AMDO: An Over-Sampling Technique for Multi-Class Imbalanced Problems

Xuebing Yang, Qiuming Kuang, Wensheng Zhang, and Guoping Zhang

Abstract—Multi-class imbalanced problems have attracted growing attention from the real-world classification tasks in engineering. The underlying skewed distribution of multiple classes poses difficulties for learning algorithms, which becomes more challenging when considering overlapping between classes, lack of representative data, and mixed-type data. In this work, we address this problem in a data-oriented way. Motivated by a recently proposed over-sampling technique designed for numeric data sets, Mahalanobis Distance-based Over-sampling (MDO), we use this technique to capture the covariance structure of the minority class and to generate synthetic samples along the probability contours for learning algorithms. Based on MDO, we further improve the over-sampling strategy and generalize it for mixed-type data sets. The established technique, Adaptive Mahalanobis Distance-based Over-sampling (AMDO), introduces GSVD (Generalized Singular Value Decomposition) for mixed-type data, develops a partially balanced resampling scheme and optimizes the sample synthesis. Theoretical analysis is conducted to demonstrate the reasonability of AMDO. Extensive experimental testing is performed on 15 multi-class imbalanced benchmarks and two data sets for precipitation phase recognition in comparison with several state-of-the-art multi-class imbalanced learning methods. The results validate the effectiveness and robustness of our proposal.

Index Terms—Multi-class imbalanced problems, over-sampling, MDO, mixed-type data

1 Introduction

The imbalanced data originate from a variety of real-world applications, such as activity recognition [1], disease diagnosis [2] and fraud detection [3]. The skewed class distribution of imbalanced data sets causes performance degradation in many conventional machine learning algorithms. How to learn from imbalanced data sets has been defined as a challenge for the data mining research community [4]. The purpose of imbalanced learning is to provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class [5]. During the past two decades researchers have made great efforts to tackle this problem. For a comprehensive review of imbalanced learning, please refer to [6].

In this work we focus on multi-class imbalanced problems, which appear frequently but are not as well-developed as their binary counterpart. In the scenario of binary classification, it is straightforward to balance the class distribution using resampling techniques or to shift classifiers towards the minority class. However, the situation becomes more complicated for multi-class classification [7]: 1) the relations among classes are no longer obvious; 2) the boundaries among the classes may overlap. Thus, for our problems, the methods designed for the binary case may not be directly applicable or may suffer from lower performance [8].

An intuitive and widely used strategy to combat multi-class imbalanced problems is to apply class decomposition to reduce the problem to a set of binary subproblems. Two class decomposition strategies are common: one-versus-one (OVO) [9] and one-versus-all (OVA) [10]. Fernández et al. [40] developed an experimental study and verified the good behavior of OVO and OVA with resampling techniques/cost-sensitive learning. Various researches have worked on resampling techniques with class decomposition [11], [12], [13]. Combining OVO/OVA with ensemble learning has also shown promising results in recent years [14], [15].

By contrast, Wang and Yao [42] considered decomposition unnecessary and suggested direct learning from the entire data set for multi-class imbalanced problems. Without class decomposition, a specific design is required to modify the existing methods. At the data level, Lin et al. [16] and Fernández-Navarro et al. [39] utilized dynamic over-sampling to finely tune neural networks. At the algorithm level, several learning algorithms have been generalized to solve this problem, such as support vector machines [17], [18] and extreme learning machines [19], [20].

A novel over-sampling technique, Mahalanobis Distance-based Over-sampling (MDO), was recently proposed by Abdi and Hashemi [21]. This approach, which is a data level solution, generates synthetic samples for minority classes without class decomposition. In contrast to the well-known Synthetic Minority Over-sampling TEchnique (SMOTE) [22], the samples obtained by MDO maintain the same *Mahalanobis distance* [23] from the corresponding class mean. Experimental research and theoretical analysis in [21] confirmed

- *X. Yang, Q. Kuang, and W. Zhang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and the University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: {yangxuebing2013, kuangqiuming2014}@ia.ac.cn, zhangwenshengia@hotmail.com.*
- *G. Zhang is with the Joint Laboratory of Meteorological Data and Machine Learning, Public Meteorological Service Center of CMA, Beijing 100081, China. E-mail: zhanggp@cma.gov.cn.*

Manuscript received 28 Mar. 2017; revised 24 July 2017; accepted 3 Oct. 2017. Date of publication 11 Oct. 2017; date of current version 3 Aug. 2018.
(Corresponding author: Wensheng Zhang.)
Recommended for acceptance by J. Xu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2761347

that MDO has great potential even for multi-class imbalanced problems with overlap between classes.

Unfortunately, the use of MDO is limited because MDO is only applicable to data sets with numeric attributes. In real-world data mining problems, the data sets with mixed-type attributes are extremely common. For example, the data set for precipitation phase recognition is a multi-class imbalanced data set with overlap between classes. It consists of multiple meteorological observations (numeric) and climatic conditions (nominal). For such cases, MDO should be adapted to address mixed-type attributes. Moreover, two drawbacks of MDO are often found in practice: 1) MDO generates excessive synthetic samples for the minority classes, which may jeopardize the accuracy of the majority class; 2) MDO may generate unrealistic samples, which can make the sampling process less computationally efficient. Improved sampling strategies are needed to improve the effectiveness and efficiency of this technique.

In this paper we extend Abdi and Hashemi's [21] study and propose Adaptive Mahalanobis Distance-based Over-sampling (AMDO). The main contributions of this paper are summarized as follows:

- We adapt MDO to effectively handle multi-class imbalanced data sets with mixed-type attributes.
- We improve the performance of MDO by introducing a partially balanced resampling scheme and developing a new strategy to obtain adaptive samples.
- We conduct theoretical analysis for the necessity and relatively low computational complexity of AMDO.
- We conduct an extensive evaluation of AMDO on several numeric and nominal/mixed-type problems. The results confirm the advantages of our proposed technique and demonstrate the potential of AMDO in multi-class imbalanced problems.

The rest of this paper is organized as follows. Section 2 introduces related works, including a brief review of MDO. In Section 3 we describe our proposed approach in detail and analyze its computational complexity. Experimental study, including comparison of the results and analyses, are presented in Section 4. Finally, we conclude the paper and present our future work in Section 5.

2 Related Work

As our proposed method is a type of over-sampling technique, we first briefly review the resampling technique in this section. Next, related studies using resampling for multi-class imbalanced problems are reviewed. Then, we introduce our inspiration, MDO, and describe the motivation for our extensions and improvements to MDO.

2.1 Resampling Technique

Resampling techniques are applied directly to balance skewed distributions. They are versatile because their use is independent of the selected classifiers [40]. In general, resampling methods fall into three groups: over-sampling, under-sampling and hybrid.

The simplest resampling techniques involve duplicating the minority samples and/or eliminating some of the majority samples, which may cause over-fitting or a loss of useful information. Deciding which samples should be eliminated and which samples should be duplicated requires further investigation [24].

Another widely-used approach to balance the class distribution is SMOTE [22]. The nature of SMOTE is similar to interpolation. SMOTE selects one of the K-nearest neighbours of a minority sample $\mathbf{x}_i$ and calculates the difference between them; then, the obtained difference is multiplied by a random number in the range $[0, 1]$ and is added to $\mathbf{x}_i$ to generate a synthetic sample. However, SMOTE may increase the overlap between different classes and worsen the decision boundaries, especially in multi-class cases [25]. A considerable number of variations have been proposed to overcome the drawbacks of SMOTE, such as Safe-Level-SMOTE [26], Borderline-SMOTE [27], ADASYN [28], and SMOTE + ENN [29].

SMOTE is not the only way to generate synthetic samples. Chetchotsak et al. [30] proposed GRSOM to create new data using a self-organizing map. Das et al. [31] used the joint probability distribution of data attributes and Gibbs sampling to generate new minority class samples. In contrast to SMOTE, these sampling strategies show that taking the individual properties of classes and their mutual relations into account has potential for multi-class imbalanced problems.

2.2 Utilizing Resampling Techniques for Multi-Class Imbalanced Problems

The existing methods mostly use resampling techniques in the framework of class decomposition, namely, OVO/OVA. Liao [11] adopted OVA to conduct multi-class classification of weld flaws with 22 over-sampling and under-sampling methods. Zhao et al. [32] used OVA and a combination of under-sampling and SMOTE for protein classification. However, the drawbacks of class decomposition are that the training process may suffer from unacceptable time cost in OVO and the imbalanced situation may worsen in OVA.

Sáez et al. [48] applied SMOTE according to the class and type of samples and highlighted the importance of the analysis of sample difficulty. Fernández-Navarro et al. [39] proposed that SMOTE should be applied for most imbalanced cases and developed a special mechanism to obtain a partially balanced class distribution through SMOTE. Then, they optimized the radial basis function neural networks through a memetic algorithm. Similarly, Lin et al. [16] proposed a dynamic sampling procedure, DyS, to train multilayer perceptrons (MLP). They iteratively estimated the probability of samples being selected for training MLP to make the final MLP suitable for multi-class imbalanced classification. Wang and Yao [42] used random over-sampling for their proposed AdaBoost.NC to make the method not ignore minority classes.

Although not yet a popular approach, Abdi and Hashemi [21] attempted to directly apply resampling. They developed a technique, MDO, to generate synthetic samples and obtained classifiers that performed well in multi-class imbalanced scenarios. This technique is reviewed below.

2.3 Review: Mahalanobis Distance-Based Over-Sampling Technique

To combat multi-class problems, MDO generates synthetic samples for each minority class.[1] It takes the *Principal Component* (PC) space into consideration and obtains new samples in this space. As a result, synthetic samples are shifted toward the variation of the corresponding class, preserving

[1] In this paper, imbalanced multiple classes refers to multiple minority classes and a single majority class.

the covariance structure of the data in the minority classes. The algorithm is described in Algorithm 1.

Algorithm 1. MDO

```
    Input: Training data set $\mathbf{S}$, parameters $K1$, $K2$
    Output: A new training data set $\mathbf{S}_*$
 1: Obtain $c$ and $n_{maj}$ with respect to $\mathbf{S}$;
 2: for $s = 1 : c - 1$ do
 3:     Obtain $\mathbf{X}_s \in \mathbb{R}^{n_s \times p_1}$;
 4:     for $i = 1 : n_s$ do
 5:         Compute the $K2$ nearest neighbours of $\mathbf{X}_s(i)$ using
            Euclidean distance as the metric;
 6:         Obtain $num(i)$;
 7:         Assign weights $\frac{num(i)}{K2}$ for $\mathbf{X}_s(i)$;
 8:         if $num(i) < K1$ then
 9:             Remove $\mathbf{X}_s(i)$ from $\mathbf{X}_s$;
10:         end
11:     end
12:     $\mathbf{X}_s \in \mathbb{R}^{n'_s \times p_1}$, $\boldsymbol{\mu}_s = \frac{1}{n'_s}\sum_{i=1}^{n'_s} \mathbf{X}_s(i)$;
13:     $\mathbf{X}_s = \mathbf{X}_s - \boldsymbol{\mu}_s$;
14:     Compute $\mathbf{Q}$ and $\Sigma_s$ via eigenvalue decomposition:
        $\mathbf{Q}\Sigma_s\mathbf{Q}^{\mathrm{T}} = \mathrm{cov}(\mathbf{X}_s)$;
15:     Obtain the vector of coefficients by
        $(\varsigma_1(\mathbf{X}_s), \ldots, \varsigma_{p_1}(\mathbf{X}_s)) = \mathrm{diag}(\Sigma_s)$;
16:     if $n_{maj} - n'_s > 0$ then
17:         for $j = 1 : n_{maj} - n'_s$ do
18:             Choose a random sample $\mathbf{x}_s$ from $\mathbf{X}_s\mathbf{Q}$ according
                to the weights;
19:             Compute $\alpha$ by solving (2) with $\mathbf{x}_s$ and
                $(\varsigma_1(\mathbf{X}_s), \ldots, \varsigma_{p_1}(\mathbf{X}_s))$;
20:             Let $\mathbf{x}$ be the synthetic sample, $\mathbf{x} = (x_1, \ldots, x_{p_1})$;
21:             for $k = 1 : p_1 - 1$ do
22:                 $x_k$ = choose a random number from
                    $\left[-\sqrt{\alpha \cdot \varsigma_i(\mathbf{X}_s)}, \sqrt{\alpha \cdot \varsigma_i(\mathbf{X}_s)}\right]$;
23:             end
24:             Compute $x_{p_1}$ by solving (2) with $(x_1, \ldots, x_{p_1 - 1})$
                and $(\varsigma_1(\mathbf{X}_s), \ldots, \varsigma_{p_1}(\mathbf{X}_s))$, and randomly choose
                one from two solutions;
25:             Add $\mathbf{x}$ to $\mathbf{X}_{s+}$;
26:         end
27:     end
28:     $\mathbf{X}_{s+} = \mathbf{X}_{s+}\mathbf{Q}^{\mathrm{T}} + \boldsymbol{\mu}_s$;
29: end
30: Add labels for the synthetic samples of each minority class
    and obtain $\mathbf{S}_+$;
31: $\mathbf{S}_* = \mathbf{S} + \mathbf{S}_+$;
32: Return A new training data set $\mathbf{S}_*$.
```

Let us consider a data set of c classes and p_1 numeric attributes. For a given minority class $\mathbf{X}_s \in \mathbb{R}^{n_s \times p_1}$, in steps 4-11 of Algorithm 1 MDO first selects minority class candidates that are located in the dense areas of $\mathbf{X}_s$ and assigns weights to them. MDO then computes the $K2$ nearest neighbours from $\mathbf{X}_s$. Among the $K2$ neighbours, the number of neighbours belonging to $\mathbf{X}_s$ is denoted as num. For the ith data point, if $num(i)$ is larger than a threshold $K1$, MDO selects it as a candidate and assigns it a selection weight. This process can ensure that new synthetic samples are generated along the probability contours of the samples with more neighbours in the same class. Then, in steps 12-13 the considered n'_s samples are normalized to have zero mean. Thus, in the original data space samples with the same Euclidean distance from the class mean are located in several circle contours with the origin as the center. By using a parameter α, the circle contour of sample $(x_1, \ldots, x_d)$ is

$$\frac{x_1^2}{\alpha} + \frac{x_2^2}{\alpha} + \cdots + \frac{x_i^2}{\alpha} + \cdots + \frac{x_d^2}{\alpha} = 1. \tag{1}$$

To achieve the goal of synthesizing samples that preserve the covariance structure of $\mathbf{X}_s$, a vector of coefficients $(V_1, \ldots, V_d)$ is required to make the components with high variability receive higher weights. Instead of circle contours, ellipse contours based on the Mahalanobis distance are required

$$\frac{x_1^2}{\alpha \cdot V_1} + \frac{x_2^2}{\alpha \cdot V_2} + \cdots + \frac{x_i^2}{\alpha \cdot V_i} + \cdots + \frac{x_d^2}{\alpha \cdot V_d} = 1. \tag{2}$$

In steps 14-15 $\mathbf{Q}\Sigma_s\mathbf{Q}^{\mathrm{T}} = \mathrm{cov}(\mathbf{X}_s)$ is the *eigenvalue decomposition* of $\mathbf{X}_s$. In this way the PC space ($\mathbf{X}_s\mathbf{Q}$) is constructed. In this space, data are uncorrelated and $(\varsigma_1(\mathbf{X}_s), \ldots, \varsigma_{p_1}(\mathbf{X}_s))$ can be regarded as the vector of coefficients to describe the variance of each dimension, where $\varsigma_i(\mathbf{X}_s)$ is the ith largest coefficient. Then, in steps 16-27 $\mathbf{x}_s$ is selected from the PC space through weighted sampling and a synthetic sample $\mathbf{x}$ is generated along the ellipse contour that $\mathbf{x}_s$ belongs to by solving (2) with $(\varsigma_1(\mathbf{X}_s), \ldots, \varsigma_{p_1}(\mathbf{X}_s))$. This procedure is repeated $n_{maj} - n'_s$ times, where n_{maj} is the number of samples in the majority class. Finally, in step 28 the synthetic samples are transformed to the original space and added to the original data set.

MDO was shown to be suitable for multi-class imbalanced problems in [21]. However, as the authors point out, the technique cannot handle data with nominal/mixed-type attributes. Furthermore, there are risks of over-generation and generating unrealistic samples for MDO. The above viewpoints motivate our following extensions and improvements of MDO.

3 The Proposed Approach

In this section we describe the details of our proposed technique, AMDO. Fig. 1 shows the AMDO flowchart. The framework of over-sampling remains the same as that of MDO: 1) suitable samples are chosen and assigned weights; 2) samples are transformed to the PC space; and 3) new synthetic samples are generated. Our work extends and improves MDO in three aspects. We first adapt MDO to handle mixed-type attributes by using *HVDM* [33] and *GSVD* [35]. Next, to reduce the risk of over-generation, we propose a partially balanced resampling scheme to generate an adaptive class distribution for learning algorithms. Finally, for the sampling strategy in PC space, we provide theoretical analysis about why unrealistic samples can be generated by MDO in most cases and develop a new strategy to obtain adaptive synthetic samples. The pseudocode of AMDO is presented in Algorithm 2.

3.1 Handling Mixed-Type Data

From numeric attributes to mixed-type attributes, two issues must be considered. For mixed-type data, Euclidean distance cannot be used to select minority class candidates (Algorithm 1, step 5). When constructing the PC space, eigenvalue decomposition also cannot be used for mixed-type data (Algorithm 1, steps 12-15). In the following we address these issues by applying HVDM as a metric to

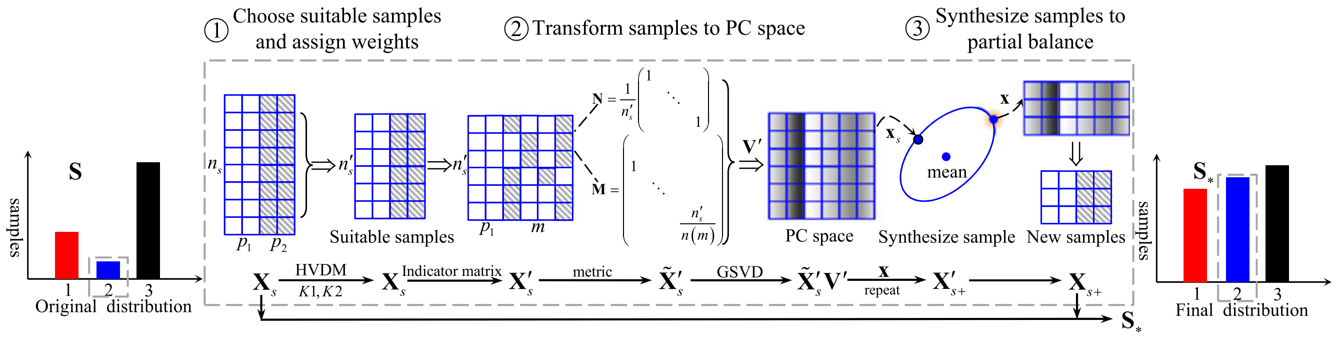

① Choose suitable samples and assign weights
② Transform samples to PC space
③ Synthesize samples to partial balance
S
samples
1 2 3
Original distribution
n_s
p_1 p_2
n_s'
Suitable samples
n_s'
p_1 m
$\mathbf{N}=\frac{1}{n_s'}$
$\mathbf{M}=$
$\frac{n_s'}{n(m)}$
$\mathbf{V}'$
PC space
$\mathbf{x}_s$
mean
$\mathbf{x}$
Synthesize sample
New samples
$\mathbf{S}_*$
Final distribution
$\mathbf{X}_s$ HVDM $K1,K2$ $\mathbf{X}_s$ Indicator matrix $\mathbf{X}_s'$ metric $\tilde{\mathbf{X}}_s'$ GSVD $\tilde{\mathbf{X}}_s'\mathbf{V}'$ $\mathbf{x}$ repeat $\mathbf{X}_{s+}'$ $\mathbf{X}_{s+}$
$\mathbf{S}_*$

Fig. 1. The flowchart of AMDO.

search neighbours and by using GSVD to transform the mixed-type data to the PC space.

As mixed-type data include both numeric and nominal attributes, we use HVDM as the distance metric. Given two samples $\mathbf{x}$ and $\mathbf{z}$, the HVDM distance between them can be computed according to [33]. HVDM provides a way to approximately scale the two types of measurements into the same range and makes each attribute have a similar influence on the overall distance between $\mathbf{x}$ and $\mathbf{z}$. Thus, the $K2$ nearest neighbours in HVDM can be easily computed and the minority class candidates can be obtained.

In addition to eigenvalue decomposition, SVD is an alternative method to transform the data points in the original space to PC space[34]. For the selected samples of one class $\mathbf{X}_s$, $\mathbf{X}_s$ also needs to be normalized before SVD. Let $\mathbf{X}_s = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$ be the SVD of $\mathbf{X}_s$; then, $\mathbf{X}_s$ is transformed to the PC space by $\mathbf{X}_s\mathbf{V}$. For mixed-type data, we can generalize matrix SVD to achieve the corresponding PC space using the GSVD [35] procedure.

After the minority class candidates are obtained, $\mathbf{X}_s$ is updated to be a $n_s' \times (p_1 + p_2)$ data matrix, where the first p_1 columns correspond to the p_1 numeric attributes of the n_s' samples and the subsequent p_2 columns correspond to the p_2 nominal attributes. For each nominal attribute, we represent the different levels as discrete numeric values. As the levels of each nominal attribute are finite, we can construct an $n_s' \times m$ indicator matrix for the p_2 nominal attributes, where m denotes the total number of levels. In this way a numeric version of $\mathbf{X}_s$, denoted by $\mathbf{X}_s'$, can be constructed with dimensions $n_s' \times (p_1 + m)$. Similarly, $\mathbf{X}_s'$ is normalized (see Algorithm 2, steps 14-15) to ensure that the origin of the PC space is the center of every ellipse contour. As $\mathbf{X}_s$ has been changed to $\mathbf{X}_s'$, metrics are needed to introduce weights to the rows and columns of $\mathbf{X}_s'$ to construct the appropriate PC space. GSVD can provide a matrix decomposition of $\mathbf{X}_s'$ using the two positive definite square matrices $\mathbf{N}$ and $\mathbf{M}$, where $\mathbf{N}$ is a metric on $\mathbb{R}^n$ and $\mathbf{M}$ is a metric on $\mathbb{R}^{p_1+m}$. The GSVD of $\mathbf{X}_s'$ can be obtained by performing the standard SVD of the matrix $\tilde{\mathbf{X}}_s' = \mathbf{N}^{1/2}\mathbf{X}_s'\mathbf{M}^{1/2}$, that is, $\tilde{\mathbf{X}}_s' = \mathbf{U}'\Sigma'\mathbf{V}'^{\mathrm{T}}$. The metrics of $\mathbf{N}$ and $\mathbf{M}$ are as follows:

$$\begin{cases} \mathbf{N} = \frac{1}{n_s'}\mathbb{I}_n \\ \mathbf{M} = \mathrm{diag}(\underbrace{1,\ldots,1}_{p_1}, \underbrace{n_s'/n(1),\ldots,n_s'/n(m)}_{m}) \end{cases}, \qquad (3)$$

where $n(i)$ denotes the number of samples that belong to the ith level.

After GSVD of $\mathbf{X}_s'$, the matrix of data points in the PC space is $\tilde{\mathbf{X}}_s'\mathbf{V}'$. Synthetic samples are generated in $\tilde{\mathbf{X}}_s'\mathbf{V}'$. As $\mathbf{V}'$ is orthogonal and $n_s' \gg (p_1 + m)$ in this study, the vector of coefficients $(\varsigma_1(\mathbf{X}_s'), \ldots, \varsigma_{p_1+m}(\mathbf{X}_s'))$ describing the covariance structure of $\mathbf{X}_s'$ can be obtained by computing the covariance of $\tilde{\mathbf{X}}_s'\mathbf{V}'$

$$\begin{bmatrix} \varsigma_1(\mathbf{X}_s') & & \\ & \ddots & \\ & & \varsigma_{p_1+m}(\mathbf{X}_s') \end{bmatrix} = \mathrm{cov}(\tilde{\mathbf{X}}_s'\mathbf{V}'). \qquad (4)$$

Thus, for any data points $\mathbf{x}_s$ in PC space, the corresponding ellipse contour can be obtained with $(\varsigma_1(\mathbf{X}_s'), \ldots, \varsigma_{p_1+m}(\mathbf{X}_s'))$ according to (2). The new sample is then generated along the same ellipse contour of $\mathbf{x}_s$.

3.2 Partially Balanced Resampling Scheme

The main aim of over-sampling is to balance the class distribution. However, there is no consensus in the research community on which type of distribution should be achieved for imbalanced data sets. As mentioned before, MDO generates $n_{maj} - n_s'$ synthetic samples for $\mathbf{X}_s$. Usually, $n_s' < n_s$ for $\mathbf{X}_s$, which makes the final number of samples in $\mathbf{X}_s$ larger than n_{maj}. As a result, MDO imposes a handicap on the classifiers due to the risk of over-generation/over-fitting.

In this study we use the imbalance ratio (IR)[36] to identify whether a data set is imbalanced. IR is computed as the proportion of n_{maj} to n_{min}, where n_{min} is the smallest number of class samples among the minority classes. A data set is considered to be imbalanced if IR $>$ 1.5, as suggested by [37] and [38]. The purpose of our proposed partially balanced resampling scheme is to adjust the IR of the current data set to be less than 1.5 (IR $\leq$ 1.5).

We now consider a data set with c classes of distribution $\mathcal{D}$. Initially, we can obtain n_{min} and the number of total samples in the data set, N. Inspired by the dynamic over-sampling approach reported in [39], we develop our proposal, which is detailed in Algorithm 3. The procedure is applied for several iterations (we explain how to determine the maximum number of iterations later). In each iteration, we first obtain the current class distribution $\mathcal{D}_*$. According to $\mathcal{D}_*$, the current number of total samples, N_* and the number of samples in the current minimum class, n_{cmin}^{curr} can be obtained. Next, we take the current minimum ratio p_{min} into consideration, where $p_{min} = n_{min}^{curr}/N_*$ denotes the minimum of the current prior probabilities. If p_{min} is less than a threshold θ, the current class distribution is considered to be imbalanced. Then, the current class with the minimum size is selected, and the same number of samples that it had in the original data set is added.

Algorithm 2. AMDO

```
Input: Training data set $\mathbf{S}$, attribute indicator $\mathbb{A}$, parameters $K1, K2$
Output: A new training data set $\mathbf{S}_*$
Obtain $c, p_1, p_2, m$ and $\mathcal{D}$ with respect to $\mathbf{S}$ and $\mathbb{A}$;
Obtain $Orate(1), \ldots, Orate(c-1)$ by using Algorithm 3;
for $s = 1 : c-1$ do
    Obtain $\mathbf{X}_s \in \mathbb{R}^{n_s \times (p_1+p_2)}$;
    for $i = 1 : n_s$ do
        Compute the $K2$ nearest neighbours of $\mathbf{X}_s(i)$ using $HVDM$ as the metric;
        Obtain $num(i)$;
        Assign weights $\frac{num(i)}{K2}$ for $\mathbf{X}_s(i)$;
        if $num(i) < K1$ then
            Remove $\mathbf{X}_s(i)$ from $\mathbf{X}_s$;
        end
    end
    Transform the last $p_2$ columns of $\mathbf{X}_s$ into $m$ columns and obtain $\mathbf{X}'_s \in \mathbb{R}^{n'_s \times (p_1+m)}$;
    $\boldsymbol{\mu}_s = \frac{1}{n'_s}\sum_{i=1}^{n'_s}\mathbf{X}'_s(i)$, $\boldsymbol{\sigma}_s = \sqrt{\frac{1}{n'_s}\sum_{i=1}^{n'_s}(\mathbf{X}'_s(i)-\boldsymbol{\mu}_s)^2}$;
    $\mathbf{X}'_s = \frac{\mathbf{X}'_s-\boldsymbol{\mu}_s}{\boldsymbol{\sigma}_s}$;
    Compute $\mathbf{N}$ and $\mathbf{M}$ via (3);
    $\tilde{\mathbf{X}}'_s = \mathbf{N}^{1/2}\mathbf{X}'_s\mathbf{M}^{1/2}$;
    Compute $\mathbf{V}'$ via $GSVD$: $\tilde{\mathbf{X}}'_s = \mathbf{U}'\Sigma'\mathbf{V}'^{\mathrm{T}}$;
    Obtain the vector of coefficients using (4);
    if $Orate(s) > 0$ then
        for $j = 1 : Orate(s)$ do
            Choose a random sample $\mathbf{x}_s$ from $\tilde{\mathbf{X}}'_s\mathbf{V}'$ according to the weights;
            Compute $\alpha$ by solving (2) with $\mathbf{x}_s$ and $(\varsigma_1(\mathbf{X}'_s), \ldots, \varsigma_{p_1+m}(\mathbf{X}'_s))$;
            Let $\mathbf{x}$ be the synthetic sample, $\mathbf{x} = (x_1, \ldots, x_{p_1+m})$;
            Generate $p_1 + m$ positive random numbers $r_1, \ldots, r_{p_1+m}$ and make them sum to one;
            for $k = 1 : p_1 + m$ do
                $x_k$ = randomly choose one solution from solving $\frac{x_k^2}{\alpha\cdot\varsigma_k(\mathbf{X}'_s)} = r_k$;
            end
            Add $\mathbf{x}$ to $\mathbf{X}'_{s+}$;
        end
    end
    $\mathbf{X}'_{s+} = \boldsymbol{\sigma}_s(\mathbf{N}^{-1/2}\mathbf{X}'_{s+}\mathbf{V}^{\mathrm{T}}\mathbf{M}^{-1/2}) + \boldsymbol{\mu}_s$;
    Transform the last $m$ columns of $\mathbf{X}'_{s+}$ into $p_2$ columns and obtain $\mathbf{X}_{s+}$;
end
Add labels for the synthetic samples of each minority class and obtain $\mathbf{S}_+$;
$\mathbf{S}_* = \mathbf{S} + \mathbf{S}_+$;
Return A new training data set $\mathbf{S}_*$.
```

The corresponding θ is the stopping criterion to make IR ≤ 1.5 in the final data set

$$p_{min} > \theta. \tag{5}$$

In each iteration, n_{cmin}^{curr} changes, while n_{maj} is a constant. As $c \geq 3$ and IR > 1 in our study, we can rewrite (5) as

$$\theta < \frac{n_{maj}}{(n_{min}^{curr} + \sum_{i=1}^{c-2} n_i^{curr} + n_{maj})\mathrm{IR}}, \tag{6}$$

where $n_1^{curr}, \ldots, n_{c-2}^{curr}$ are the sizes of the other $c-2$ classes, with values between n_{min}^{curr} and n_{maj}. To ensure IR ≤ 1.5 in the final data set, we set θ as the lower bound of the right-hand formula in (6), that is

$$\frac{n_{maj}}{(n_{min}^{curr} + \sum_{i=1}^{c-2} n_i^{curr} + n_{maj})\mathrm{IR}} > \frac{1}{1+(c-1)\mathrm{IR}} \geq \frac{2}{3c-1}. \tag{7}$$

Hence, we set $\theta = \frac{2}{3c-1}$ for this study. Note that when focusing on imbalanced data sets with other IR constraints, θ should be adjusted accordingly.

Algorithm 3. Partially Balanced Resampling

```
Input: Number of classes $c$, class distribution $\mathcal{D}$
Output: Over-sampling rate for each minority class, $Orate(1), \ldots, Orate(c-1)$
Obtain $n_{min}$, $N$ via $\mathcal{D}$;
$maxit = \lceil\frac{n_{max}\cdot(c-1)}{n_{min}}\rceil$;
Initialize $\mathcal{D}_* = \mathcal{D}$, $N_* = N$;
for $i = 1 : maxit$ do
    Obtain the current minimum class $cmin$ and its current size $n_{cmin}^{curr}$ via $\mathcal{D}_*$;
    Compute the current minimum ratio $p_{min} = \frac{n_{cmin}^{curr}}{N_*}$;
    if $p_{min} \leq \frac{2}{3c-1}$ then
        $n_{cmin}^{curr} = n_{cmin}^{curr} + n_{cmin}$;
    end
    Update $\mathcal{D}_*$, $N_*$;
end
Compare $\mathcal{D}_*$ and $\mathcal{D}$ and obtain $Orate(1), \ldots, Orate(c-1)$;
Return $Orate(1), \ldots, Orate(c-1)$.
```

At the end of this section let us return to the maximum number of iterations for the resampling procedure. After each iteration the size of the current minimum class has changed. According the above analysis, a minority class will take at most $\lceil\frac{n_{maj}}{n_{min}}\rceil$ iterations to increase its size to be larger than the majority class. Consequently, for our multi-minority imbalanced problems, we set the maximum number of iterations to be $\lceil\frac{n_{maj}\cdot(c-1)}{n_{min}}\rceil$. It should be noted that the iterations will stop earlier in practice. Finally, we can obtain the over-sampling rate for each minority class, $Orate(1), \ldots, Orate(c-1)$, by comparing $\mathcal{D}_*$ and $\mathcal{D}$.

3.3 Adaptive Synthetic Sample

Similar to the original MDO procedure, our proposal generates synthetic samples in the PC space. The difference is in the way new sample $\mathbf{x}$ is obtained along the same ellipse contour of $\mathbf{x}_s$. We first introduce the following theorem and discuss why MDO usually generates unrealistic samples.

Theorem 1. *For a $d(d > 2)$ dimensional synthetic sample $\mathbf{x}$ of class s, when $i = 1, 2, \ldots, d-1$, let x_i randomly take a value from $\left[-\sqrt{\alpha\cdot\varsigma_i(\mathbf{X}_s)}, \sqrt{\alpha\cdot\varsigma_i(\mathbf{X}_s)}\right]$, where $\alpha\cdot\varsigma_i(\mathbf{X}_s) > 0$, for $i = 1, 2, \ldots, d$. Then, x_d is obtained by solving $\sum_i \frac{x_i^2}{\alpha\cdot\varsigma_i(\mathbf{X}_s)} = 1$. With probability $1 - \frac{1}{(d-1)!}$, x_d has an imaginary component.*

The proof can be found in the Appendix. Theorem 1 indicates that with probability $1 - \frac{1}{(d-1)!}$, the last dimension of the samples generated by MDO in PC space will have an imaginary component. As the transformation from PC space to the original space is a linear operation with real matrices, the new samples are unrealistic for data sets with real values. Note that this probability is relatively high for

TABLE 1
Summary Description of the Data Sets

Data Set	Size	Attr.	Cl.	Class Distribution	IR
Balance	625	4(4/-)	3	288/49/288	5.88
Hayes-roth	132	4(4/-)	3	51/51/30	1.70
New-thyroid	215	5(5/-)	3	150/35/30	5.00
Page-blocks	5,472	10(10/-)	5	4,913/329/28/87/115	175.46
Dermatology	358	34(34/-)	6	111/60/71/48/48/20	5.55
Breast-tissue	106	9(9/-)	6	21/15/18/16/14/22	1.57
User-knowledge-modelling (UKM)	403	5(5/-)	4	50/102/129/122	2.58
Vertebral-column	310	6(6/-)	3	60/150/100	2.50
Ecoli	327	7(7/-)	5	143/77/52/35/20	7.15
Pre2D	58,876	2(2/-)	3	40,374/2,112/16,390	19.12
Contraceptive	1,473	9(6/3)	3	629/333/511	1.89
Flare	1,066	11(-/11)	6	331/239/211/147/95/43	7.70
Thyroid	7,200	21(6/15)	3	166/368/6,666	40.16
Car	1,728	6(-/6)	4	1,210/384/69/65	18.62
Nursery	12,958	8(-/8)	4	4,320/328/4,266/4,044	13.17
Splice	3,190	60(-/60)	3	767/768/1,655	2.16
PreND	57,907	21(16/5)	3	39,762/2,090/16,055	19.02

Attr.: number of attributes(numeric/nominal), Cl.: number of classes.

most data sets (e.g., nearly 99.2 percent for $d = 5$). Thus, it is necessary to improve the method used to generate $\mathbf{x}$ in the PC space.

Given $\mathbf{x}_s$ and $(\varsigma_1(\mathbf{X}'_s), \ldots, \varsigma_{p_1}(\mathbf{X}'_s))$, the corresponding parameter α can be computed by solving (2). Then, given $(\varsigma_1(\mathbf{X}'_s), \ldots, \varsigma_{p_1}(\mathbf{X}'_s))$ and α, we need to determine each dimension of $(x_1, \ldots, x_d)$ to satisfy (2). Here, we first generate d (in this study, $d = p_1 + m$) positive random numbers $r_1, \ldots, r_d$ and make them sum to one. Thus, for the ith dimension we can randomly choose one solution (positive/negative) from solving $\frac{x_i^2}{\alpha \cdot \varsigma_i(\mathbf{X}_s)} = r_i$ as its value to generate adaptive synthetic samples.

3.4 Computational Complexity

Since AMDO generates synthetic samples for each minority class, we can first consider the computational complexity of over-sampling for one minority class. The computational bottleneck of the over-sampling procedure lies in obtaining candidates of $\mathbf{X}_s \in \mathbb{R}^{n_s \times (p_1+p_2)}$ and performing SVD of $\tilde{\mathbf{X}}'_s \in \mathbb{R}^{n'_s \times (p_1+m)}$. When searching for candidates, calculating the $K2$ nearest neighbours dominates the computation in each iteration. It takes $\mathcal{O}(n_s(p_1+p_2) + n_s\mathrm{log}(n_s))$ in each iteration, and $\mathcal{O}(n_s^2(p_1+p_2) + n_s^2\mathrm{log}(n_s))$ to obtain all candidates for $\mathbf{X}_s$. For SVD, as $n'_s \gg (p_1+m)$ in this study, it takes $\mathcal{O}(n'^2_s(p_1+m))$. Hence, the computational complexity of over-sampling for one minority class is

$$\mathcal{O}(n_s^2(p_1+p_2) + n_s^2\mathrm{log}(n_s)) + \mathcal{O}(n'^2_s(p_1+m)). \tag{8}$$

Because $p_2 \leq m$ and $n'_s \leq n_s$, (8) is actually

$$\mathcal{O}(n_s^2\mathrm{max}(\mathrm{log}(n_s), (p_1+m))). \tag{9}$$

For a data set with N samples, we have $\sum_s n_s^2 < N^2$ and $\sum_s n_s^2\mathrm{log}(n_s) < N^2\mathrm{log}(N)$. So, the computational complexity of Algorithm 2 is

$$\mathcal{O}(N^2\mathrm{max}(\mathrm{log}(N), (p_1+m))). \tag{10}$$

Note that (10) is simply an upper bound. For large-scale data sets, (10) can be reduced with some approximations. Combining the FLANN software package [53] with the LazySVD method [54] can reduce the complexity to $\mathcal{O}(N(p_1+m)\mathrm{max}(\mathrm{log}(N), \sqrt{p_1+m}))$.

4 Experimental Study

In this section experiments are performed with a double purpose. First, the aim is justification of our proposed technique compared with MDO and other multi-class imbalanced learning algorithms for numeric data sets. Second, the goal is to verify the performance of AMDO for mixed-type data sets.

We first give a description of the data sets, competitors, parameters and evaluation metrics for the performance evaluation. Then, the experimental results, statistical tests and analysis are presented in subsequent sections.

4.1 Data Sets

The proposed method is applied to 15 multi-class data sets from the UCI[2] and KEEL[3] repositories, of which 10 data sets are numeric and 5 data sets are nominal/mixed-type. Two additional data sets described in Section 4.1.1, which correspond to a real-world problem of precipitation phase recognition for the meteorological service, are included. In this work, as stated in Section 3.2, only data sets with IR > 1.5 are selected. Furthermore, to ensure fair comparison via cross-validation, we do not include data sets where the number of samples in the minimum class is less than 10.

Table 1 summarizes the characteristics of the selected data sets. We remove the samples with missing values (from *Dermatology*) from the data set. Furthermore, we eliminate the classes with fewer than 10 samples in *Ecoli* and *Nursery*.

The experiments are conducted using five-fold cross-validation with 10 independent runs, except for the *Pre2D* and *PreND* data sets.

4.1.1 *Description and Experimental Design of the "Pre2D" and "PreND" Data Sets*

The original data sets were provided by the Public Meteorological Service Center of the China Meteorological

[2] http://mlr.cs.umass.edu/ml/datasets.html
[3] http://sci2s.ugr.es/keel/datasets.php

Administration (CMA) and describe the precipitation phase (rain, snow or sleet). The class sleet has a significantly smaller number of samples than rain and snow. The *Pre2D* data consist of two continuous attributes (surface temperature and dew-point temperature), which are widely used in operation. To better describe the relationship between the local vertical profile of temperature and precipitation phase, additional numeric attributes of the temperature and height at 700, 850, 925 and 1,000 hPa, relative humidity, wind information and location information are included. Furthermore, *PreND* contains five meteorological characteristics (nominal attributes): season, climate, city level, coastal region and land use.

Both *Pre2D* and *PreND* were obtained from 888 automatic weather stations in North China from September 2004 to April 2015. In this study the data from September 2004 to April 2013 are used for training and the data from September 2013 to April 2015 are used for testing. The experiments are conducted with ten runs for *Pre2D* and *PreND* since the tested methods are stochastic algorithms.

4.2 Competitors and Parameters

Our proposed technique is first compared with MDO [21]. However, we have demonstrated that MDO produces unrealistic values in most cases, which should be removed to ensure the classifiers' operation. We compare our method with two related techniques: MDO and MDO+. For MDO, all unrealistic samples are eliminated after performing Algorithm 1. For MDO+, the strategy detailed in Section 3.3 is applied to ensure that no unrealistic samples are generated. The performance of MDO and MDO+ is tested for only 10 numeric data sets because they canot be used for mixed-type data sets.

Furthermore, we also compare the proposed technique with four other well-known and representative methods for multi-class imbalanced problems:

- SSMOTE [39][4]: refers to Static-SMOTE. This method develops a mechanism to iteratively generate synthetic samples via SMOTE in c steps and to modify the distribution of classes in the training data set.
- GCS [41]: refers to RESCALE$_{\text{new}}$. For cost-sensitive learning, this method generalizes the rescaling approach for multi-class problems. If the costs are consistent, rescaling is conducted directly; otherwise, rescaling is applied after pairwise coupling.
- ABNC [42]: refers to AdaBoost.NC. This method combines the AdaBoost algorithm [43] with negative correlation learning [44]. The ensemble diversity can be emphasized in this way.
- OSMOTE [40]: refers to OVOSMOTE. This method uses SMOTE as an ad hoc approach after OVO decomposition.

OSMOTE is a representative technique of the combination of binarization techniques and ad hoc approaches [40]. Moreover, in [21] the authors compared MDO with several two-class over-sampling approaches applying OAA and demonstrated the potential of MDO. Hence, in our experiments we do not consider two-class over-sampling approaches that use class decomposition except for OSMOTE.

We use the Matlab programming language to implement all the tested methods except ABNC, which is implemented using the KEEL software tool [46]. To perform a fair comparison, we use a uniform and widely adopted base classifier in the learning stage. Here, the C4.5 decision tree [45] with the default settings which is implemented in KEEL is selected. Because the previous study in [21] showed similar results with different base classifiers, we do not conduct additional experiments using different base classifiers in this study.

We set the parameter values for different methods according to the recommendations of the corresponding authors. In the case of SSMOTE and OSMOTE, 5 nearest neighbours of the minority class are considered. For GCS, we perform rescaling through instance-weighting. For ABNC, the penalty strength λ is set to 2, and the number of classifiers composing the ensemble is 51. In MDO, MDO+ and AMDO, $K1$ and $K2$ are set to 5 and 10, respectively, except for two data sets (*Balance* and *Breast-tissue*), because we could not find any candidates for some of the minority classes in this setting. Thus, we set $K1 = 1$ and $K2 = 10$ for *Balance* and *Breast-tissue*.

4.3 Evaluation Metrics

Considering the characteristics of imbalanced problems, we consider both the overall performance and the accuracy of the minority class in this study. P_{min} is used to reflect the precision of the minority class with the minimum size [21], [42]. The precision of the ith class can be computed as

$$P_i = \frac{TP_i}{TP_i + FP_i}, \tag{11}$$

where TP_i is the number of well-classified samples in the ith class, and FP_i is the number of samples with incorrect predictions in the ith class. Correspondingly, we use P_{avg} to reflect the average precision over all classes [47] (also called the average accuracy in [40], [48])

$$P_{avg} = \frac{1}{c}\sum_{i=1}^{c} P_i. \tag{12}$$

AUC is also widely used [21], [42], [47], [49]; thus, we also consider MAUC to describe the average ability to separate any pair of classes [50], [51]

$$\mathrm{MAUC} = \frac{2}{c(c-1)}\sum_{i<j}\frac{A_{i,j} + A_{j,i}}{2}, \tag{13}$$

where $A_{i,j}$ is the AUC between class i and class j. Note that for multi-class problems, $A_{i,j}$ and $A_{j,i}$ may not be equal. Correspondingly, we develop the AUC of the minority class (AUCm) to reflect how the minority class with minimum size can be separated from the other classes

$$\mathrm{AUCm} = \frac{1}{c-1}\sum_{i\neq min}\frac{A_{i,min} + A_{min,i}}{2}. \tag{14}$$

In summary, with respect to precision, we regard P_{min} and P_{avg} as the evaluation metrics, while we consider AUCm and MAUC for AUC. For each of the metrics, the higher it is, the better the performance is.

[4] SSMOTE only refers to the preprocessing stage, as a uniform classifier is applied for learning. A description can be found in Section 3.1 of [40].

TABLE 2
Means and Standard Deviations of P_{min} (%) and P_{avg} (%) Results from MDO, MDO+, and AMDO

	P_{min} (%)		
Data Set	MDO	MDO+	AMDO
Balance	$2.00_{4.47}$	$0.00_{0.00}$	$\mathbf{10.22}_{\mathbf{0.50}}$
Hayes-roth	$\mathbf{100.00}_{\mathbf{0.00}}$	$\mathbf{100.00}_{\mathbf{0.00}}$	$\mathbf{100.00}_{\mathbf{0.00}}$
New-thyroid	$83.33_{16.67}$	$96.67_{7.45}$	$\mathbf{100.00}_{\mathbf{0.00}}$
Page-blocks	$75.33_{16.26}$	$93.33_{9.13}$	$\mathbf{96.67}_{\mathbf{7.45}}$
Dermatology	$95.00_{11.18}$	$95.00_{11.18}$	$\mathbf{100.00}_{\mathbf{0.00}}$
Breast-tissue	$\mathbf{60.00}_{\mathbf{27.89}}$	$53.33_{29.81}$	$\mathbf{60.00}_{\mathbf{27.89}}$
UKM	$86.00_{13.42}$	$86.00_{13.42}$	$\mathbf{94.00}_{\mathbf{8.94}}$
Vertebral-column	$68.33_{22.36}$	$66.67_{28.87}$	$\mathbf{86.67}_{\mathbf{9.50}}$
Ecoli	$75.00_{30.62}$	$\mathbf{90.00}_{\mathbf{13.69}}$	$\mathbf{90.00}_{\mathbf{13.69}}$
Pre2D	$33.44_{8.77}$	$33.33_{9.53}$	$\mathbf{48.65}_{\mathbf{6.55}}$
Average	67.84	*71.43*	**78.62**
Mean rank	*2.35*	2.45	**1.20**

	P_{avg} (%)		
Data Set	MDO	MDO+	AMDO
Balance	$57.28_{2.92}$	$55.45_{1.66}$	$\mathbf{60.37}_{\mathbf{2.06}}$
Hayes-roth	$84.91_{6.71}$	$\mathbf{84.97}_{\mathbf{6.74}}$	$\mathbf{84.97}_{\mathbf{6.74}}$
New-thyroid	$89.81_{6.84}$	$94.98_{3.59}$	$\mathbf{96.54}_{\mathbf{2.99}}$
Page-blocks	$81.24_{1.79}$	$86.13_{2.48}$	$\mathbf{88.77}_{\mathbf{1.92}}$
Dermatology	$95.67_{2.05}$	$96.06_{1.62}$	$\mathbf{96.88}_{\mathbf{0.26}}$
Breast-tissue	$63.22_{3.74}$	$\mathbf{66.56}_{\mathbf{2.17}}$	$63.22_{3.74}$
UKM	$91.45_{2.48}$	$91.92_{2.50}$	$\mathbf{94.23}_{\mathbf{2.14}}$
Vertebral-column	$78.22_{5.55}$	$76.56_{7.41}$	$\mathbf{81.89}_{\mathbf{2.38}}$
Ecoli	$77.24_{7.03}$	$82.30_{5.21}$	$\mathbf{82.44}_{\mathbf{5.08}}$
Pre2D	$74.16_{5.21}$	$74.03_{5.27}$	$\mathbf{77.15}_{\mathbf{4.88}}$
Average	79.32	*80.90*	**82.65**
Mean rank	2.65	*2.15*	**1.20**

The best result is in bold face, and the second best result is in italics.

TABLE 3
Means and Standard Deviations of AUCm (%) and MAUC (%) Results from MDO, MDO+, and AMDO

	AUCm (%)		
Data Set	MDO	MDO+	AMDO
Balance	$57.40_{2.78}$	$56.60_{0.88}$	$\mathbf{60.61}_{\mathbf{1.24}}$
Hayes-roth	$94.34_{2.52}$	$\mathbf{94.36}_{\mathbf{2.53}}$	$\mathbf{94.36}_{\mathbf{2.53}}$
New-thyroid	$91.85_{6.74}$	$97.04_{2.94}$	$\mathbf{98.37}_{\mathbf{1.41}}$
Page-blocks	$87.90_{4.66}$	$93.97_{2.91}$	$\mathbf{95.51}_{\mathbf{2.07}}$
Dermatology	$97.32_{3.46}$	$97.48_{3.23}$	$\mathbf{98.98}_{\mathbf{0.26}}$
Breast-tissue	$76.80_{7.31}$	$\mathbf{77.30}_{\mathbf{8.09}}$	$76.80_{7.31}$
UKM	$93.33_{3.88}$	$93.55_{3.92}$	$\mathbf{96.45}_{\mathbf{3.02}}$
Vertebral-column	$81.13_{6.76}$	$79.54_{8.76}$	$\mathbf{86.04}_{\mathbf{1.72}}$
Ecoli	$86.38_{9.56}$	$\mathbf{91.97}_{\mathbf{4.61}}$	$91.59_{4.19}$
Pre2D	$72.82_{6.83}$	$72.44_{6.91}$	$\mathbf{76.72}_{\mathbf{5.21}}$
Average	83.93	*85.43*	**87.54**
Mean rank	2.65	*2.05*	**1.30**

	MAUC (%)		
Data Set	MDO	MDO+	AMDO
Balance	$67.96_{2.19}$	$66.59_{1.25}$	$\mathbf{70.27}_{\mathbf{1.54}}$
Hayes-roth	$88.68_{5.03}$	$\mathbf{88.73}_{\mathbf{5.06}}$	$\mathbf{88.73}_{\mathbf{5.06}}$
New-thyroid	$92.36_{5.13}$	$96.24_{2.69}$	$\mathbf{97.40}_{\mathbf{2.24}}$
Page-blocks	$88.27_{1.12}$	$91.33_{1.55}$	$\mathbf{92.98}_{\mathbf{1.20}}$
Dermatology	$97.40_{1.23}$	$97.63_{0.97}$	$\mathbf{98.13}_{\mathbf{0.16}}$
Breast-tissue	$77.93_{2.24}$	$\mathbf{79.93}_{\mathbf{1.30}}$	$77.93_{2.24}$
UKM	$94.30_{1.65}$	$94.61_{1.67}$	$\mathbf{96.15}_{\mathbf{1.42}}$
Vertebral-column	$83.67_{4.16}$	$82.42_{5.55}$	$\mathbf{86.42}_{\mathbf{1.78}}$
Ecoli	$85.77_{4.39}$	$88.93_{3.26}$	$\mathbf{89.03}_{\mathbf{3.18}}$
Pre2D	$80.62_{6.22}$	$80.52_{6.27}$	$\mathbf{82.86}_{\mathbf{4.99}}$
Average	85.70	*86.70*	**88.00**
Mean rank	2.65	*2.15*	**1.20**

The best result is in bold face, and the second best result is in italics.

4.4 Comparison of AMDO with MDO and MDO+

In this section the proposed AMDO is compared with MDO and MDO+ on 10 numeric data sets. The purpose of this section is to show that our considerations in Section 3.2 and Section 3.3 improve the classifier performance. Tables 2 and 3 show the performance results of C4.5 for each data set when applying MDO, MDO+ and AMDO. We summarize the results as follows:

- Based on the individual results for each data set, AMDO performs the best for most of the considered data sets. The average performance and mean ranks of AMDO verify this conclusion. Compared with MDO, improvements of 15.89 percent in the average P_{min}, 4.20 percent in the average P_{avg}, 4.30 percent in the average AUCm and 2.68 percent in the average MAUC are obtained by AMDO. Compared with MDO+, improvements of 10.07 percent in the average P_{min}, 2.16 percent in the average P_{avg}, 2.47 percent in the average AUCm and 1.50 percent in the average MAUC are obtained by AMDO. Thus, AMDO is better than the original MDO method with respect to both overall performance and the accuracy of the minority class.
- In some data sets (e.g., *Hayes-roth* and *Breast-tissue*) AMDO performs worse than MDO/MDO+, indicating that the partially balanced resampling scheme may not always be the best for this type of over-sampling.
- From MDO to MDO+, the number of adaptive synthetic samples is increased and the performance is generally improved. However, for *Pre2D*, a data set with two numeric attributes, MDO performs similarly to MDO+, which can be regarded as a justification of Theorem 1 because MDO does not generate unrealistic samples when $d = 2$.

To evaluate the significance of the results in Tables 2 and 3, we apply Wilcoxon's test [52] to compare the differences statistically via pairwise comparisons. The Wilcoxon signed-rank test is applied, and the p-values and asymptotic p-values (p^*) corresponding to different pairs of comparisons for the 10 numeric data sets are obtained in Table 4. Additionally, for each comparison, the sum of the ranks in favor of the first algorithm (R^+) and the sum of the ranks in favor of the second algorithm (R^-) are provided. The p-values represent the degree of difference between two algorithms and enable us to determine whether they are significantly different. In this paper, we consider a difference to be significant at $p < 0.05$.

The results in Table 4 show that there are no statistical differences between MDO and MDO+ for any of the evaluation metrics. When comparing AMDO with MDO, AMDO always obtains a higher R^+ and the associated p-values show statistical differences for P_{min}, P_{avg}, AUCm and MAUC. At the same time, although AMDO always performs better than MDO+, the two algorithms are significantly different only for P_{min} and AUCm.

TABLE 4
Results of Wilcoxon's Test after Comparing MDO, MDO+, and AMDO Using Each of the Evaluation Metrics (P_{min}, P_{avg}, AUCm, MAUC)

	Comparison	R^+	R^-	p	p^*
P_{min}	MDO+/MDO	25.5	19.5	≥0.2	0.6784
	AMDO/MDO	53.5	1.5	0.0049	0.0069
	AMDO/MDO+	53.5	1.5	0.0049	0.0069
P_{avg}	MDO+/MDO	42.0	13.0	0.1602	0.1263
	AMDO/MDO	45.0	0.0	0.0039	0.0064
	AMDO/MDO+	38.0	7.0	0.0742	0.0580
AUCm	MDO+/MDO	38.0	17.0	≥0.2	0.2622
	AMDO/MDO	45.0	0.0	0.0039	0.0064
	AMDO/MDO+	42.0	3.0	0.0195	0.0178
MAUC	MDO+/MDO	42.0	13.0	0.1602	0.1263
	AMDO/MDO	45.0	0.0	0.0039	0.0064
	AMDO/MDO+	39.0	6.0	0.0547	0.0440

p: exact p-value, p^*: asymptotic p-value.

To explain the results of this statistical analysis, let us recall the procedures of MDO and MDO+. MDO often generates unrealistic synthetic samples, which are removed before learning, while MDO+ ensures that no unrealistic samples are generated. Since for $\mathbf{X}_s$, $n'_s < n_s$, after MDO+, the size of $\mathbf{X}_s$ is actually $n_{maj} - n'_s + n_s$, that is, the size of any minority class is larger than that of the majority class. As a result, the synthetic samples of MDO are *insufficient* for learning, while the excessive synthetic samples of MDO+ may cause *over-fitting*.

In general, the statistical analysis supports that our proposed AMDO can result in significant improvement in performance compared to MDO technique.

4.5 Comparison of AMDO with Other Multi-Class Imbalanced Learning Algorithms

In this section the proposed AMDO is compared with 4 other multi-class imbalanced learning algorithms on 10 numeric data sets and 7 nominal/mixed-type data sets. As a baseline, we provide the performance results obtained directly by C4.5 (Base).

Similarly, we also conduct non-parametric statistical analysis to evaluate whether the obtained results are significantly different. As recommended in [52], the Friedman test is applied. If the null hypothesis stating that all algorithms perform equally in mean rank is rejected, the corresponding post hoc (Bonferroni-Dunn) test is used to compare all algorithms to each other. Here, we also reject the null hypothesis in the case of $p < 0.05$ for the Friedman test. For the post hoc test, AMDO is selected as the "control" method. The performance of any algorithm and AMDO is deemed significantly different if their mean ranks differ by at least the associated critical difference (CD)

$$\mathrm{CD} = q_{\alpha}\sqrt{\frac{K(K+1)}{6D}}, \tag{15}$$

where K is the number of comparative algorithms, D is the number of data sets and q_{α} is the critical value. In the following $q_{\alpha=0.05}$ is considered for all post hoc tests.

4.5.1 Numeric Data Sets

In Tables 5 and 6 the means and standard deviations of the performance are provided for each numeric data set. The results of the Friedman test and corresponding Bonferroni-Dunn tests for the numeric data sets are shown in Table 7.

From a purely statistical point of view, AMDO does not demonstrate overwhelming superiority, except for P_{min}, where AMDO yields the best result for eight data sets. Taking the average performance over ten data sets and the mean ranks into account, we obtain three remarkable findings:

- First, the C4.5 decision tree is outperformed by each multi-class imbalanced learning algorithm for P_{min}, P_{avg}, AUCm and MAUC. This observation supports the advantages of applying ad hoc methods for this type of problem.
- Second, AMDO achieves the best average results and mean ranks for P_{min}, P_{avg}, AUCm and MAUC.
- Third, with repsect to the second best performance, the situation is more complex. OSMOTE obtains the second best result for the average P_{min}, whereas GCS yields the second best results for the average P_{avg}, AUCm and MAUC. However, ABNC always achieves the second best mean ranks results.

According to Table 7, we reject the null hypothesis for P_{min} ($p = 0.0344$) and AUCm ($p = 0.0135$) at a significance level of 0.05. Based on the rejection, a post hoc test is conducted, and the rank differences between the comparative method (Base, SSMOTE, GCS, ABNC and OSMOTE) and control method are computed. In this case $\mathrm{CD} = 2.596\sqrt{\frac{6\cdot 7}{6\cdot 10}} = 2.16$. As a result, AMDO significantly outperforms Base with respect to P_{min} and AUCm, GCS with respect to P_{min} and SSMOTE with respect to AUCm. Further, this post hoc test suggests that the performance of ABNC and AMDO are comparable for P_{min} and AUCm. However, this conclusion is inconsistent with the third findings from Tables 5 and 6 because the average results are more sensitive to the performance for a single data set (e.g., *Breast-tissue* and *Pre2D*) than are the mean ranks. Thus, although GCS produces the best result in only a few data sets, the absolute values are prominent, which leads to lower mean ranks. On the other hand, ABNC is more robust to some degree (i.e., there is no prominent advantage or disadvantage of ABNC), which leads to higher mean ranks but lower average results.

Hence, for numeric data sets, AMDO is recommended to improve the performance of C4.5 for multi-class imbalanced problems. Both the overall performance (P_{avg} and MAUC) and accuracy for the minority class (P_{min} and AUCm) are improved, and the improvements in P_{min} and AUCm are significant.

4.5.2 Nominal/Mixed-Type Data Sets

This set of experiments is devoted to verifying whether AMDO has successfully adapted MDO to nominal/mixed-type cases. The detailed results of seven nominal/mixed-type data sets can be found in Tables 8 and 9. The corresponding statistical tests are shown in Table 10.

The trend is quite different from that of the numeric data sets, where AMDO shows very promising behaviour, except for *Thyroid*. Generally, AMDO achieves the best performance for three data sets with respect to P_{min}, six data sets with respect to P_{avg}, five data sets with respect to AUCm and six

TABLE 5
Means and Standard Deviations of P_{min} (%) and P_{avg} (%) Results from Base, SSMOTE, GCS, ABNC, OSMOTE, and AMDO for the Numeric Data Sets

Data set	P_{min} (%)						P_{avg} (%)					
	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO
Balance	$0.00_{0.00}$	$8.44_{9.18}$	$6.44_{9.83}$	$2.22_{4.97}$	$\mathbf{12.44_{13.41}}$	$10.22_{0.50}$	$56.38_{1.75}$	$58.38_{2.94}$	$56.65_{4.03}$	$\mathbf{62.33_{3.29}}$	$57.86_{2.78}$	$60.37_{2.06}$
Hayes-roth	$\mathbf{100.00_{0.00}}$	$\mathbf{100.00_{0.00}}$	$\mathbf{100.00_{0.00}}$	$\mathbf{100.00_{0.00}}$	$\mathbf{100.00_{0.00}}$	$\mathbf{100.00_{0.00}}$	$84.91_{6.71}$	$84.67_{5.06}$	$\mathbf{85.33_{5.58}}$	$83.52_{7.08}$	$84.97_{6.74}$	$84.97_{6.74}$
New-thyroid	$83.33_{11.79}$	$90.00_{14.91}$	$93.33_{9.13}$	$93.33_{9.13}$	$90.00_{9.13}$	$\mathbf{100.00_{0.00}}$	$88.86_{6.02}$	$91.08_{3.02}$	$93.14_{4.55}$	$93.59_{1.45}$	$92.54_{6.61}$	$\mathbf{96.54_{2.99}}$
Page-blocks	$82.67_{11.88}$	$78.67_{6.91}$	$93.33_{14.91}$	$72.67_{24.99}$	$93.33_{9.13}$	$\mathbf{96.67_{7.45}}$	$84.30_{2.14}$	$84.57_{1.88}$	$88.77_{4.49}$	$79.70_{5.57}$	$\mathbf{89.61_{2.98}}$	$88.77_{1.92}$
Dermatology	$95.00_{11.18}$	$95.00_{11.18}$	$90.00_{13.69}$	$\mathbf{100.00_{0.00}}$	$90.00_{13.69}$	$\mathbf{100.00_{0.00}}$	$95.67_{2.05}$	$95.73_{1.83}$	$93.50_{2.71}$	$\mathbf{97.10_{0.75}}$	$95.31_{2.45}$	$96.88_{0.26}$
Breast-tissue	$\mathbf{60.00_{27.89}}$	$40.00_{36.51}$	$46.67_{29.81}$	$\mathbf{60.00_{27.89}}$	$53.33_{29.81}$	$\mathbf{60.00_{27.89}}$	$63.22_{3.74}$	$60.89_{3.94}$	$\mathbf{68.78_{5.77}}$	$66.00_{4.88}$	$65.83_{7.05}$	$63.22_{3.74}$
UKM	$88.00_{13.04}$	$92.00_{13.04}$	$90.00_{10.00}$	$\mathbf{94.00_{8.94}}$	$88.00_{16.43}$	$\mathbf{94.00_{8.94}}$	$92.18_{2.02}$	$92.57_{4.85}$	$91.03_{2.00}$	$\mathbf{94.49_{2.45}}$	$91.78_{2.32}$	$94.23_{2.14}$
Vertebral-column	$65.00_{16.03}$	$65.00_{19.00}$	$60.00_{19.90}$	$61.67_{18.26}$	$66.67_{5.89}$	$\mathbf{86.67_{9.50}}$	$76.44_{2.65}$	$77.22_{4.66}$	$75.67_{5.86}$	$76.67_{3.12}$	$77.00_{4.71}$	$\mathbf{81.89_{2.38}}$
Ecoli	$65.00_{37.91}$	$70.00_{27.39}$	$55.00_{32.60}$	$70.00_{27.39}$	$55.00_{32.60}$	$\mathbf{90.00_{13.69}}$	$74.64_{7.88}$	$72.81_{12.01}$	$72.73_{11.35}$	$76.23_{7.14}$	$73.12_{8.72}$	$\mathbf{82.44_{5.08}}$
Pre2D	$3.00_{0.00}$	$47.45_{6.23}$	$\mathbf{88.89_{0.55}}$	$27.63_{13.23}$	$81.98_{1.21}$	$48.65_{6.55}$	$66.19_{0.00}$	$78.01_{5.24}$	$\mathbf{90.24_{0.78}}$	$71.86_{7.06}$	$87.13_{1.70}$	$77.15_{4.88}$
Average	64.20	68.66	72.37	68.15	*73.08*	**78.62**	78.28	79.59	*81.58*	80.15	81.52	**82.65**
Mean rank	4.30	3.90	4.00	*3.45*	3.60	**1.75**	4.75	3.90	3.80	*3.00*	3.35	**2.20**

The best result is in bold face, and the second best result is in italics.

TABLE 6
Means and Standard Deviations of AUCm (%) and MAUC (%) Results from Base, SSMOTE, GCS, ABNC, OSMOTE, and AMDO for the Numeric Data Sets

Data set	AUCm (%)						MAUC (%)					
	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO
Balance	$56.95_{1.91}$	$58.12_{3.27}$	$57.10_{3.43}$	$60.52_{2.89}$	$58.58_{3.18}$	$\mathbf{60.61_{1.24}}$	$67.29_{1.31}$	$68.79_{2.20}$	$67.49_{3.02}$	$\mathbf{71.75_{2.47}}$	$68.39_{2.09}$	$70.27_{1.54}$
Hayes-roth	$94.34_{2.52}$	$94.25_{1.90}$	$\mathbf{94.50_{2.09}}$	$93.82_{2.65}$	$94.36_{2.53}$	$94.36_{2.53}$	$88.68_{5.03}$	$88.50_{3.79}$	$\mathbf{89.00_{4.18}}$	$87.64_{5.31}$	$88.73_{5.06}$	$88.73_{5.06}$
New-thyroid	$91.40_{5.33}$	$93.90_{4.45}$	$95.43_{3.76}$	$95.76_{2.37}$	$94.04_{5.11}$	$\mathbf{98.37_{1.41}}$	$91.64_{4.52}$	$93.31_{2.27}$	$94.86_{3.41}$	$95.19_{1.09}$	$94.40_{4.96}$	$\mathbf{97.40_{2.24}}$
Page-blocks	$90.75_{3.57}$	$89.84_{2.16}$	$94.81_{4.66}$	$86.82_{7.87}$	$94.96_{3.16}$	$\mathbf{95.51_{2.07}}$	$90.19_{1.34}$	$90.35_{1.18}$	$92.98_{2.80}$	$87.31_{3.48}$	$\mathbf{93.51_{1.86}}$	$92.98_{1.20}$
Dermatology	$97.32_{3.46}$	$97.39_{3.27}$	$95.55_{4.17}$	$\mathbf{99.05_{0.28}}$	$96.09_{4.09}$	$98.98_{0.26}$	$97.40_{1.23}$	$97.44_{1.10}$	$96.10_{1.62}$	$\mathbf{98.26_{0.45}}$	$97.19_{1.47}$	$98.13_{0.16}$
Breast-tissue	$76.80_{7.31}$	$72.18_{10.10}$	$75.80_{9.99}$	$\mathbf{78.30_{8.92}}$	$76.92_{9.75}$	$76.80_{7.31}$	$77.93_{2.24}$	$76.53_{2.36}$	$\mathbf{81.27_{3.46}}$	$79.60_{2.93}$	$79.50_{4.23}$	$77.93_{2.24}$
UKM	$94.14_{3.65}$	$94.94_{5.27}$	$94.19_{3.04}$	$96.41_{2.61}$	$94.01_{4.64}$	$\mathbf{96.45_{3.02}}$	$94.79_{1.34}$	$95.05_{3.23}$	$94.02_{1.33}$	$\mathbf{96.32_{1.64}}$	$94.52_{1.54}$	$96.15_{1.42}$
Vertebral-column	$79.25_{3.17}$	$79.96_{5.56}$	$78.38_{6.36}$	$79.42_{5.04}$	$80.04_{2.48}$	$\mathbf{86.04_{1.72}}$	$82.33_{1.99}$	$82.92_{3.50}$	$81.75_{4.39}$	$82.50_{2.34}$	$82.75_{3.53}$	$\mathbf{86.42_{1.78}}$
Ecoli	$83.07_{11.54}$	$83.43_{10.88}$	$79.78_{11.44}$	$84.78_{9.21}$	$79.74_{11.00}$	$\mathbf{91.59_{4.19}}$	$84.15_{4.93}$	$83.01_{7.50}$	$82.96_{7.10}$	$85.15_{4.46}$	$83.20_{5.45}$	$\mathbf{89.03_{3.18}}$
Pre2D	$63.02_{0.00}$	$77.25_{7.74}$	$\mathbf{91.53_{0.80}}$	$70.52_{8.86}$	$88.19_{1.33}$	$76.72_{5.21}$	$74.64_{0.00}$	$83.51_{5.32}$	$\mathbf{92.68_{0.55}}$	$78.89_{9.77}$	$90.35_{1.46}$	$82.86_{4.99}$
Average	82.70	84.13	*85.71*	84.54	85.69	**87.54**	84.90	85.94	*87.31*	86.26	87.25	**87.99**
Mean rank	4.75	4.00	3.90	*3.10*	3.45	**1.80**	4.75	3.90	3.80	*3.00*	3.35	**2.20**

The best result is in bold face, and the second best result is in italics.

data sets with respect to MAUC. The highest average results and mean ranks also support AMDO's superiority. The results obtained by GCS are the second best and comparable to those of AMDO. Surprisingly, ABNC, for which we observed very robust performance for the numeric data sets, shows an obvious decrease in performance.

We observe that the performance of AMDO decreases with respect to the baseline for *Thyroid*, where it is clear that all the classes have been well classified. As AMDO generates synthetic samples and increases diversity, the synthetic samples may alter the boundaries of well-classified classes and have a negative impact on classification in this case.

TABLE 7
Friedman Test with Corresponding Post Hoc Test for the Numeric Data Sets Using AMDO as the Control Method

	Friedman test (p-value)	Bonferroni-Dunn test (rank difference)				
		Base	SSMOTE	GCS	ABNC	OSMOTE
P_{min}	0.0344	**2.55**	2.15	**2.25**	1.70	1.85
P_{avg}	0.0661	-	-	-	-	-
AUCm	0.0135	**2.95**	**2.20**	2.10	1.30	1.65
MAUC	0.0661	-	-	-	-	-

$\alpha = 0.05$, CD $= 2.16$, *the value larger than* CD, *indicating a significant difference, is in bold face.*

Meanwhile, we also observe that the mean ranks of P_{min} and AUCm are inconsistent, while the mean ranks of P_{avg} and MAUC are consistent. This phenomenon is to be expected. A higher P_{min} indicates that a classifier is good at recognizing the minority class, but it is still possible that samples of other classes may be misclassified, which is harmful for separating the minority class from other classes and decreases AUCm. However, a higher P_{avg} indicates every class should be recognized better without severely jeopardizing the precision of other classes, which in turn, requires good separation of each pair of classes, producing a higher MAUC.

As shown in Table 10, the null hypothesis is rejected for P_{min} ($p = 0.0359$), P_{avg} ($p = 0.0362$) and MAUC ($p = 0.0362$) at a significance level of 0.05. Here, CD $= 2.576\sqrt{\frac{6\cdot 7}{6\cdot 7}} = 2.58$. AMDO significantly outperforms ABNC in P_{avg} and MAUC. Additionally, the post hoc test demonstrates that the performances of GCS and AMDO are

TABLE 8
Means and Standard Deviations of P_{min} (%) and P_{avg} (%) Results from Base, SSMOTE, GCS, ABNC, OSMOTE, and AMDO for the Nominal/Mixed-Type Data Sets

Data set	P_{min} (%)						P_{avg} (%)					
	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO
Contraceptive	$43.55_{6.04}$	$38.12_{6.78}$	$40.24_{5.81}$	$36.02_{4.51}$	**$47.10_{12.20}$**	$45.36_{6.98}$	$52.07_{2.15}$	$48.30_{3.56}$	$49.24_{0.74}$	$49.21_{2.19}$	$51.86_{3.04}$	**$52.52_{2.09}$**
Flare	$2.22_{4.97}$	$20.56_{14.65}$	**$30.28_{6.39}$**	$20.56_{12.36}$	$6.67_{9.94}$	$25.56_{18.26}$	$59.84_{1.51}$	$61.37_{3.40}$	$63.04_{1.93}$	$58.02_{1.58}$	$58.57_{1.80}$	**$63.45_{1.74}$**
Thyroid	$96.40_{2.49}$	$97.01_{2.99}$	$97.61_{2.47}$	$95.79_{2.68}$	**$98.81_{1.64}$**	$93.99_{2.97}$	$96.70_{1.58}$	$96.96_{0.49}$	$97.34_{0.89}$	$96.55_{0.74}$	**$97.72_{0.81}$**	$95.67_{0.98}$
Car	$63.08_{22.03}$	$61.54_{19.61}$	$89.23_{6.88}$	$67.69_{26.87}$	$61.54_{26.09}$	**$98.46_{3.44}$**	$77.26_{5.59}$	$76.81_{4.82}$	$91.72_{4.12}$	$78.54_{5.63}$	$80.37_{9.40}$	**$91.97_{1.31}$**
Nursery	$67.71_{3.70}$	$68.64_{6.64}$	$77.45_{6.54}$	$66.48_{3.31}$	$64.36_{6.46}$	**$93.90_{2.85}$**	$90.12_{1.19}$	$90.32_{1.94}$	$93.45_{1.78}$	$89.84_{1.05}$	$89.31_{1.86}$	**$95.75_{1.20}$**
Splice	$93.99_{3.50}$	$93.59_{3.96}$	**$95.16_{2.87}$**	$90.98_{6.17}$	$93.46_{4.78}$	$94.51_{3.02}$	$94.23_{1.28}$	$93.97_{0.98}$	$93.90_{1.23}$	$91.16_{2.67}$	$94.25_{1.13}$	**$94.79_{0.73}$**
PreND	$61.14_{0.00}$	$77.41_{2.53}$	$87.65_{1.31}$	$87.65_{1.22}$	$85.54_{2.65}$	**$87.88_{1.53}$**	$86.23_{0.00}$	$91.37_{1.99}$	$95.23_{0.64}$	$95.51_{0.78}$	$93.90_{1.35}$	**$95.58_{0.85}$**
Average	61.16	65.27	*73.95*	66.45	65.35	**77.09**	79.49	79.87	*83.42*	79.83	80.85	**84.25**
Mean rank	4.29	4.14	**2.14**	4.29	*4.00*	**2.14**	4.00	4.29	*2.86*	4.71	3.43	**1.71**

The best result is in bold face, and the second best result is in italics.

TABLE 9
Means and Standard Deviations of AUCm (%) and MAUC (%) Results from Base, SSMOTE, GCS, ABNC, OSMOTE, and AMDO for the Nominal/Mixed-Type Data Sets

Data set	AUCm (%)						MAUC (%)					
	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO	Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO
Contraceptive	$64.23_{2.14}$	$60.41_{3.20}$	$60.78_{1.44}$	$61.57_{1.32}$	$64.04_{3.10}$	**$64.63_{2.39}$**	$64.05_{1.61}$	$61.22_{2.67}$	$61.93_{0.55}$	$61.91_{1.64}$	$63.89_{2.28}$	**$64.39_{1.56}$**
Flare	$63.18_{0.96}$	$67.42_{4.40}$	**$70.28_{1.76}$**	$66.68_{3.30}$	$63.93_{2.43}$	$69.53_{4.81}$	$75.91_{0.90}$	$76.82_{2.04}$	$77.82_{1.16}$	$74.81_{0.95}$	$75.14_{1.08}$	**$78.07_{1.04}$**
Thyroid	$97.84_{1.11}$	$98.08_{0.86}$	$98.37_{0.94}$	$97.63_{0.87}$	**$98.82_{0.66}$**	$96.84_{1.07}$	$97.53_{1.19}$	$97.72_{0.37}$	$98.00_{0.67}$	$97.41_{0.55}$	**$98.29_{0.60}$**	$96.75_{0.73}$
Car	$81.64_{6.67}$	$81.08_{5.80}$	$94.52_{3.03}$	$82.99_{7.54}$	$82.94_{9.49}$	**$95.52_{1.08}$**	$84.84_{3.73}$	$84.54_{3.21}$	$94.48_{2.75}$	$85.69_{3.75}$	$86.92_{6.27}$	**$94.64_{0.87}$**
Nursery	$88.53_{1.33}$	$88.82_{2.32}$	$92.12_{2.23}$	$88.15_{1.18}$	$87.42_{2.24}$	**$96.65_{1.20}$**	$93.41_{0.79}$	$93.54_{1.29}$	$95.63_{1.18}$	$93.23_{0.70}$	$92.87_{1.24}$	**$97.16_{0.80}$**
Splice	$95.54_{1.27}$	$95.36_{1.21}$	$95.36_{1.29}$	$93.31_{2.51}$	$95.45_{1.33}$	**$95.99_{1.01}$**	$95.67_{0.96}$	$95.47_{0.73}$	$95.43_{0.92}$	$93.37_{2.00}$	$95.69_{0.85}$	**$96.09_{0.55}$**
PreND	$84.98_{0.00}$	$90.84_{3.25}$	$95.00_{0.34}$	$95.12_{0.47}$	$93.74_{1.99}$	**$95.15_{0.55}$**	$89.68_{0.00}$	$93.52_{2.53}$	$96.42_{0.23}$	$96.63_{0.42}$	$95.43_{0.78}$	**$96.70_{0.48}$**
Average	82.28	83.14	*86.63*	83.64	83.76	**87.76**	85.87	86.12	*88.53*	86.15	86.89	**89.11**
Mean rank	3.29	3.57	*2.43*	3.86	3.14	**1.71**	4.00	4.29	*2.86*	4.71	3.43	**1.71**

The best result is in bold face, and the second best result is in italics.

similar especially for P_{min}, where the rank difference between them is zero.

The results indicate that AMDO is comparable to GCS for nominal/mixed-type data sets. Furthermore, in most cases, AMDO achieves the best performance, which confirms that AMDO has successfully adapted MDO to nominal/mixed-type case.

4.5.3 All Data Sets

To complete our experimental study, we compare AMDO with SSMOTE, GCS, ABNC, and OSMOTE for all data sets. We aim to justify the effectiveness and robustness of AMDO when considering all data sets.

TABLE 10
Friedman Test with Corresponding Post Hoc Test for the Nominal/Mixed-Type Data Sets Using AMDO as the Control Method

	Friedman test (p-value)	Bonferroni-Dunn test(rank difference)				
		Base	SSMOTE	GCS	ABNC	OSMOTE
P_{min}	0.0359	2.14	2.00	0	2.14	1.86
P_{avg}	0.0362	2.29	2.57	1.14	**3.00**	1.71
AUCm	0.0863	-	-	-	-	-
MAUC	0.0362	2.29	2.57	1.14	**3.00**	1.71

$\alpha = 0.05$, CD $= 2.58$, *the value larger than CD, indicating a significant difference, is in bold face.*

In Table 11 the average results and mean ranks (presented in parentheses) are shown for all data sets. The situations of P_{min}, P_{avg}, AUCm and MAUC are very similar: AMDO has the best performance, followed by GCS. We provide the results of the statistical analysis in Table 12. In this case the Friedman test shows that the effect of the method used is statistically significant at the level of 0.05. Here, CD is $2.498\sqrt{\frac{5\cdot 6}{6\cdot 17}} = 1.35$. AMDO is significantly better than SSMOTE (for P_{min}, P_{avg}, AUCm and MAUC), ABNC (for P_{min} and AUCm) and OSMOTE (for P_{min} and AUCm). Although AMDO outperforms GCS in general, the statistical study indicates that the two methods are comparable.

Meanwhile, the improvement obtained by AMDO in P_{min} is more significant than that in P_{avg}, AUCm and

TABLE 11
Average Results (%) and Mean Ranks from SSMOTE, GCS, ABNC, OSMOTE, and AMDO for All Data Sets

	SSMOTE	GCS	ABNC	OSMOTE	AMDO
P_{min}	67.26(3.59)	*73.02(2.85)*	67.45(3.44)	69.90(3.32)	**77.99(1.79)**
P_{avg}	79.71(3.71)	*82.34(3.03)*	80.02(3.29)	81.24(3.03)	**83.31(1.94)**
AUCm	83.72(3.79)	*86.09(3.09)*	84.17(3.18)	84.90(3.21)	**87.63(1.74)**
MAUC	86.01(3.71)	*87.81(3.03)*	86.22(3.29)	87.10(3.03)	**88.45(1.94)**

The best result is in bold face, the second best result is in italics, and the mean rank is presented in parentheses.

TABLE 12
Friedman Test with Corresponding Post Hoc Test for All Data Sets Using AMDO as the Control Method

	Friedman test (p-value)	Bonferroni-Dunn test (rank difference)			
		SSMOTE	GCS	ABNC	OSMOTE
P_{min}	0.0061	**1.79**	1.06	**1.65**	**1.53**
P_{avg}	0.0205	**1.76**	1.09	1.35	1.09
AUCm	0.0034	**2.06**	1.35	**1.44**	**1.47**
MAUC	0.0205	**1.76**	1.09	1.35	1.09

$\alpha = 0.05$, $\mathrm{CD} = 1.35$, the value larger than CD, indicating a significant differencet, is in bold face.

TABLE 13
Average Results (%) and Mean Ranks of Minimum Sensitivity from Base, SSMOTE, GCS, ABNC, OSMOTE, and AMDO for All Data Sets

Base	SSMOTE	GCS	ABNC	OSMOTE	AMDO
52.94(4.47)	58.21(3.94)	*64.86(2.91)*	57.10(4.03)	61.28(3.56)	**65.18(2.09)**

The best result is in bold face, the second best result is in italics, and the mean rank is presented in parentheses.

MAUC. We wonder whether this implies a reduced performance in other classes, especially in the worst classified class. To clearly show the influence of applying ad hoc methods in the worst classified class, we refer to minimum sensitivity [39] and compare AMDO with other competitors. Note that the minimum sensitivity metric can measure the performance for the worst classified class. The average performance results are briefly shown in Table 13. We conclude that AMDO improves the performance not only in the minority class but also in the worst classified class.

4.6 Analysis of AMDO for the Precipitation Phase Recognition Problem

Pre2D and *PreND* are two typical cases where overlapping occurs between classes. The authors in [21] claim that for multi-class cases, the existing over-sampling techniques usually produce synthetic samples that increase the overlap between class regions, which worsens the performance of learning algorithms. They show that MDO can increase the generalizability of the classifier and reduce the risk of overlap. We want to examine whether AMDO, as an extension of MDO, can perform well for mixed-type data sets with overlap between classes.

In the case of mixed-type data sets, it is difficult to visualize the data to see the phenomenon of overlapping. However, we can rely on num (detailed in Section 2.3) to describe this situation. As shown in Fig. 2, the percentage of num of rain, snow and sleet in the *PreND* data set are provided. Note that $num < 5$ indicates that samples that belong to one class are surrounded by more samples of other classes. Thus, we can consider that samples with $num < 5$ are located in the overlapping regions. According to Fig. 2, more than 72 percent of the sleet samples suffer from overlapping. Tables 8 and 9 show that AMDO results in improvements of 43.74 percent in P_{min}, 10.84 percent in P_{avg}, 11.97 percent in AUCm and 7.83 percent in MAUC compared to the direct use of C4.5. Additionally, AMDO shows the best performance for *PreND* among the different multi-class imbalanced learning algorithms, which confirms that AMDO can handle mixed-type data sets with overlap between classes.

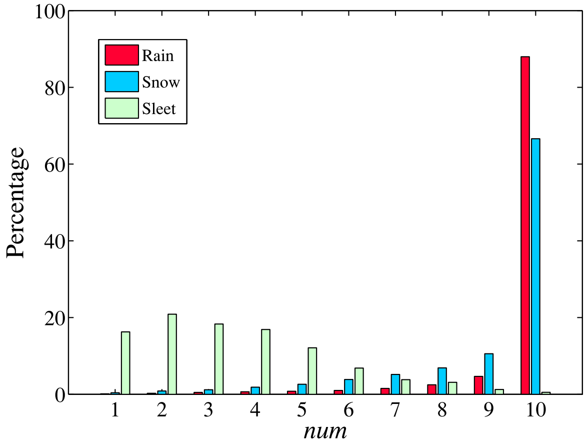


Fig. 2. Percentage of num of rain, snow, and sleet in *PreND* data set.

The synthetic samples generated by AMDO maintain the covariance structures of the different classes. If the dimensions of the data are low, preserving the covariance structures may not be a good choice. This can be seen from the performance of AMDO in *Pre2D* (see Tables 5 and 6), where the improvements produced by AMDO are limited. At the same time, we should highlight the effect of new attributes included in *PreND* for precipitation phase recognition.

5 Conclusion and Future Work

This paper addressed multi-class imbalanced problems, generating adaptive synthetic samples for classifiers. Our proposed technique, AMDO, generalizes the original MDO [21] and improves its performance. AMDO inherits the core idea of MDO, that is, synthetic samples are generated while maintaining the same Mahalanobis distance from their corresponding class mean. Meanwhile, AMDO adapts MDO to mixed-type data sets, develops a new scheme to partially balance the class distribution and optimizes the method used to synthesize samples. AMDO is realized with a theoretical guarantee and relatively low computational complexity. Our proposal was applied to 15 multi-class imbalanced benchmarks and two real-world classification problems of precipitation phase recognition.

The results confirm that AMDO improves both the accuracy of the minority class and the overall performance of the classifier in most data sets, showing promising precision and AUC. The experimental study indicates that AMDO successfully adapted MDO to mixed-type data sets and outperformed MDO for numeric data sets. With respect to all data sets, AMDO yielded the best results. Based on the statistical analysis, we concluded that AMDO significantly outperformed SSMOTE (for P_{min}, P_{avg}, AUCm and MAUC) and ABNC and OSMOTE (for P_{min} and AUCm) and was competitive with GCS. It is important to note that the performance of AMDO may be limited for low-dimensional data sets.

Future research will include the following: 1) as AMDO increases the diversity of samples, the combination of AMDO and ensemble learning will be investigated; 2) a more efficient approach to speed up the AMDO procedure; and 3) suitable parameter-optimizing methods for AMDO.

Appendix
Proof of the Theorem 1

Proof. Let $r_i = \frac{x_i^2}{\alpha \cdot \varsigma_i(\mathbf{X}_s)}$. When $i = 1, 2, \ldots, d-1$, we have $r_i \sim \mathcal{U}[0,1]$. Then, $r_d = 1 - \sum_{i=1}^{d-1} r_i$ and $x_d = \pm\sqrt{r_d \cdot \alpha \cdot \varsigma_d(\mathbf{X}_s)}$. Since $\alpha \cdot \varsigma_i(\mathbf{X}_s) > 0$, for $i = 1, 2, \ldots, d$, when x_d has an imaginary component, we have $r_d < 0$, that is, $\sum_{i=1}^{d-1} r_i > 1$. Thus, we aim to prove

$$\mathbb{P}\left(\sum_{i=1}^{d-1} r_i \leq 1\right) = \frac{1}{(d-1)!}. \tag{16}$$

Here, we first apply the inductive method to prove a strengthened equation

$$\mathbb{P}\left(\sum_{i=1}^{d-1} r_i \leq \tau\right) = \frac{\tau^{d-1}}{(d-1)!}, \tag{17}$$

where $0 \leq \tau \leq 1$.

For $d = 2$, it is easy to verify that $\mathbb{P}(r_1 \leq \tau) = \frac{\tau^{2-1}}{(2-1)!}$.

Now, we assume that $\mathbb{P}(\sum_{i=1}^{k-2} r_i \leq \tau) = \frac{\tau^{k-2}}{(k-2)!}$ holds when $d = k-1$. For $d = k$, let $p_{r_{k-1}}(r) = 1$ be the probability density function of r_{k-1}, we have

$$\begin{aligned}
&\mathbb{P}\left(\sum_{i=1}^{k-1} r_i \leq \tau\right)\\
=&\mathbb{P}\left(\sum_{i=1}^{k-2} r_i + r_{k-1} \leq \tau\right)\\
=&\int_{-\infty}^{\infty} \mathbb{P}\left(\sum_{i=1}^{k-2} r_i + r_{k-1} \leq \tau | r_{k-1} = q\right) p_{r_{k-1}}(q) dq\\
=&\int_{-\infty}^{\infty} \mathbb{P}\left(\sum_{i=1}^{k-2} r_i \leq \tau - q\right) p_{r_{k-1}}(q) dq\\
=&\int_{0}^{\tau} \mathbb{P}\left(\sum_{i=1}^{k-2} r_i \leq \tau - q\right) p_{r_{k-1}}(q) dq\\
=&\int_{0}^{\tau} \frac{(\tau - q)^{k-2}}{(k-2)!} \times 1 dq\\
=&\frac{\tau^{k-1}}{(k-1)!}.
\end{aligned} \tag{18}$$

In particular, it holds for $\tau = 1$, which concludes our proof. □

Acknowledgments

The authors would like to express their gratitude to the CMA Public Meteorological Service Center for providing the meteorological data. The authors are thankful for the financial support from the National Natural Science Foundation of China (61432008, 61472423, and U1636220).

References

[1] X.-Y. Gao, Z.-Y. Chen, S. Tang, Y.-D. Zhang, and J.-D. Li, "Adaptive weighted imbalance learning with application to abnormal activity recognition," *Neurocomputing*, vol. 173, pp. 1927–1935, Jan. 2016.

[2] M. Bach, A. Werner, J. Żywiec, and W. Pluskiewicz, "The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis," *Inf. Sci.*, vol. 384, pp. 174–190, Apr. 2017.

[3] J. A. Sanz, D. Bernardo, F. Herrera, H. Bustince, and H. Hagras, "A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 973–990, Aug. 2015.

[4] Q. Yang and X.-D. Wu, "10 challenging problems in data mining research," *Int. J. Inf. Tech. Dec. Mak.*, vol. 5, no. 4, pp. 597–604, Dec. 2006.

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

[6] H.-X. Guo, Y.-J. Li, J. Shang, M.-Y. Gu, and Y.-Y. Huang, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[7] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[8] 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.

[9] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Ann. Statist.*, vol. 26, no. 2, pp. 451–471, 1998.

[10] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, 2004.

[11] T.-W. Liao, "Classification of weld flaws with imbalanced class data," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 1041–1052, Oct. 2008.

[12] 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 *Proc. Comput. Intell. Knowl.-Based Syst. Des.*, vol. 6178, pp. 89–98, 2010.

[13] K. Chen, B.-L. Lu, and J. T. Kwok, "Efficient classification of multi-label and imbalanced data using min-max modular classifiers," in *Proc. Int. Joint Conf. Neural Netw.*, 2006, pp. 1770–1775.

[14] L. Cerf, D. Gay, N. Selmaoui-Folcher, B. Crémilleux, and J.-F. Boulicaut, "Parameter-free classification in multi-class imbalanced data sets," *Data Knowl. Eng.*, vol. 87, pp. 109–129, Sep. 2013.

[15] Z.-L. Zhang, B. Krawczyk, S. Garcia, A. Rosales-Pérez, and F. Herrera, "Empowering ono-vs-one decomposition with ensemble learning for multi-class imbalanced data," *Knowl.-Based Syst.*, vol. 106, no. C, pp. 251–263, Aug. 2016.

[16] M.-L. Lin, K. Tang, and X. Yao, "Dynamic sampling approach to training neural networks for multiclass imbalance classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 4, pp. 647–660, Apr. 2013.

[17] L.-X. Duan, M.-Y. Xie, T.-B. Bai, and J.-J. Wang, "A new support vector data description method for machinery fault diagnosis with unbalanced datasets," *Expert Syst. Appl.*, vol. 64, pp. 239–246, Dec. 2016.

[18] Y.-H. Shao, W.-J. Chen, J.-J. Zhang, Z. Wang, and N.-Y. Deng, "An efficient weighted Lagrangian twin support vector machine for imbalanced data classification," *Pattern Recog.*, vol. 47, no. 9, pp. 3158–3167, Sep. 2014.

[19] H.-L. Yu, C.-Y. Sun, X.-B. Yang, J.-F. Shen, and Y.-S. Qi, "ODOC-ELM: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data," *Knowl.-Based Syst.*, vol. 92, pp. 55–70, Jan. 2016.

[20] B. Mirza, Z.-P. Lin, J.-W. Gao, and X.-P. Lai, "Voting based weighted online sequential extreme learning machine for imbalance multi-class classification," in *Proc. IEEE Symp. Circuits Syst.*, May 2015, pp. 565–568.

[21] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[22] 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, no. 1, pp. 321–357, Jan. 2002.

[23] P. C. Mahalanobis, "On the generalized distance in statistics," in *Proc. Nat. Instit. Sci.*, vol. 2, pp. 49–55, 1936.

[24] C.-X. Jian, J. Gao, and Y.-H. Ao, "A new sampling method for classifying imbalanced data based on support vector machine ensemble," *Neurocomputing*, vol. 193, no. C, pp. 115–122, Jun. 2016.

[25] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "Class imbalances versus class overlapping: An analysis of a learning system behavior," in *Proc. Mexican Int. Conf. Adv. Artif. Intell.*, 2004, vol. 2972, pp. 312–321.

[26] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. Pacific-Asia Conf. Adv. Knowl. Dis. Data Eng.*, 2009, vol. 5476, pp. 475–482.

[27] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Adv. Intell. Comput.*, 2005, vol. 3644, pp. 878–887.

[28] H.-B. He, Y. Bai, E. A. Garcia, and S.-T. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[29] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behaviour of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.

[30] D. Chetchotsak, S. Pattanapairoj, and B. Arnonkijpanich, "Integrating new data balancing technique with committee networks for imbalanced data: GRSOM approach," *Cogn. Neurodyn.*, vol. 9, no. 6, pp. 627–638, Dec. 2015.

[31] B. Das, N. C. Krishnan, and D. J. Cook, "RACOG and wRACOG: Two probabilistic oversampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 222–234, Jan. 2015.

[32] X.-M. Zhao, X. Li, L.-N. Chen, and K. Aihara, "Protein classification with imbalanced data," *Proteins: Strcture Function Bioinf.*, vol. 70, no. 4, pp. 1125–1132, Mar. 2008.

[33] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *J. Artif. Intell. Res.*, vol. 6, no. 1, pp. 1–34, Jan. 1997.

[34] J. Shlens, "A tutorial on principal component analysis," 2014. [Online]. Available: http://arxiv.org/abs/1404.1100

[35] M. Chavent, V. Kuentz-Simonet, A. Labenne, and J. Saracco, "Multivariate analysis of mixed type data: The PCAmixdata R package," 2014. [Online]. Available: http://arxiv.org/abs/1411.4911

[36] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data sets," *Soft. Comput.*, vol. 13, no. 3, pp. 213–225, Oct. 2008.

[37] A. Fernández, M. J. del Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," *Int. J. Approx. Reason.*, vol. 50, no. 3, pp. 561–577, Mar. 2009.

[38] A. Fernández, M. J. del Jesus, and F. Herrera, "On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets," *Inform. Sci.*, vol. 180, no. 8, pp. 1268–1291, Apr. 2010.

[39] F. Fernández-Navarro, C. Hervás-Martínez, and P. A. Gutiérrez, "A dynamic over-sampling procedure based on sensitivity for multi-class problems," *Pattern Recog.*, vol. 44, no. 8, pp. 1821–1833, Aug. 2011.

[40] A. Fernández, V. López, M. Galar, M. J. del Jesus, and F. Herrera, "Analysing the classification of imbalacned data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowl.-Based Syst.*, vol. 42, pp. 97–110, Apr. 2013.

[41] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," *Comput. Intell.*, vol. 26, no. 3, pp. 232–257, Aug. 2010.

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

[43] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th. Int. Conf. Mach. Learn.*, 1996, pp. 148–156.

[44] Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Trans. Syst. Man Cybern. B*, vol. 29, no. 6, pp. 716–725, Dec. 1999.

[45] J. R. Quinlan, *C4.5:Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[46] J. Alcalá-Fdez, et al., "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft. Comput.*, vol. 13, no. 3, pp. 307–318, Oct. 2009.

[47] C. Ferri, J. Hernández-Navarro, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recog. Lett.*, vol. 30, no. 1, pp. 27–38, Jan. 2009.

[48] J. A. Sáez, B. Krawczyk, and M. Woźniak, "Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets," *Pattern Recog.*, vol. 57, no. C, pp. 164–178, Sep. 2016.

[49] O. Loyola-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and M. García-Borroto, "Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases," *Neurocomputing*, vol. 175, no. PB, pp. 935–947, Jan. 2016.

[50] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, Nov. 2001.

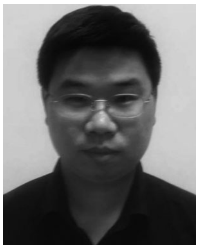
[51] K. Tang, R. Wang, and T. Chen, "Towards maximizing the area under the ROC curve for multi-class classification problems," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 483–488.

[52] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[53] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

[54] A. Zhu and Y.-Z. Li, "LazySVD: Even faster SVD decomposition yet without agonizing pain," 2017. [Online]. Available: http://arxiv.org/abs/1607.03463v2

Xuebing Yang received the BS degree from the Department of Precision Instruments and Mechanology, Tsinghua University, Beijing, China, in 2013. He is currently working toward the PhD degree in the Institute of Automation, Chinese Academy of Sciences (CAS). His research interests include machine learning and data mining.



Qiuming Kuang received the master's degree from the Department of Industry, Beijing Forest University, Beijing, China, in 2007. He is currently working toward the PhD degree in the Institute of Automation, Chinese Academy of Sciences (CAS). His research interests include big data, machine learning, and pattern recognition.

Wensheng Zhang received the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2000. He joined the Institute of Software, CAS, in 2001. He is a professor of machine learning and data mining and the director of Research and Development Department, Institute of Automation, CAS. His research interests include computer vision, pattern recognition, and artificial intelligence.

Guoping Zhang received the PhD degree in environmental remote sensing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), in 2002. He joined the National Meteorological Center, China Meteorological Administration in 2002. He is a senior research fellow at the Early Warning of Meteorological Hazards, Public Weather Services Center of CMA. His research interests include meteorological hazards forecasting and applications of weather radar.