUC metric, the proposed two approaches even achieve the best performance for all 15 data sets. While we here only highlight the best approach in term of the average accuracy, we also notice that different approaches have different standard deviations. To further illustrate this improvement in a statistical way, we perform the Welch's *t*-test to determine whether the performance is significantly improved or not, considering both average and stan-

Table 5
AUC of all compared methods. The AUC average is followed by the standard deviation. The method with the best AUC average is highlighted with **Bold** value.

Dataset	*Pima*	*Ionosphere*	*Haberman*	*Parkinsons*	*Vertebral*
ORI	0.622 ± 0.08	0.885 ± 0.08	0.340 ± 0.14	0.762 ± 0.15	0.730 ± 0.11
BAGG	0.679 ± 0.07	0.938 ± 0.05	0.348 ± 0.14	0.837 ± 0.13	0.806 ± 0.10
ADA	0.789 ± 0.04	0.976 ± 0.03	0.634 ± 0.11	0.977 ± 0.03	0.899 ± 0.05
RF	0.657 ± 0.06	0.950 ± 0.04	0.352 ± 0.15	0.833 ± 0.12	0.759 ± 0.11
USBAGG	0.675 ± 0.07	0.913 ± 0.06	0.361 ± 0.14	0.822 ± 0.13	0.802 ± 0.10
OSBAGG	0.649 ± 0.08	0.918 ± 0.05	0.306 ± 0.15	0.833 ± 0.13	0.778 ± 0.11
SMOTEADA	0.784 ± 0.05	0.975 ± 0.03	0.622 ± 0.11	0.975 ± 0.03	0.896 ± 0.05
RFUS	0.659 ± 0.06	0.952 ± 0.04	0.353 ± 0.15	0.828 ± 0.12	0.760 ± 0.11
RFOS	0.644 ± 0.06	0.956 ± 0.04	0.303 ± 0.13	0.875 ± 0.10	0.736 ± 0.11
EASY	0.802 ± 0.04	0.970 ± 0.03	0.648 ± 0.10	0.953 ± 0.04	0.905 ± 0.05
ADASYN	0.647 ± 0.07	0.917 ± 0.06	0.322 ± 0.15	0.845 ± 0.13	0.782 ± 0.11
RBBagg	0.677 ± 0.07	0.942 ± 0.05	0.349 ± 0.14	0.837 ± 0.12	0.815 ± 0.09
GIREnUS	**0.825 ± 0.04**	0.979 ± 0.02	**0.703 ± 0.10**	0.969 ± 0.03	**0.923 ± 0.04**
GIREnOS	0.812 ± 0.04	**0.982 ± 0.02**	0.672 ± 0.11	**0.984 ± 0.03**	0.921 ± 0.04
Dataset	*Breastcancer*	*Movement*	*Breasttissue*	*Glass*	*ILPD*
ORI	0.944 ± 0.04	0.598 ± 0.26	0.856 ± 0.19	0.879 ± 0.11	0.360 ± 0.10
BAGG	0.979 ± 0.02	0.695 ± 0.24	0.871 ± 0.21	0.925 ± 0.09	0.406 ± 0.09
ADA	0.992 ± 0.01	0.975 ± 0.05	0.977 ± 0.04	0.986 ± 0.02	0.730 ± 0.05
RF	0.979 ± 0.02	0.837 ± 0.18	0.894 ± 0.20	0.939 ± 0.08	0.422 ± 0.08
USBAGG	0.974 ± 0.02	0.814 ± 0.20	0.889 ± 0.18	0.896 ± 0.11	0.395 ± 0.09
OSBAGG	0.967 ± 0.03	0.846 ± 0.19	0.808 ± 0.25	0.914 ± 0.10	0.375 ± 0.09
SMOTEADA	0.991 ± 0.01	0.965 ± 0.07	0.962 ± 0.08	0.982 ± 0.02	0.721 ± 0.05
RFUS	0.979 ± 0.02	0.847 ± 0.18	0.896 ± 0.19	0.941 ± 0.08	0.421 ± 0.09
RFOS	0.982 ± 0.02	0.890 ± 0.15	0.882 ± 0.22	0.953 ± 0.07	0.426 ± 0.09
EASY	0.990 ± 0.01	0.930 ± 0.10	0.975 ± 0.05	0.982 ± 0.03	0.725 ± 0.05
ADASYN	0.961 ± 0.03	0.861 ± 0.19	0.858 ± 0.18	0.902 ± 0.11	0.431 ± 0.10
RBBagg	0.979 ± 0.02	0.868 ± 0.17	0.888 ± 0.20	0.930 ± 0.09	0.413 ± 0.10
GIREnUS	0.992 ± 0.01	0.960 ± 0.07	**0.988 ± 0.03**	**0.988 ± 0.02**	**0.757 ± 0.05**
GIREnOS	**0.993 ± 0.01**	**0.983 ± 0.03**	0.973 ± 0.06	**0.988 ± 0.02**	**0.757 ± 0.06**
Dataset	*SPECT*	*Mfeat-mor*	*Mfeat-zer*	*Wine*	*Yeast*
ORI	0.532 ± 0.18	0.248 ± 0.11	0.215 ± 0.08	0.949 ± 0.05	0.764 ± 0.06
BAGG	0.548 ± 0.18	0.209 ± 0.12	0.165 ± 0.07	0.974 ± 0.05	0.837 ± 0.05
ADA	0.791 ± 0.10	0.889 ± 0.03	0.794 ± 0.03	0.998 ± 0.01	0.927 ± 0.02
RF	0.593 ± 0.18	0.261 ± 0.13	0.288 ± 0.12	0.999 ± 0.00	0.807 ± 0.05
USBAGG	0.574 ± 0.17	0.262 ± 0.13	0.247 ± 0.11	0.986 ± 0.03	0.839 ± 0.05
OSBAGG	0.540 ± 0.18	0.175 ± 0.10	0.124 ± 0.06	0.965 ± 0.06	0.806 ± 0.06
SMOTEADA	0.771 ± 0.12	0.889 ± 0.03	0.779 ± 0.04	0.997 ± 0.01	0.927 ± 0.02
RFUS	0.593 ± 0.18	0.260 ± 0.13	0.284 ± 0.12	**0.999 ± 0.00**	0.806 ± 0.05
RFOS	0.593 ± 0.18	0.159 ± 0.09	0.151 ± 0.07	1.000 ± 0.00	0.792 ± 0.06
EASY	0.788 ± 0.11	0.914 ± 0.02	0.899 ± 0.03	0.997 ± 0.01	0.932 ± 0.02
ADASYN	0.501 ± 0.18	0.226 ± 0.10	0.138 ± 0.06	0.961 ± 0.06	0.808 ± 0.06
RBBagg	0.561 ± 0.18	0.218 ± 0.12	0.178 ± 0.09	0.985 ± 0.03	0.844 ± 0.05
GIREnUS	**0.849 ± 0.10**	**0.928 ± 0.02**	**0.915 ± 0.02**	0.998 ± 0.00	**0.942 ± 0.02**
GIREnOS	0.815 ± 0.10	0.900 ± 0.02	0.846 ± 0.03	**0.999 ± 0.01**	0.938 ± 0.02

dard deviations. The Welch's *t*-test is defined as follows:

$$t = \frac{\bar{X}_2 - \bar{X}_1}{\sqrt{S_2^2/n + S_1^2/n}} \tag{22}$$

where $\bar{X}_1$ and $\bar{X}_2$ are the mean of the classification accuracy of two methods, and S_1^2 and S_2^2 are the variance of the classification accuracy of two method. The statistic t follows the t-distribution. The degrees of freedom ν is approximated using the following Welch–Satterthwaite equation:

$$\nu \approx \frac{n^2(n-1)\left(S_2^2/n + S_1^2/n\right)^2}{S_2^4 + S_1^4} \tag{23}$$

The summaries of *t*-test of F-Measure, G-Mean, and AUC between any pair of methods are given in Tables 6, 7, and 8, respectively, with a significance level at 0.05. In each row of these three tables, one individual method is compared with the remaining methods, and the amounts of win-tie-lose are given over 15 UCI data sets.

The results show that the imbalanced learning approaches usually outperform the methods without imbalanced learning, particularly for those data sets with low F-Measure, G-Mean and AUC, such as *Haberman, Movement, ILPD*, etc. This illustrates that the use of imbalanced learning algorithms is beneficial for the learning in imbalanced data sets. In those imbalanced learning approaches, both undersampling and oversampling schemes exhibit their unique advantages for imbalanced learning. For some data sets such as *Pima-indians-diabetes* and *ILPD*, the undersampling scheme could be more effective than the oversampling scheme, but it also could be less effective for others, such as *Parkinsons* and *Seeds*, as compared between *USBAGG* and *OSBAGG*, *USRF* and *OSRF*, and between *GIREnUS* and *GIREnOS*.

Comparing the approaches of *BAGG* and *ADA*, both of which work on original data sets without sampling, the *ADA* always performs better than *BAGG*, as shown in the *t*-test Table 8, where *ADA* can significantly improve the AUC for all 15 data sets. The combination of both boosting and bagging schemes in *GIREnUS* and *GIREnOS* approaches further improves the learning performance. It is also very interesting to notice that, for some data sets, like *Haberman, ILPD, Mfeat-mor* and *Mfeat-zer*, AdaBoost is more effective than bagging no matter they integrate oversampling or undersampling approaches, while it is not true for other data sets. Such observation indicates that a combination of bagging and AdaBoost may improve the generalization of imbalance learning methods, like *EASY* and our two proposed approaches: *GIREnUS* and *GIREnOS*.

Compared with *Easy*, the proposed *GIREnUS* uses the GIR metric to measure class imbalance and adopts an adaptive learning mechanism to form balanced data sets. From Table 8, it can be shown

Table 6
Summary of *t*-test of F-Measure with the significance level at 0.05. In each row, one individual method is compared with all remaining methods, and the amount of win-tie-lose is given over 15 UCI data sets.

Methods	*ORI*	*BAGG*	*ADA*	*RF*	*USBAGG*	*OSBAGG*	*SMOTEADA*	*RFUS*	*RFOS*	*EASY*	*ADASYN*	*RBBagg*	*GIREnUS*	*GIREnOS*
ORI	–	3-6-6	3-4-8	0-4-11	0-2-13	1-10-4	0-12-3	0-3-12	2-5-8	0-3-12	0-6-9	0-2-13	0-0-15	0-2-13
BAGG	6-6-3	–	4-5-6	1-4-10	1-8-6	5-6-4	0-11-4	1-4-10	4-5-6	1-8-6	3-7-5	0-9-6	0-1-14	0-3-12
ADA	8-4-3	6-5-4	–	4-3-8	4-4-7	5-8-2	0-13-2	4-2-9	7-4-4	2-6-7	4-5-6	2-5-8	1-4-10	0-4-11
RF	11-4-0	10-4-1	8-3-4	–	4-9-2	10-4-1	4-11-0	0-15-0	7-7-1	4-10-1	10-2-3	5-7-3	2-5-8	3-7-5
USBAGG	13-2-0	6-8-1	7-4-4	2-9-4	–	10-4-1	3-11-1	2-9-4	7-4-4	3-10-2	8-5-2	3-9-3	0-5-10	3-7-5
OSBAGG	4-10-1	4-6-5	2-8-5	1-4-10	1-4-10	–	0-13-2	2-3-10	3-7-5	1-5-9	1-8-6	0-3-12	0-2-13	0-2-13
SMOTEADA	3-12-0	4-11-0	2-13-0	0-11-4	1-11-3	2-13-0	–	0-11-4	2-13-0	1-11-3	0-15-0	0-12-3	1-9-5	0-15-0
RFUS	12-3-0	10-4-1	9-2-4	0-15-0	4-9-2	10-3-2	4-11-0	–	7-7-1	4-10-1	10-2-3	5-8-2	2-5-8	4-6-5
RFOS	8-5-2	6-5-4	4-4-7	1-7-7	4-4-7	5-7-3	0-13-2	1-7-7	–	6-2-7	5-4-6	2-5-8	1-6-8	0-5-10
EASY	12-3-0	6-8-1	7-6-2	1-10-4	2-10-3	9-5-1	3-11-1	1-10-4	7-2-6	–	7-6-2	2-10-3	0-3-12	2-6-7
ADASYN	9-6-0	5-7-3	6-5-4	3-2-10	2-5-8	6-8-1	0-15-0	3-2-10	6-4-5	2-6-7	–	0-4-11	1-2-12	0-3-12
RBBagg	13-2-0	6-9-0	8-5-2	3-7-5	3-9-3	12-3-0	3-12-0	2-8-5	8-5-2	3-10-2	11-4-0	–	1-4-10	2-9-4
GIREnUS	15-0-0	14-1-0	10-4-1	8-5-2	10-5-0	13-2-0	5-9-1	8-5-2	8-6-1	12-3-0	12-2-1	10-4-1	–	5-8-2
GIREnOS	13-2-0	12-3-0	11-4-0	5-7-3	5-7-3	13-2-0	0-15-0	5-6-4	10-5-0	7-6-2	12-3-0	4-9-2	2-8-5	–

Table 7
Summary of *t*-test of G-Mean with the significance level at 0.05. In each row, one individual method is compared with all remaining methods, and the amount of win-tie-lose is given over 15 UCI data sets.

Methods	*ORI*	*BAGG*	*ADA*	*RF*	*USBAGG*	*OSBAGG*	*SMOTEADA*	*RFUS*	*RFOS*	*EASY*	*ADASYN*	*RBBagg*	*GIREnUS*	*GIREnOS*
ORI	–	3-7-5	3-4-8	0-1-14	0-1-14	1-10-4	0-4-11	0-1-14	3-5-7	0-1-14	0-6-9	0-1-14	0-0-15	0-2-13
BAGG	5-7-3	–	3-5-7	1-3-11	1-4-10	3-7-5	0-9-6	1-3-11	4-5-6	0-3-12	1-9-5	0-3-12	0-0-15	0-2-13
ADA	8-4-3	7-5-3	–	1-3-11	2-2-11	4-9-2	0-10-5	1-3-11	7-3-5	0-6-9	3-5-7	1-5-9	0-2-13	0-3-12
RF	14-1-0	11-3-1	11-3-1	–	5-7-3	13-2-0	10-4-1	0-15-0	8-7-0	4-10-1	10-5-0	7-7-1	1-6-8	6-5-4
USBAGG	14-1-0	10-4-1	11-2-2	3-7-5	–	11-4-0	8-5-2	3-7-5	8-4-3	1-12-2	10-5-0	3-11-1	0-4-11	4-7-4
OSBAGG	4-10-1	5-7-3	2-9-4	0-2-13	0-4-11	–	0-7-8	0-2-13	3-7-5	0-2-13	1-8-6	0-3-12	0-0-15	0-2-13
SMOTEADA	11-4-0	6-9-0	5-10-0	1-4-10	2-5-8	8-7-0	–	1-4-10	8-3-4	1-5-9	4-10-1	1-6-8	0-4-11	0-5-10
RFUS	14-1-0	11-3-1	11-3-1	0-15-0	5-7-3	13-2-0	10-4-1	–	9-6-0	5-8-2	11-4-0	7-7-1	1-6-8	5-6-4
RFOS	7-5-3	6-5-4	5-3-7	0-7-8	3-4-8	5-7-3	4-3-8	0-6-9	–	3-4-8	5-3-7	3-4-8	0-4-11	1-4-10
EASY	14-1-0	12-3-0	9-6-0	1-10-4	2-12-1	13-2-0	9-5-1	2-8-5	8-4-3	–	10-5-0	2-13-0	0-3-12	2-9-4
ADASYN	9-6-0	5-9-1	7-5-3	0-5-10	0-5-10	6-8-1	1-10-4	0-4-11	7-3-5	0-5-10	–	0-4-11	0-1-14	0-4-11
RBBagg	14-1-0	12-3-0	9-5-1	1-7-7	1-11-3	12-3-0	8-6-1	1-7-7	8-4-3	0-13-2	11-4-0	–	0-4-11	2-10-3
GIREnUS	15-0-0	15-0-0	13-2-0	8-6-1	11-4-0	15-0-0	11-4-0	8-6-1	11-4-0	12-3-0	14-1-0	11-4-0	–	9-5-1
GIREnOS	13-2-0	13-2-0	12-3-0	4-5-6	4-7-4	13-2-0	10-5-0	4-6-5	10-4-1	4-9-2	11-4-0	3-10-2	1-5-9	–

Table 8
Summary of *t*-test of AUC with the significance level at 0.05. In each row, one individual method is compared with all remaining methods, and the amount of win-tie-lose is given over 15 UCI data sets.

Methods	*ORI*	*BAGG*	*ADA*	*RF*	*USBAGG*	*OSBAGG*	*SMOTEADA*	*RFUS*	*RFOS*	*EASY*	*ADASYN*	*RBBagg*	*GIREnUS*	*GIREnOS*
ORI	–	2-3-10	0-0-15	0-4-11	0-5-10	2-6-7	0-0-15	0-4-11	2-4-9	0-0-15	1-4-10	1-7-7	0-0-15	0-0-15
BAGG	10-3-2	–	0-0-15	2-9-4	1-11-3	6-8-1	0-0-15	2-9-4	5-6-4	0-0-15	0-14-1	5-9-1	0-0-15	0-0-15
ADA	15-0-0	15-0-0	–	14-1-0	15-0-0	15-0-0	1-14-0	14-1-0	14-1-0	2-11-2	15-0-0	15-0-0	0-7-8	0-8-7
RF	11-4-0	4-9-2	0-1-14	–	4-9-2	7-8-0	0-1-14	0-15-0	3-11-1	1-0-14	3-10-2	6-9-0	0-1-14	0-1-14
USBAGG	10-5-0	3-11-1	0-0-15	2-9-4	–	7-8-0	0-0-15	2-10-3	6-2-7	0-0-15	2-11-2	6-8-1	0-0-15	0-0-15
OSBAGG	7-6-2	1-8-6	0-0-15	0-8-7	0-8-7	–	0-0-15	0-8-7	1-7-7	0-0-15	0-5-10	0-13-2	0-0-15	0-0-15
SMOTEADA	15-0-0	15-0-0	0-14-1	14-1-0	15-0-0	15-0-0	–	14-0-1	14-0-1	2-10-3	15-0-0	15-0-0	0-6-9	0-6-9
RFUS	11-4-0	4-9-2	0-1-14	0-15-0	3-10-2	7-8-0	1-0-14	–	3-11-1	1-0-14	3-10-2	6-9-0	0-1-14	0-1-14
RFOS	9-4-2	4-6-5	0-1-14	1-11-3	7-2-6	7-7-1	1-0-14	1-11-3	–	1-1-13	2-7-6	5-8-2	1-0-14	1-0-14
EASY	15-0-0	15-0-0	2-11-2	14-0-1	15-0-0	15-0-0	3-10-2	14-0-1	13-1-1	–	15-0-0	15-0-0	0-5-10	2-8-5
ADASYN	10-4-1	1-14-0	0-0-15	2-10-3	2-11-2	10-5-0	0-0-15	2-10-3	6-7-2	0-0-15	–	6-9-0	0-0-15	0-0-15
RBBagg	7-7-1	1-9-5	0-0-15	0-9-6	1-8-6	2-13-0	0-0-15	0-9-6	2-8-5	0-0-15	0-9-6	–	0-0-15	0-0-15
GIREnUS	15-0-0	15-0-0	8-7-0	14-1-0	15-0-0	15-0-0	9-6-0	14-1-0	14-0-1	10-5-0	15-0-0	15-0-0	–	3-10-2
GIREnOS	15-0-0	15-0-0	7-8-0	14-1-0	15-0-0	15-0-0	9-6-0	14-1-0	14-0-1	5-8-2	15-0-0	15-0-0	2-10-3	–

that *GIREnUS* can significantly improve the AUC compared with *Easy* for 10 data sets. While the proposed *GIREnUS* approach tries to explore the negative samples, the proposed *GIREnOS* attempts to utilize the limited available positive samples. It can be shown that both of these approaches can improve the performance for most tasks, and *GIREnOS* outperforms *GIREnUS* for two more data sets.

5.4. Complexity analysis

In this section, we analyze the computational complexity of different approaches. Since *RF*, *RFUS* and *RFOS* use random forest as base classifier which has different implementation, it is unfair to compare the training time with others which are all based on CART. Therefore, we only compare the running time of the remaining 10 CART based approaches. We summarize the running time of all 11 approaches for each data set in Table 9, each of which is averaged over all 100 runs. The code is implemented in Matlab R2014a, and runs on a computer with a 2.5GHz Intel Core i5-3210M CPU.

Since all these 10 approaches use the same base classifier and have the same number of weak classifiers, their running time for a given data set is determined by the total number training data and the time used to form the training data set. The undersampling approach *USBagg* randomly remove most majority data and have a smallest number of training data $2N_+$ for classification, and thus it is the fastest approach among all 10 compared approaches. The other two undersampling approaches, *Easy* and our *GIREnUS*,

Table 9
Summary of running time (in seconds) of 10 compared approaches for all data sets. Each value is averaged over 100 runs.

	BAGG	*ADA*	*USBAGG*	*OSBAGG*	*SMOTEADA*	*EASY*	*ADASYN*	*RBBagg*	*GIREnUS*	*GIREnOS*
Pima	1.01	2.65	0.96	1.17	3.23	1.95	3.39	3.81	2.36	4.17
Ionosphere	0.96	1.67	0.95	1.11	2.10	1.39	2.34	2.37	1.63	2.91
Haberman	0.58	0.97	0.58	0.63	1.16	0.70	1.04	1.24	0.78	1.49
Parkinsons	0.80	1.02	0.76	0.88	1.27	0.72	1.39	1.44	0.79	1.50
Vertebral	0.78	1.12	0.77	0.86	1.33	0.85	1.40	1.65	0.94	1.58
Breastcancer	0.76	1.43	0.73	0.80	1.72	1.03	2.80	3.54	1.36	2.65
Movement	1.40	2.26	0.79	1.55	3.83	0.88	4.75	5.90	1.30	5.67
Breasttissue	0.64	0.49	0.65	0.67	0.54	0.37	0.84	0.91	0.39	0.62
Glass	0.68	0.66	0.68	0.72	0.80	0.49	1.11	1.27	0.55	0.99
ILPD	1.10	2.67	1.01	1.36	3.48	1.80	3.16	3.37	2.05	4.28
SPECT	0.76	2.37	0.73	0.84	2.90	1.41	1.42	1.76	1.34	3.61
Mfeat-mor	0.99	6.33	0.72	1.79	11.10	1.37	9.37	14.57	3.15	11.21
Mfeat-zer	4.78	15.32	2.04	24.31	29.49	3.09	31.21	35.15	7.37	32.32
Wine	0.55	0.42	0.57	0.58	0.46	0.36	0.82	0.93	0.42	0.58
Yeast	0.93	3.47	0.78	1.15	5.20	1.86	6.83	10.43	3.01	7.55

need additional time to form a new negative subset. However, this additional time is negligible, which is consistent with previous observations in [6]. For those oversampling approaches, including *OSBagg, SMOTEADA, AdaSyn*, and our *GIREnOS*, the size of final training data is around $2N_-$, which leads to a higher computational cost. Similarly, *OSBagg* is the fastest approach among these four oversampling-based approaches, since additional time is required to form new positive samples. From Table 9, it can be seen that our proposed two approaches *GIREnUS* and *GIREnOS* usually need more additional time to form a new data set than other corresponding approaches that have the same type. This is true because our approaches consider the change of difficulty of each data sample when one data sample is removed for undersampling or when one artificial data sample is generated for oversampling. However, we notice that this additional time is also negligible for all data sets, as shown in Table 9. The reason behind this is that we use the KNN graph to update the probability distributions in Eqs. (15) and (18). Instead of recalculating the nearest neighbors of all data samples when a new data sample is added or removed, we only maintain this KNN graph, which can greatly speed up the formation of new training data sets in our *GIREnUS* and *GIREnOS* approaches.

6. Conclusions

In this paper, we presented two novel adaptive ensemble learning approaches: GIREnUS and GIREnOS, for imbalanced data classification problems. For a given imbalanced data set, the proposed approaches adaptively split it into multiple subproblems, in each of which a balanced data set is also adaptively formed using an either undersampling or oversampling scheme, leading to GIREnUS and GIREnOS, respectively. Instead of using sample size ratio to measure class imbalance, a novel measurement, termed GIR, based on the intra-class coherence metric, is proposed. The new GIR metric considers the imbalance of class-wise distribution and offers us a new insight to the concept of “imbalance” in imbalanced learning. Both boosting and bagging schemes are used in our proposed approaches to reduce the bias and achieve a stronger generalization. The superior learning performance on real-life data sets demonstrates the effectiveness of our proposed methods and further illustrates wide potential applications on data mining.

In the future, we would like to extend our imbalance measure and two imbalanced learning methods for multiple classes classification problems. Also, we plan to study our methods for online class imbalance learning with data streams, particularly with concept drifts, which requires to adaptively update the learning rules for prediction.

Acknowledgment

This research was partially supported by National Science Foundation (NSF) under grant ECCS 1053717 and CCF 1439011, and the Army Research Office under grant W911NF-12-1-0378.

References

[1] H. He, E. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.

[2] M. Elter, R. Schulz-Wendtland, T. Wittenberg, The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process, Med. Phys. 34 (11) (2007) 4164–4172.

[3] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J.Artif.Intell.Res. (2002) 321–357.

[4] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: Advances in Intelligent Computing, 2005, pp. 878–887.

[5] H. He, Y. Bai, E. Garcia, S. Li, et al., ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Neural Networks, 2008, pp. 1322–1328.

[6] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern.Part B 39 (2) (2009) 539–550.

[7] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, in: Conference on AI in Medicine in Europe: Artificial Intelligence Medicine, 2001, pp. 63–66.

[8] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, Comput. Intell. 20 (1) (2004) 18–36.

[9] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 20–29.

[10] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: improving classification performance when training data is imbalanced, in: International Workshop on Computer Science and Engineering, 2, 2009, pp. 13–17.

[11] B. Tang, H. He, KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning, IEEE Congress on Evolutionary Computation (CEC), 2015.

[12] D. Mease, A. Wyner, A. Buja, Cost-weighted boosting with jittering and over/under-sampling: Jous-boost, J. Mach. Learn. Res. 8 (2007a) 409–439.

[13] D. Mease, A.J. Wyner, A. Buja, Boosted classification trees and class probability/quantile estimation, J. Mach. Learn. Res. 8 (2007b) 409–439.

[14] A.B. Owen, Infinitely imbalanced logistic regression, J. Mach. Learn. Res. 8 (2007) 761–773.

[15] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.

[16] R.E. Schapire, The strength of weak learnability, Mach. Learn. 5 (2) (1990) 197–227.

[17] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 148–156.

[18] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in: Knowledge Discovery in Databases: PKDD 2003, Springer, 2003, pp. 107–119.

[19] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, S.Y. Philip, et al., Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1–37.

[20] R. Barandela, R.M. Valdovinos, J.S. Sánchez, New applications of ensembles of classifiers, Pattern Anal. Appl. 6 (3) (2003) 245–256.

[21] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: IEEE Symposium on Computational Intelligence and Data Mining, 2009, pp. 324–331.

[22] C. Elkan, The foundations of cost-sensitive learning, in: International Joint Conference on Artifical Intelligence, 2001, pp. 973–978.

[23] K.M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Trans. Knowl. Data Eng. 14 (3) (2002) 659–665.

[24] P. Domingos, Metacost: a general method for making classifiers cost-sensitive, in: International Conference on Knowledge Discovery and Data Mining, 1999, pp. 155–164.

[25] W. Fan, S.J. Stolfo, J. Zhang, P.K. Chan, AdaCost: misclassification cost-sensitive boosting, in: International Conference on Machine Learning, 1999, pp. 97–105.

[26] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Trans. Knowl. Data Eng. 18 (1) (2006) 63–77.

[27] M.A. Maloof, Learning when data sets are imbalanced and when costs are unequal and unknown, in: International Conference Machine Learning Workshop on Learning from Imbalanced Data Sets II, 2001, pp. 973–978.

[28] R. Kohavi, D.H. Wolpert, et al., Bias plus variance decomposition for zero-one loss functions, in: International Conference on Machine Learning, 96, 1996, pp. 275–283.

[29] J. Gama, Iterative bayes, Theor. Comput. Sci. 292 (2) (2003) 417–430.

[30] G. Wu, E.Y. Chang, Kba: kernel boundary alignment considering imbalanced data distribution, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 786–795.

[31] F. Vilariño, P. Spyridonos, J. Vitrià, P. Radeva, Experiments with Svm and Stratified Sampling with an Imbalanced Problem: Detection of Intestinal Contractions, in: Pattern Recognition and Image Analysis, 2005, pp. 783–791.

[32] P. Kang, S. Cho, EUS SVMs: ensemble of under-sampled svms for data imbalance problems, in: Neural Information Processing, 2006, pp. 837–846.

[33] Y. Liu, A. An, X. Huang, Boosting prediction accuracy on imbalanced datasets with svm ensembles, in: Advances in Knowledge Discovery and Data Mining, 2006, pp. 107–118.

[34] J.-R. Cano, Analysis of data complexity measures for classification, Expert Syst. Appl. 40 (12) (2013) 4820–4831.

[35] J. Luengo, A. Fernández, S. García, F. Herrera, Addressing data complexity for imbalanced data sets: analysis of smote-based oversampling and evolutionary undersampling, Soft comput. 15 (10) (2011) 1909–1936.

[36] C.G. Weng, J. Poon, A data complexity analysis on imbalanced datasets and an alternative imbalance recovering strategy, in: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp. 270–276.

[37] S. Mahmoudi, P. Moradi, F. Akhlaghian, R. Moradi, Diversity and separable metrics in over-sampling technique for imbalanced data classification, in: 2014 4th International eConference on Computer and Knowledge Engineering (ICCKE), 2014, pp. 152–158.

[38] N. Anwar, G. Jones, S. Ganesh, Measurement of data complexity for classification problems with unbalanced data, Stat. Anal. Data Mining 7 (3) (2014) 194–211.

[39] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.

[40] B. Tang, H. He, ENN: extended nearest neighbor method for pattern recognition [research frontier], IEEE Comput. Intell. Mag. 10 (3) (2015) 52–60.

[41] M.F. Schilling, Multivariate two-sample tests based on nearest neighbors, J. Am. Stat. Assoc. 81 (395) (1986) 799–806.

[42] L. Chen, W.W. Dou, Z. Qiao, Ensemble subsampling for imbalanced multivariate two-sample tests, J. Am. Stat. Assoc. 108 (504) (2013) 1308–1323.

[43] M. Lichman, UCI Machine Learning Repository, School of Information and Computer Science, Irvine, CA: University of California, 2013. Web: http://archive.ics.uci.edu/ml/.

[44] S. Chen, H. He, E. Garcia, Ramoboost: ranked minority oversampling in boosting, IEEE Trans. Neural Netw. 21 (10) (2010) 1624–1642.

[45] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC press, 1984.

[46] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[47] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, L.I. Kuncheva, Random balance: ensembles of variable priors classifiers for imbalanced data, Knowl. Based Syst. 85 (2015) 96–111.

Bo Tang is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS. His current research interests include statistical machine learning, computational intelligence, computer vision, and robotics.

Haibo He is currently the Robert Haas Endowed Professor of Electrical Engineering with the University of Rhode Island, Kingston, RI, USA. His current research interests include machine learning, cyber-physical systems, computational intelligence, hardware design for machine intelligence, and various applications.