Front. Comput. Sci., 2017, 11(5): 836–851
DOI 10.1007/s11704-016-5250-y

RESEARCH ARTICLE

Boosting imbalanced data learning with Wiener process oversampling

Qian LI[1], Gang LI[2], Wenjia NIU (✉)[1], Yanan CAO[1], Liang CHANG[3], Jianlong TAN[1], Li GUO[1]

1 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
2 School of Information Technology, Deakin University, Geelong VIC 3125, Australia
3 Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

Abstract Learning from imbalanced data is a challenging task in a wide range of applications, which attracts significant research efforts from machine learning and data mining community. As a natural approach to this issue, *oversampling* balances the training samples through *replicating* existing samples or *synthesizing* new samples. In general, *synthesization* outperforms *replication* by supplying additional information on the minority class. However, the additional information needs to follow the same normal distribution of the training set, which further constrains the new samples within the predefined range of training set. In this paper, we present the *Wiener process oversampling* (WPO) technique that brings the physics phenomena into sample synthesization. WPO constructs a robust decision region by expanding the attribute ranges in training set while keeping the same normal distribution. The satisfactory performance of WPO can be achieved with much lower computing complexity. In addition, by integrating WPO with *ensemble learning*, the *WPOBoost* algorithm outperforms many prevalent imbalance learning solutions.

Keywords imbalanced-data learning, oversampling, ensemble learning, Wiener process, AdaBoost

1 Introduction

Imbalanced data are pervasive in a wide range of applications [1–4], where the minority class is important and with unequal misclassification costs. In this circumstance, traditional strategy that assumes a balanced distribution usually encounters a significant bottleneck with suboptimal performance. To reduce the influence of the imbalanced classes, many solutions have been proposed through applying *resampling*, *ensemble learning* or their hybrid [5–7].

Resampling [8–11] balances the data set via oversampling the minority class samples or undersampling the majority class samples. *Undersampling* potentially leads to more efficient classification, but as it discards useful information in the majority class, it eventually trembles the decision boundary and leads to poor classifiers [2,12]. On the contrary, *oversampling* preserves more information via *replicating* existing samples or *synthesizing* new samples. *Replication* (also known as *Random Oversampling*) randomly duplicates a few minority samples, hence it usually produces smaller regions of minority classes that possibly results in overfitting [8,13]; while *Synthesization* alleviates this issue through supplying additional information on the minority class, such as *synthetic minority oversampling technique* (SMOTE) [10]. *Ensemble learning* addresses the problem of imbalanced dataset [1–4,14] through boosting a set of weak learners. Popular ensemble algorithms include *Boosting* and *Bagging*, both realize the classifier ensembles from a set of weak learners. The hybrid approach addresses the imbalanced data issue by integrating oversampling with ensemble learning, such as SMOTEBoost [5,6,15].

In spite of the improved performance, existing approaches

Received January 24, 2015; accepted February 19, 2016

E-mail: niuwenjia@iie.ac.cn

share some common limitations. For example, *oversampling* attempts to make full use of both majority and minority classes. However, no matter for Replication or Synthesization, the synthesized samples remain within the range of original training dataset and are prone to tremble the decision boundary. Since the training dataset is sampled from amounts of original set and may narrow down the range of the values of original set. A robust oversampling strategy is expected to break the attribute range confined in the training set and represent the as many characteristics of the original set as possible. Moreover, the ensemble learning still dominates in tackling imbalanced learning issue [5,6,15], though training base learners in each iteration results in significant computing complexity. The oversampling technique is also expected to be efficient for integration with ensemble learning. In conclusion, a better oversampling is in demand to break the confined attribute range and exert less computational cost to ensemble learning.

Wiener Process (also known as *Brownian Motion*), origins from physics, formulates the particles' macroscopic movement in d-dimensional space with linear complexity [16]. In this paper, we adopt the Wiener Process into oversampling to generate new minority class samples, by simulating the value of a new sample as the particle's position after movement. As Wiener Process exhibits regularity when repeating, it can be proved that synthesized samples of oversampling based on Wiener Process conform to the distribution of the original minority class. In addition, oversampling based on Wiener Process imposes no constraint on the values of new samples, which keeps the opportunity of breaking the attribute range confined by the training set.

The contributions of this paper are two-fold:

- To the best of our knowledge, this is the first work that brings the Wiener Process into oversampling. The proposed *Wiener process oversampling* (WPO) constructs a robust decision region by expanding the range of values in the training data set while preserving the same normal distribution. This decision boundary is not constrained by the given training data set and allows the classifier to gain a stable and robust decision. Experiment results also indicate that WPO consistently enhances classifiers' performance on imbalanced datasets. Especially, we propose WPOBoost algorithm by integrating WPO with ensemble learning, which achieves the best performance among the state-of-the-art ensemble methods.
- Most oversampling techniques achieve better performance through extensive computation. WPO consumes less time complexity to achieve superior capability on imbalanced dataset. This further reduces the complexity of integrating it into other classifiers.

The remainder of the paper is organized as follows: Section 2 provides the preliminary and related work; Section 3 analyzes the WPO algorithm and the superiority properties over the prevalent oversampling technique. Section 4 presents the experimental results compared with the prevalent approaches. Finally, Section 5 concludes the paper and envisages future directions.

2 Preliminary and related work

The imbalanced datasets are encountered by numerous real-world applications where the class distributions are highly imbalanced [1,2]. Most classification algorithms implicitly assume a balanced distribution and equal misclassification cost when optimizing the overall accuracy [17,18]. However, the imbalanced datasets usually violate these assumptions and compromise the classification performance. Without loss of generality, in this paper, we discuss the binary classification and assume the minority class as the positive class, and the majority class as the negative class.

2.1 Three strategies

Existing imbalanced data learning methods can be categorized into three strategies: oversampling, ensemble learning and their hybrid. Oversampling and ensemble learning tackle the issue on the data and the algorithm level respectively, while their hybrid oversamples the data set before feeding them into ensemble learning.

In real world applications, the dataset is *imbalanced* when the ratio of the minority class to the majority class is significantly low [19]. As a popular solution to the imbalanced data, *Oversampling* re-balances the data through increasing the size of minority class. A variety of *oversampling* methods have been proposed in recent years, such as *Random Oversampling* (RO) and SMOTE [9,10,20]. Among them, RO randomly selects some minority samples to replicate such that the dataset can be re-balanced [8]. As a typical synthesization method that explores the characteristics of minority samples, SMOTE constructs the synthetic sample via linear interpolations between two adjacent samples: it first finds k nearest-neighbors for each minority sample; then draws a random point from the line connecting every pair of neighbors.

Instead of directly processing the imbalanced data, Ensemble Learning invokes a series of individually trained base classifiers through *Boosting* or *Bagging*. A typical Boosting approach is *AdaBoost* that adapts weight-update rule to guarantee that misclassified minority examples are assigned with higher weights. A variant of AdaBoost, cost-sensitive boosting algorithms (e.g., *Easy-Ensemble* and *BalanceCascade*) [21], intentionally increases the weights of samples with higher misclassification cost in the boosting process. A final decision is made by voting from these trained learners. Both Boosting and Bagging approaches exhibit high flexibility in combining base learner without prior knowledge [17,22–24].

The hybrid of oversampling with Ensemble Learning is another common imbalanced data learning strategy [5–7,15], where every successive classifier of ensemble learning emphasizes more on the minority class through oversampling. The hybrid strategy broadens the decision region for the minority class, since every trained classifier is constructed from a different oversampled data set. For instance, SMOTEBoost [6] synthesized new minority-class samples using the SMOTE algorithm in each boosting iteration.

2.2 Summary

Although existing approaches demonstrated their success in imbalanced data learning, two issues remain open. Firstly, in the oversampling strategy, synthesization provides a broad decision region than Replication, but resulting in heavy computational cost. Secondly, the hybrid strategy re-balances the dataset with a broad decision region, but the Ensemble Learning strategy requires significant computational cost in training, especially when combining with the oversampling strategy. Hence, this paper aims to alleviate the above limitations.

3 Wiener process oversampling

Oversampling in general outperforms over *undersampling*, but those synthesized samples remain within the range of original training dataset and the approach is prone to tremble the classification decision boundary. To alleviate this issue, we propose WPO, through which the new samples exhibits regularity and conform to the mathematical expectations and variances of the original minority class. More importantly, WPO with the typical stochastic feature imposes no constraint on new samples' generation, so the decision region of the training classifier is augmented.

Definition 1 (Wiener process) A standard Wiener Process is a stochastic process by a family of random variables $\{W_t\}_{t\geqslant 0^+}$, where t is a nonnegative real number, satisfying the following properties [16]:

- $W_0 = 0$.
- Function $t \to W_t$ is continuous in t with probability 1.
- The process $\{W_t\}_{t\geqslant 0^+}$ has stationary, independent increments.
- The increment $W_{t+s} - W_s$ follows *Gaussian* distribution $N(0, t)$.

The distribution of the increment $W_{t+s} - W_s$ is the same as $W_t - W_0 = W$ for any $0 < s$ and $t < \infty$. Meanwhile, "*independent increments*" means that for every choice of nonnegative real numbers $s_0 \leqslant s_1 < t_1 \cdots \leqslant s_n < t_n < \infty$, the increment variables $W_{t1} - W_{s1}, W_{t2} - W_{s2}, \ldots, W_{tn} - W_{sn}$ are mutually independent.

By modeling new samples as the new positions of the particle, Wiener Process can effectively handle issues such as the constraint on the range of synthetic sample values or the high complexity in the current oversampling techniques. Since "*stationary increments*" simulates the particle movement, Wiener Process imposes no predefined constraints on the new positions of the particle. The decision region is the hyperspace that partitions the underlying vector space int positive and negative class. Values of new samples by WPO can go beyond the range of original training set, which potentially provides more decision region for modeling. Moreover, "*independent increments*" allows the *Wiener Process* to determine the relationship between the new samples and the existing ones such that the new samples conform to the distribution of existing data set.

As an approximation to random physical processes, WPO can oversample the new samples with above mentioned properties. The mathematical model of WPO can be defined as follows. Let vector $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iK}\}$ be the sample instance of K attributes from the dataset and $X_i = \{x'_{i1}, x'_{i2}, \ldots, x'_{iK}\}$ be the new synthetic sample. Wiener Process can be used to oversample K attributes of X_i and move each attribute value x_{ij} along the *Brownian* motion path. Let $x_{ij} = x_{ij}^{(t)}$ denote the attribute value j at t time, each attribute j can be modeled as a set of stochastic variables independent of each other. $x_{ij}^{(t+\Delta t)}$ represents the synthetic attribute x'_{ij} after time increases Δt:

$$Z = \frac{x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}}{\sqrt{c^2 \Delta t}} \sim N(0, 1);$$

$$x_{ij}^{(t+\Delta t)} = x_{ij}^{(t)} + Z \cdot \sqrt{c^2 \Delta t}, \tag{1}$$

where c is a constant.

3.1 Wiener process oversampling algorithm

Algorithm 1 presents the pseudocode of WPO and adapts different strategies for continuous or nominal attributes. For *continuous* attributes, let Δt be the time interval. According to Eq. (1), the synthetic sample x_{ij} can be regarded as $x_{ij}(t+\Delta t)$, a new position of $x_{ij}(t)$ after Δt duration. In light of this, WPO oversamples x_{ij} from step 1) to step 3). In step 2), the variable *randn* specifies an output following a Gaussian distribution, which is scaled by f in step 3). For nominal attribute j, WPO generates the x_{ij} by randomly replicating existing values of attribute j. After the Wiener Process on all K attributes, a new synthetic sample $X_i = \{x'_{i1}, x'_{i2}, \ldots, x'_{iK}\}$ is produced.

Algorithm 1 WPO: Wiener process oversampling

Inputs:

1) An attribute $x_{ij}(i = 1, 2, \ldots, N, j = 1, 2, \ldots, K)$ of sample X_i

2) Oversampling rate R

3) Time interval in seconds Δt

4) Diffusion coefficient d

5) Incremental quantity of Brown motion *incr*

Initialize: The amounts of synthetic samples $S = \lceil (R/100) \rceil \cdot N$

Outputs: The set of new samples X

Do for $s = 1, 2, \ldots, S$

  Do for $j = 1, 2, \ldots, K$

    For continuous attributes:

    1) Compute parameter factor $f = \sqrt{2\Delta t}$

    2) Calculate $incr = f \cdot randn$

    3) The synthetic attribute $x'_{ij} = x_{ij} + incr$

    For nominal attributes:

    1) Collect the attribute set A_j for j attribute.

    2) Randomly assign one value from A_j to x'_{ij}.

    Involve the attribute x'_{ij} into the synthetic sample X'_i.

  End Loop.

  Update $X = X \cup X'_i$

End Loop

• Complexity analysis

WPO is efficient in the time and the space complexity that are both mainly affected by the synthetic generation. Given the amounts of synthetic samples S to be generated, the total complexity is $O(S)$. WPO is more efficient than algorithms with complexity $O(k \cdot S)$ such as SMOTE that require the distances among k neighbors. In addition, WPO consumes less space than most existing oversampling techniques. For instance, instead of extra space for storing neighbors in SMOTE, the space consumed in WPO is allocated merely according to the number of synthetic samples S.

3.2 Properties of Wiener process oversampling

Oversampling has gained its advantages over undersampling, but those synthesized samples remain within the range of original training dataset and is prone to tremble the classification decision boundary. To alleviate this issue, we propose WPO as an alternative oversampling technique, because the new samples generated from WPO exhibit regularity and conform to the mathematical expectations and variances of the original minority class. More importantly, WPO with the typical stochastic feature imposes no constraint on new samples' generation, so the decision region of the training classifier is augmented. In this subsection, we will formally prove these two properties.

3.2.1 Distribution preservation

Based on the collection of samples, the original dataset is characterized by certain descriptive quantities, such as the expectation and the variance. Let μ_j and σ_j be the expected value and variance of attribute j in the original minority class. Similarity, Let μ'_j and σ'_j be the corresponding values of attribute j in synthetic samples.

Proposition 1 Each attribute j in the synthetic sample X_i approximately satisfies $\mu'_j = \mu_j$, for $n \to \infty$. Specifically, this proposition demonstrates that each attribute j for new samples by WPO conforms to the mathematical expectations of the original minority class $\mu_j = \mu'_j$ when the number of samples n in the training set is large enough.

$$Z = \frac{x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}}{\sqrt{c^2 \Delta t}} \sim N(0, 1). \tag{2}$$

Proof According to Eq. (2), the average value of synthetic samples for attribute j represented by $\overline{x}_j^{(t+\Delta t)}$ satisfying

$$\overline{x}_j^{(t+\Delta t)} = \frac{1}{n} \sum_{i=1}^{n} (x_{ij}^{(t)} + Z \cdot \sqrt{c^2 \Delta t}),$$

where n is the number of synthetic samples. The right part of the equation can be further simplified as

$$\overline{x}_j^{(t)} + c \cdot \sqrt{\Delta t} \cdot (\frac{1}{n} \sum_{i=1}^{n} Z),$$

which yields the equation

$$\overline{x}_j^{(t+\Delta t)} = \overline{x}_j^{(t)} + c \cdot \sqrt{\Delta t} \cdot \overline{Z}.$$

In light with the law of large numbers theorem, when $n \to \infty$, the average should be close to the expected value $\mu'_j = \overline{x}_j^{(t+\Delta t)}$

and $\mu_j = x_j^{(t)}$. That is

$$\mu_j' = \mu_j + c \cdot \sqrt{\Delta t} \cdot \overline{Z}.$$

As $Z \sim N(0,1)$ and $\overline{Z} = 0$ produces

$$\mu_j' = \mu_j,$$

the Proposition 1 is proved. □

Proposition 2 Each attribute j in the synthetic sample X_i approximately satisfies $\sigma_j'^2 = \sigma_j^2$, when $n \to \infty$.

Variance is another statistics that characterizes the distribution of a dataset. This proposition states that the attribute variance of new sample j is consistent with the original attribute j, namely $\sigma_j'^2 = \sigma_j^2$ when the number of synthetic samples n is large enough.

Proof For the variance, we have

$$\begin{aligned}\sigma_j(x_{ij}^{(t+\Delta t)}) &= \lim_{n\to\infty}\{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t+\Delta t)} - \overline{x_{ij}^{(t+\Delta t)}})^2\}\\ &= \lim_{n\to\infty}\{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t+\Delta t)} - \overline{x_{ij}^{(t)}})^2\}\\ &= \lim_{n\to\infty}\{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t)} + Z\cdot\sqrt{c^2\Delta t} - \overline{x_{ij}^{(t)}})^2\},\end{aligned}$$

which can be further factorized

$$\begin{aligned}&\lim_{n\to\infty}\{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t)} - \overline{x_{ij}^{(t)}})^2\} + \lim_{n\to\infty}\{c^2\cdot\Delta t\cdot\frac{1}{n}\sum_{i=1}^{n}\cdot Z^2\}\\ &+ \lim_{n\to\infty}\{2Z\cdot\sqrt{c^2\Delta t}\cdot\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t)} - \overline{x_{ij}^{(t)}})\}.\end{aligned}$$

Among them, the first term can be simplified as:

$$\lim_{n\to\infty}\{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t)} - \overline{x_{ij}^{(t)}})^2\} = \sigma_j(x_{ij}^{(t)}). \tag{3}$$

As $Z \sim N(0,1)$, Z is a bounded function and $\lim_{n\to\infty}\{\frac{1}{n}\sum_{i=1}^{n} Z\} = \mu_j(Z) = 0$ thus the second term is,

$$\lim_{n\to\infty}\{Z\cdot c^2\cdot\Delta t\cdot\frac{1}{n}\sum_{i=1}^{n}\cdot Z\} = c^2\cdot\Delta t\cdot Z\cdot\mu_j(Z) = 0. \tag{4}$$

The third term can be simplified as:

$$\begin{aligned}&\lim_{n\to\infty}\{2Z\cdot\sqrt{c^2\Delta t}\cdot\frac{1}{n}\sum_{i=1}^{n}(x_{ij}^{(t)} - \overline{x_{ij}^{(t)}})\}\\ &= 2Z\cdot\sqrt{c^2\Delta t}\cdot\{\lim_{n\to\infty}(\frac{1}{n}\sum_{i=1}^{n}x_{ij}^{(t)}) - \overline{x_{ij}^{(t)}}\}\\ &= 0.\end{aligned} \tag{5}$$

Accordingly, the Proposition 2 is satisfied based on Eqs. (3)–(5). □

With two propositions being proved, each attribute j in new samples synthesized by WPO is with the same expectation and variance as the attribute j in the original data set.

3.2.2 Decision region broadening

Traditional oversampling techniques such as SMOTE confine the new samples within the range of the original training set. In contrast, this range can be broadened by WPO as it allows the new samples to cover all possible values of minority class. In the following discussion, we assume the oversampling rate to be 300. For visualization purpose, we assume that each sample contains three attributes in all figures of this section.

Theorem 1 Denote a hypersphere $S(r_{\max})$ with radius $r_{\max}$ covering the range of original minority samples X_i, the synthetic sample X_i generated by SMOTE satisfies $X_i \in S(r_{\max})$.

Proof Since we assume each sample with three attributes, we draw 11 black points to represent them in Fig. 1(a) in 3 dimensions. According to the SMOTE algorithm [10], one synthetic sample is generated by multiplying the difference between the sample and its nearest neighbor by a random ratio between 0 and 1. Figure 1(a) depicts an original sample $orsa_1$ and its three nearest neighbours. One new sample $synsa_1$ with yellow color is produced by applying Eq. (6) on $orsa_1$ and one nearest neighbor $orsa_2$.

$$synsa_1 = orsa_1 + random(0,1)\cdot(orsa_1 - orsa_2). \tag{6}$$

There are two other synthetic samples produced in this way and colored with yellow in Fig. 1(a). The figure also shows the projection from 3 dimension to 2 dimension, where the blue points and green points corresponding to the original and synthetic samples in (X, Y) space.

In light of the convex theory [25], $\theta_1 x_1 + \cdots + \theta_k x_k$ is the convex combination of points $x_1, \ldots, x_k$, where $\theta_i > 0$ with $\theta_1 + \cdots + \theta_k = 1$. The set of all convex combinations of points $x_1, \ldots, x_k$ constructs a convex hull. One critical characteristic of Eq. (6) is that these three coefficients sum to 1.

Connecting all the blue points with black line segments can generate a convex hull in (X, Y) as shown in Fig. 1(a). To cover this convex hull extensively, a red big circle with $radius = r_{\max}$ confines all original samples and synthetic samples. Similar to the (X, Y) space, we can project these samples to (X, Z) and (Y, Z) space individually, with two big circles covering the samples in the corresponding space as well.

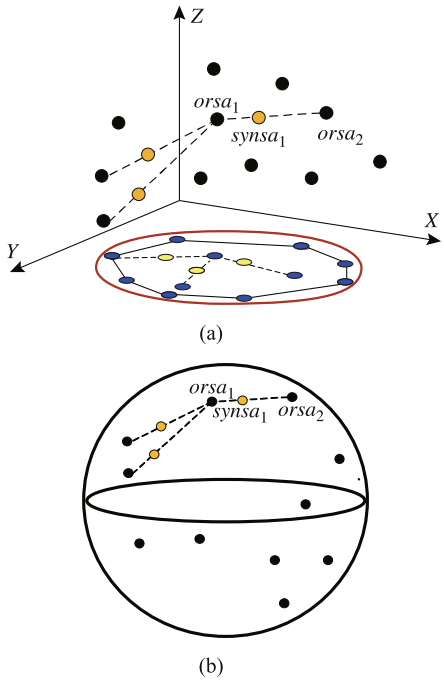


Fig. 1 Simulate the original and synthetic samples in three dimensions by SMOTE. (a) The generation of synthetic samples by SMOTE; (b) Synthetic samples are confined within a hypersphere

Accordingly, a hypersphere with the radius of the maximum value $r_{\max}$ among these three circles can contain all the samples in 3 dimension shown in Fig. 1(b). This case demonstrates that the samples by SMOTE are confined within a hypersphere. Apparently, the radius $r_{\max}$ of this hypersphere is specified by the values in (X, Y), (X, Z) and (Y, Z) of the original dataset. This statement can be extended into multi-dimension space. In conclusion, SMOTE confines the synthetic samples within a limited space formed by the original dataset. □

Theorem 2 Denote a hypersphere $S(r_{\max})$ with radius $r_{\max}$ covering the range of original minority samples X_i, any synthetic attribute x_{ij} generated by WPO satisfies that $x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)} \notin S(r_{\max})$.

Proof Since the range of the original dataset in SMOTE is $r_{\max}$, a hypersphere with radius more than $r_{\max}$ can cover all the original and synthetic samples. Equation (2) generates one value for attribute j in the synthetic sample X_i. For simplicity, the following part proves that $x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}$ can lie outside of the circle with $r_{\max}$.

From the mathematical integration and $Z \leqslant \epsilon$,

$$P(Z > \epsilon) = \int_{\epsilon}^{\infty} \phi(Z)\mathrm{d}Z \leqslant \frac{1}{\epsilon}\int_{\epsilon}^{\infty} Z\phi(Z)\mathrm{d}Z,$$

where Z follows the Gaussian distribution as in Eq. (2), then the probability density function of Z is $\phi(Z) = (2\pi)^{\frac{-1}{2}} e^{\frac{-Z^2}{2}}$, and then we have:

$$\frac{1}{\epsilon}\int_{\epsilon}^{\infty} Z(2\pi)^{\frac{-1}{2}} e^{\frac{-Z^2}{2}}\mathrm{d}Z = -\frac{1}{\epsilon}\int_{\epsilon}^{\infty} (-Z)(2\pi^{\frac{-1}{2}} e^{\frac{-Z^2}{2}})\mathrm{d}Z.$$

Since $\phi(Z)' = (-Z)(2\pi^{\frac{-1}{2}} e^{\frac{-Z^2}{2}})$, thus we have:

$$P(Z > \epsilon) \leqslant -\frac{1}{\epsilon}\int_{\epsilon}^{\infty} \phi(Z)'\mathrm{d}Z = \frac{\phi(\epsilon)}{\epsilon} = \frac{e^{\frac{-\epsilon^2}{2}}}{\epsilon\sqrt{2\pi}} \leqslant \frac{e^{\frac{-\epsilon^2}{2}}}{\epsilon}.$$

Similarity,

$$P(Z < -\epsilon) = \int_{-\infty}^{-\epsilon} \phi(Z)\mathrm{d}Z \leqslant \frac{1}{\epsilon}\int_{-\infty}^{-\epsilon} (-Z)\phi(Z)\mathrm{d}Z = \frac{e^{\frac{-\epsilon^2}{2}}}{\epsilon}.$$

Consequently,

$$P(|Z| > \epsilon) = P(Z > \epsilon) + P(Z < -\epsilon) \leqslant \frac{2e^{-\epsilon^2}}{\epsilon}.$$

From Eq. (2),

$$P(|Z| > \epsilon) = P(|\frac{x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}}{\sqrt{c^2\Delta t}}| > \epsilon) \leqslant \frac{2e^{-\epsilon^2}}{\epsilon}$$

$$P(|x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}| > \epsilon \cdot \sqrt{c^2\Delta t}) \leqslant \frac{2e^{-\epsilon^2}}{\epsilon}.$$

As the original attribute of sample X_i, $x_{ij}^{(t)}$ is confined in hypersphere with $r_{\max}$ radius. Let $\epsilon \cdot \sqrt{c^2\Delta t} = 2r_{\max}$. This formula $|x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}| > 2r_{\max}$ implies that the distance between the new synthetic attribute and $x_{ij}^{(t)}$ is larger than the diameter of the hypersphere. In other words, $x_{ij}^{(t+\Delta t)}$ lies out of the hypersphere that confines the new dataset generated by SMOTE. If this formula stands, we can conclude that the new synthetic attribute $x_{ij}^{(t+\Delta t)}$ goes beyond the range limits $r_{\max}$ defined by SMOTE.

The occurrence probability of the formula is

$$P(|x_{ij}^{(t+\Delta t)} - x_{ij}^{(t)}| > 2r_{\max}) \leqslant \frac{2e^{-\epsilon^2}}{\epsilon}.$$

Apparently, the scenario that a specific synthetic attribute lies out of the rang $r_{\max}$ with probability $2e^{-\epsilon^2}/\epsilon$, where $\epsilon = 2r_{\max}/c\sqrt{\Delta t}$. Other attributes can be proved in the similar way. Consequently, the synthetic sample can break the range limitation $r_{\max}$ of the hypersphere. □

Based on the proof above, we can conclude that, WPO expands the range of values in the training data set while preserving the same normal distribution. The decision boundary is not constrained by the given training data set and allows the classifier to gain a stable and robust decision.

3.3 WPOBoost algorithm

We proposed WPOBoost as a hybrid of ensemble learning and oversmapling, which adopts neural networks as basic

learners [6,15]. The details of WPOBoost in Algorithm 2 is given as follows.

Algorithm 2 The WPOBoost algorithm

Inputs:

1) Training dataset $\Gamma\{(X_i, Y_i)|x_{ij} \in X_i, Y_i \in \{-1, +1\}\}$

2) Number of iterations T

3) Over-sampling rate R

Initialize:

1) The distribution $D_1 = 1/m$

2) Randomly sample Γ into O

Outputs: The final hypothesis $h_f = sign(\sum_{t=1}^{T} \gamma_t \cdot h_t)$

Do for $t = 1, 2, \ldots, T$

1) Create new samples X' from O via Algorithm 1 with rate N, $\Gamma = \Gamma \bigcup (X', Y'), Y' \in \{+1\}$.

2) Compute the hypothesis $h_t : \Gamma \rightarrow \{-1, +1\}$ using D_t

3) Calculate the pseudo-loss $e_t = \sum D_t$

4) Update the weight parameter $\gamma_t = \frac{1}{2} \ln(\frac{1 - e_t}{e_t})$

5) Update distribution $D_{t+1}(X) = \frac{D_t(X) e^{-\gamma_t \cdot Y_i \cdot h_t(X)}}{Z_t}$, Z_t is a normalization constant.

6) Update $TrS = \{\Gamma, D_{t+1}\}$

7) Set $O = \{\Gamma | Y = +1 \text{ and } h_t(X) = -1\}$

End Loop

Similarly with the traditional *AdaBoost* algorithm, the classifier in each iteration of *WPOBoost* is trained with a probability distribution $D(X)$. $D(X)$ is increased when the sample X is misclassified. For instance, in the t + 1th iteration, the sample X_1 has a high probability to be chosen as inputs than X_2 when $D_c(X_1)$ is larger than $D_c(X_2)$. After the parameters in the hypothesis are defined by training, the error rate of each hypothesis is computed via e_t, which determines its' weight γ_t in the final hypothesis. In other words, the final classifier more likely accepts the decision of the classifier with less e_t by assigning a higher weight on it. The AdaNN trains the weak classifier by putting more emphasis on the misclassified samples. Finally, the AdaNN generates a single classifier h_{final} as a linear combination of T classifiers h_t.

Generally speaking, training stronger classifiers bring the superiority performance and somehow cause additional complexity. WPOBoost can balance the trade-off between performance and complexity based on two reasons. Firstly, WPOBoost merely oversampled the misclassified minority samples, to emphasize on those hard samples in each iteration and reduce the computing complexity. Secondly, considering the *oversampling* time in previous work is not only determined by *oversampling* rate but other factors (e.g., the number of neighbors in SMOTE), the WPO oversampling complexity is only relevant with the *oversampling* rate.

4 Experiments

Similar to existing imbalanced learning methods, we consider only two-class imbalanced problems in this experiment. We adopt six measurements in imbalanced learning, including *Precision*, *F-measure*, *G-means*, *Area Under the Curve* (AUC), *Receiver Operating Characteristic* (ROC) curve, and *precision-recall* (PR) curve. Among them, Precision, Recall, F-measure, G-means are based on the confusion matrix as shown in Table 1. The *receiver operating characteristic* (ROC) curve presents the pair (FP_{rate}, TP_{rate}), AUC denotes the area size under the ROC curve, while the PR curve visualizes the relative trade-offs between precision and recall rates.

$$
\begin{aligned}
FP_{rate} &= \frac{FP}{FP + FN} \\
TP_{rate} &= \frac{TP}{TP + FN} \\
TN_{rate} &= \frac{TN}{TN + FP} \\
Presion &= \frac{TP}{TP + FP} \\
Recall &= \frac{TP}{TP + FN} \\
G - mean &= \sqrt{TP_{rate} \cdot TN_{rate}} \\
F - measure &= \frac{2 \cdot Presion \cdot Recall}{Presion + Recall}.
\end{aligned}
$$

Table 1 The confusion matrix for the two-class classification problem

	Positive prediction	Negative prediction
Positive class	TP (true positive)	FN (false negative)
Negative class	FP (false positive)	TN (true negative)

4.1 Experiment settings

Sixteen datasets with different sparsity levels from the *UCI machine learning repository* [26] are used. Table 2 summarizes their basic information. For each dataset, a ten-fold cross-validation is performed, and the average over all folds is compared. We also perform statistical tests (t-test) to evaluate the significance of the results.

We compare thirteen different methods to balance the class distribution on training data. Among them, four methods are the *oversampling* with single classifiers. The others are ensemble methods with oversampling or under-sampling techniques. For single classifiers, we compare two prevalent oversampling techniques, Random Oversampling [8,13] and Synthetic Minority Oversampling TEchnique [10], with our method Wiener Process Oversampling. For the oversampling

rate, we set it at 100%, 200%, 300% and 400% and report their average result. Each ensemble method involve 20 weak classifiers and 50 iterations in each weak classifier training.

Table 2 Dataset characteristic

Dataset	Size	Attribute (cont.,nomi.)	Minority Tag	Distribution (min.,max.)	Imbalance ratio
Bupa	345	6 (6,0)	class 1	(0.420,0.580)	1.38
Wpbc	110	33 (33,0)	class R	(0.373,0.627)	1.68
Pima	768	8 (8,0)	class 1	(0.349,0.651)	1.87
Breast	698	9 (9,0)	class 2	(0.346,0.654)	1.89
German	2 310	20 (7,13)	class 2	(0.300,0.700)	2.33
Phoneme	5 404	5 (5,0)	class 1	(0.293,0.707)	2.41
Haberman	306	3 (2,1)	class 2	(0.265,0.735)	2.78
Hepatitis	155	19 (6,13)	class 2	(0.207,0.793)	3.84
Ecoli	336	7 (7,0)	class imU	(0.104,0.896)	8.60
Satimage	6 435	36 (36,0)	class 4	(0.098,0.902)	9.28
Vowel	988	13 (10,3)	class 11	(0.092,0.908)	9.98
Glass	214	9 (9,0)	class 3	(0.079,0.921)	11.58
Balance	625	4 (4,0)	class B	(0.078,0.922)	11.75
Abalone	4 177	8 (7,1)	class 13	(0.058,0.942)	18.65
Yeast	693	8 (8,0)	class VAC	(0.043,0.957)	22.10
Letter	20 000	16 (16,0)	class A	(0.039,0.961)	24.34

Note: *Size* denotes the amount of samples. *Attribute* specifies the attribute types and the corresponding amount, where *cont*. is continuous attribute and *nomi*. is nominal one. Majority class size *max*. divided by minority class size *min*. is *imbalance ratio*

- **CART + Over-sampling** Three over-sampling techniques, RO(R), SMOTE(S), and WPO(W), are applied to the dataset before inputting them into the *classification and regression trees* (CART) classifier [27];
- **Naive Bayes + Over-sampling** Similarly, we combined *naive bayes* (NB) [28] with three over-sampling techniques RO(R), SMOTE(S), and WPO(W) respectively for performance comparison;
- **KNN + Over-sampling** We combine K-nearest neighbor algorithm [29] (KNN) with RO(R), SMOTE(S), and WPO(W) as well, and the result is averaged from $K = 3$, $K = 5$ and $K = 7$;
- **Neural Network + Over-sampling** We used Back Propagation algorithm for training Neural network, and also applied RO (R), SMOTE (S), and WPO (W) in the Neural Network(NN) for comparison;
- **AdaCART** AdaBoost can combine several basic learners and boost the performance. AdaCART is a type of AdaBoost utilizing CART as the base learner without oversampling;
- **AdaNN** AdaNN is different from AdaCART and adapts Neural Network as the base learner;
- **SMOTEBoost** SMOTEBoost [5] combines AdaBoost with SMOTE to oversample the minority class cases in training set before training the basic learners. The goal of SMOTEBoost is to provide the learner with the broader representation of training set by oversampling from the minority class, thus indirectly changing the updating weights and compensating for skewed distributions;
- **SMOTEAdaNN** Similar to SMOTEBoost, SMOTEAdaNN introduces SMOTE into every training iteration of AdaNN. In other words, the CART as the basic learner in SMOTEBoost is replaced as neural network;
- **WPOBoost** Instead of combining SMOTE and AdaCART in SMOTEBoost, WPOBoost oversamples the more of the minority class cases by WPO in each round of boosting before feeding the data to train the neural networks of AdaNN. Moreover, WPOBoost merely oversampleS the misclassified minority samples to further reduce the bias inherent before the learning procedure due to the class imbalance.
- **EasyEnsemble** This method belongs to ensemble type and uses CART as the base learner. *EasyEnsemble* [21] samples several subsets from the majority class and outputs jointly the CART learners trained by the subsets;
- **BalanceCascade** This ensemble approach BalanceCascade [21] trains the learners sequentially, and in each step remove the classified majority class samples from further consideration;
- **Bagging** This method is proposed by [30], which generates several subsets from the training set by sampling with replacement as the training set and uses majority voting for final decision.

4.2 Evaluating the improvement on single classifiers

In general, single classifiers are not suitable for imbalanced learning, since they are trained with imbalanced dataset and unequal misclassification cost. Oversampling technique rebalances the training set and hence improves the capability of tackling imbalanced issue for single classifiers. We integrate four single classifiers including CART, Naive Bayes, KNN, Neural Network with RO, SMOTE and WPO, and consequently, their results reflects the performance of RO, SMOTE and WPO.

Tables 3–5 show the averaged results of compared F-measure, G-mean and AUC, under four oversampling rates. It is evident that WPO outperforms SMOTE on most metrics with varied improvements across different data sets. Table 6

Table 3 F-measure results for compared single classifiers

F-measure	CART+R	CART+S	CART+W	KNN+R	KNN+S	KNN+W	NB+R	NB+S	NB+W	NN+R	NN+S	NN+W
Bupa	0.596 5	0.621 0	0.598 6	0.592 5	0.593 7	**0.603 0**	0.588 5	0.586 6	**0.596 0**	0.571 8	0.555 9	**0.573 9**
Wpbc	0.393 9	0.436 5	**0.458 6**	0.503 0	0.601 5	0.431 1	0.559 8	0.515 1	**0.581 0**	0.391 3	0.542 5	**0.560 1**
Pima	0.597 0	0.608 3	**0.630 5**	0.570 9	0.567 3	**0.576 9**	0.541 8	0.571 0	**0.576 6**	0.637 3	0.623 4	**0.637 8**
Breast	0.901 8	0.912 5	**0.906 1**	0.928 4	0.932 7	0.930 7	0.923 2	0.925 6	**0.927 5**	0.921 0	0.928 1	0.926 5
German	0.484 8	0.498 2	0.487 5	0.373 4	0.382 5	**0.412 9**	0.541 1	0.556 6	0.546 9	0.438 7	0.426 3	**0.465 1**
Phoneme	0.738 2	0.739 5	**0.755 8**	0.781 2	0.788 4	0.787 8	0.670 2	0.671 6	**0.673 8**	0.654 9	0.672 5	0.663 7
Haberman	0.356 9	0.345 7	**0.379 3**	0.423 4	0.398 6	0.418 3	0.326 0	0.299 7	**0.382 4**	0.429 7	0.448 9	**0.486 2**
Hepatitis	0.837 3	0.842 7	**0.846 5**	0.821 3	0.817 3	**0.841 4**	0.878 7	0.893 6	0.888 7	0.837 6	0.838 6	**0.849 8**
Ecoli	0.859 7	0.869 5	0.863 3	0.858 5	0.866 1	**0.870 3**	0.859 2	0.861 4	0.859 3	0.842 6	0.819 8	**0.844 1**
Satimage	0.647 0	0.685 6	0.681 5	0.770 6	0.770 7	**0.786 7**	0.646 0	0.645 9	**0.650 7**	0.597 0	0.537 6	**0.609 1**
Vowel	0.932 5	0.931 1	**0.933 5**	0.946 0	0.916 3	0.935 6	0.817 5	0.815 9	**0.821 7**	0.925 7	0.941 0	0.934 9
Glass	0.580 9	0.575 3	**0.597 6**	0.581 3	0.575 9	**0.588 5**	0.584 6	0.584 4	**0.587 4**	0.583 2	0.590 1	**0.594 8**
Balance	0.069 7	0.050 9	**0.100 7**	0.070 7	0.041 2	**0.082 0**	0.179 7	0.291 2	0.285 9	0.160 6	0.091 3	0.104 5
Abalone	0.289 2	0.343 4	0.335 1	0.248 8	0.290 4	**0.295 9**	0.335 8	0.346 2	**0.349 7**	0.261 6	0.293 5	0.272 8
Yeast	0.789 6	0.769 2	**0.812 8**	0.797 5	0.817 4	0.809 8	0.850 4	0.853 3	**0.855 2**	0.829 8	0.838 7	**0.847 1**
Letter	0.805 9	0.798 7	**0.823 2**	0.795 4	0.786 1	**0.823 6**	0.854 1	0.859 9	0.856 6	0.837 1	0.838 5	**0.838 8**
Avg.	0.617 6	0.626 8	**0.638 2**	0.628 9	0.634 1	**0.637 2**	0.634 8	0.642 4	**0.652 5**	0.620 1	0.624 2	**0.638 1**

Table 4 G-mean results for compared single classifiers

G-mean	CART+R	CART+S	CART+W	KNN+R	KNN+S	KNN+W	NB+R	NB+S	NB+W	NN+R	NN+S	NN+W
Bupa	0.617 3	0.645 3	0.619 1	0.626 3	0.625 2	**0.627 1**	0.606 9	0.614 2	**0.617 7**	0.500 6	0.431 1	**0.504 9**
Wpbc	0.483 1	0.523 2	**0.656 4**	0.575 1	0.545 1	0.537 7	0.635 5	0.601 4	**0.652 4**	0.585 4	0.552 6	**0.643 5**
Pima	0.685 1	0.692 7	**0.714 8**	0.661 4	0.657 9	**0.672 2**	0.631 7	0.658 5	**0.662 7**	0.572 9	0.563 6	**0.668 8**
Breast	0.907 6	0.916 2	0.911 1	0.930 4	0.933 8	**0.934 8**	0.928 9	0.928 5	**0.932 1**	0.926 9	0.931 8	0.930 1
German	0.607 7	0.613 6	0.606 7	0.508 1	0.515 9	**0.541 6**	0.651 5	0.666 7	0.662 3	0.521 5	0.538 8	**0.573 7**
Phoneme	0.793 5	0.793 1	**0.805 5**	0.819 7	0.826 2	0.824 4	0.751 1	0.751 8	**0.753 9**	0.728 7	0.741 7	0.734 2
Haberman	0.517 6	0.509 5	**0.531 6**	0.587 4	0.570 4	**0.592 5**	0.593 5	0.572 0	**0.634 6**	0.593 6	0.540 3	**0.637 2**
Hepatitis	0.825 6	0.832 1	**0.839 3**	0.778 4	0.780 0	**0.825 6**	0.869 5	0.882 6	0.872 1	0.819 9	0.758 4	**0.831 6**
Ecoli	0.875 7	0.883 6	0.877 5	0.875 8	0.880 5	**0.881 3**	0.864 9	0.866 9	0.864 5	0.853 2	0.834 2	**0.854 3**
Satimage	0.752 9	0.787 6	**0.789 2**	0.847 0	0.853 7	**0.861 2**	0.821 6	0.822 7	**0.827 3**	0.682 1	0.631 6	**0.692 0**
Vowel	0.942 3	0.945 1	**0.946 7**	0.946 5	0.936 2	0.942 6	0.827 6	0.825 5	**0.831 2**	0.932 3	0.942 8	0.938 7
Glass	0.624 1	0.619 2	**0.644 5**	0.645 9	0.635 1	**0.653 9**	0.641 2	0.646 5	**0.648 4**	0.591 6	0.645 4	**0.651 5**
Balance	0.291 6	0.423 7	0.370 8	0.147 6	0.071 3	0.084 5	0.229 1	0.291 3	0.252 1	0.190 6	0.142 9	**0.252 7**
Abalone	0.480 0	0.513 1	**0.540 6**	0.372 2	0.408 2	**0.415 5**	0.452 1	0.465 6	0.464 3	0.387 8	0.411 1	0.341 7
Yeast	0.840 2	0.817 7	**0.849 1**	0.825 3	0.837 6	0.830 9	0.866 2	0.866 9	**0.867 3**	0.838 4	0.847 1	**0.854 1**
Letter	0.850 6	0.845 6	**0.861 4**	0.825 9	0.814 5	**0.841 5**	0.866 7	0.868 9	0.867 9	0.844 9	0.845 6	**0.847 1**
Avg.	0.693 4	0.710 1	**0.722 8**	0.685 8	0.680 7	**0.691 7**	0.702 4	0.708 2	**0.713 2**	0.660 7	0.647 4	**0.684 8**

Table 5 AUC results for compared single classifiers

AUC	CART+R	CART+S	CART+W	KNN+R	KNN+S	KNN+W	NB+R	NB+S	NB+W	NN+R	NN+S	NN+W
Bupa	0.624 7	0.649 7	0.627 8	0.629 9	0.633 0	0.632 3	0.608 1	0.622 6	**0.623 8**	0.658 3	0.571 4	0.626 2
Wpbc	0.502 6	0.525 9	**0.568 3**	0.586 9	0.621 1	0.564 1	0.644 9	0.603 5	**0.657 2**	0.505 4	0.492 6	**0.567 5**
Pima	0.693 1	0.698 7	**0.719 7**	0.665 9	0.661 4	**0.678 9**	0.668 5	0.689 5	0.688 2	0.789 0	0.751 1	**0.794 5**
Breast	0.908 0	0.916 6	0.911 3	0.930 7	0.934 1	**0.935 0**	0.929 2	0.928 8	**0.932 3**	0.950 3	0.954 1	0.943 3
German	0.623 1	0.635 1	0.623 8	0.561 2	0.565 1	**0.584 9**	0.661 1	0.671 1	0.669 5	0.778 4	0.779 2	**0.779 3**
Phoneme	0.796 6	0.796 6	**0.808 7**	0.823 5	0.829 2	0.820 8	0.751 5	0.752 4	**0.754 4**	0.859 6	0.865 1	0.853 8
Haberman	0.556 6	0.548 3	**0.559 9**	0.600 9	0.575 1	**0.622 2**	0.440 1	0.438 6	**0.510 9**	0.687 2	0.606 2	**0.762 6**
Hepatitis	0.827 7	0.833 3	**0.841 5**	0.787 7	0.788 9	**0.827 1**	0.871 7	0.885 2	0.873 7	0.862 4	0.867 1	**0.872 4**
Ecoli	0.8776	0.885 1	0.879 3	0.877 6	0.881 9	**0.884 1**	0.870 6	0.872 4	0.869 7	0.933 9	0.928 2	0.921 6
Satimage	0.7656	0.795 5	0.789 9	0.849 4	0.855 9	**0.863 4**	0.822 5	0.823 7	**0.828 2**	0.803 6	0.803 8	0.801 2
Vowel	0.942 6	0.945 3	**0.946 9**	0.947 1	0.962 1	**0.967 3**	0.837 8	0.835 3	**0.840 8**	0.954 6	0.957 7	**0.957 9**
Glass	0.775 4	0.770 6	**0.796 3**	0.797 2	0.786 4	**0.804 5**	0.792 6	0.797 3	**0.799 1**	0.817 5	0.818 4	0.810 2
Balance	0.412 9	0.602 9	0.593 5	0.488 4	0.472 3	0.475 0	0.536 5	0.581 6	0.545 2	0.672 3	0.669 5	**0.677 3**
Abalone	0.656 3	0.680 4	**0.700 8**	0.577 7	0.607 7	**0.611 8**	0.636 7	0.645 4	**0.647 3**	0.818 5	0.821 3	0.820 1
Yeast	0.846 7	0.826 2	**0.854 6**	0.834 9	0.845 6	0.839 9	0.871 7	0.872 4	**0.874 8**	0.930 3	0.919 9	**0.933 1**
Letter	0.860 1	0.850 9	**0.865 1**	0.830 7	0.827 1	**0.849 1**	0.873 1	0.873 9	0.871 4	0.934 1	0.929 4	0.930 5
Avg.	0.729 4	0.747 6	**0.755 8**	0.736 9	0.740 4	**0.747 5**	0.738 5	0.743 4	**0.749 2**	0.809 7	0.795 9	**0.815 7**

Table 6 AUC results on *Pima* and *Breast* with five oversampling rates

AUC		SMOTE				WPO				RO			
		CART	KNN	NB	NN	CART	KNN	NB	NN	CART	KNN	NB	NN
Pima	100%	0.683 9	0.677 1	0.673 1	0.723 9	**0.675 6**	**0.704 2**	**0.698 0**	**0.748 4**	0.667 4	0.654 2	0.690 2	0.725 1
	200%	0.663 0	0.663 6	0.713 3	0.746 6	**0.736 1**	**0.664 8**	0.665 6	0.693 3	0.708 7	0.617 3	0.682 2	0.687 1
	300%	0.697 3	0.667 9	0.709 7	0.751 3	**0.746 5**	**0.671 8**	**0.719 9**	**0.767 7**	0.725 5	0.653 5	0.701 4	0.749 5
	400%	0.682 4	0.628 1	0.707 2	0.781 4	**0.702 6**	**0.681 3**	0.689 6	**0.795 0**	0.678 1	0.674 4	0.683 1	0.736 5
	500%	0.678 6	0.660 2	0.694 4	0.707 8	**0.683 6**	**0.685 0**	**0.712 8**	**0.798 9**	0.633 2	0.645 7	0.703 7	0.770 4
Breast	100%	0.912 0	0.931 2	0.927 5	0.921 0	0.911 6	**0.933 2**	0.924 8	**0.922 0**	0.886 5	0.918 8	0.911 7	0.912 3
	200%	0.905 4	0.933 7	0.928 6	0.915 8	0.901 8	**0.936 2**	**0.929 9**	**0.924 0**	0.900 2	0.920 3	0.926 8	0.915 1
	300%	0.924 5	0.933 0	0.932 7	0.903 0	**0.949 1**	0.930 4	0.922 8	**0.890 4**	0.931 5	0.928 5	0.923 0	0.890 2
	400%	0.895 5	0.945 1	0.922 3	0.905 1	0.893 6	0.936 7	**0.924 1**	0.901 7	0.890 0	0.940 5	0.913 6	0.901 9
	500%	0.930 4	0.972 7	0.924 2	0.922 8	0.900 1	0.936 4	**0.924 7**	**0.923 0**	0.881 0	0.937 3	0.918 7	0.922 1

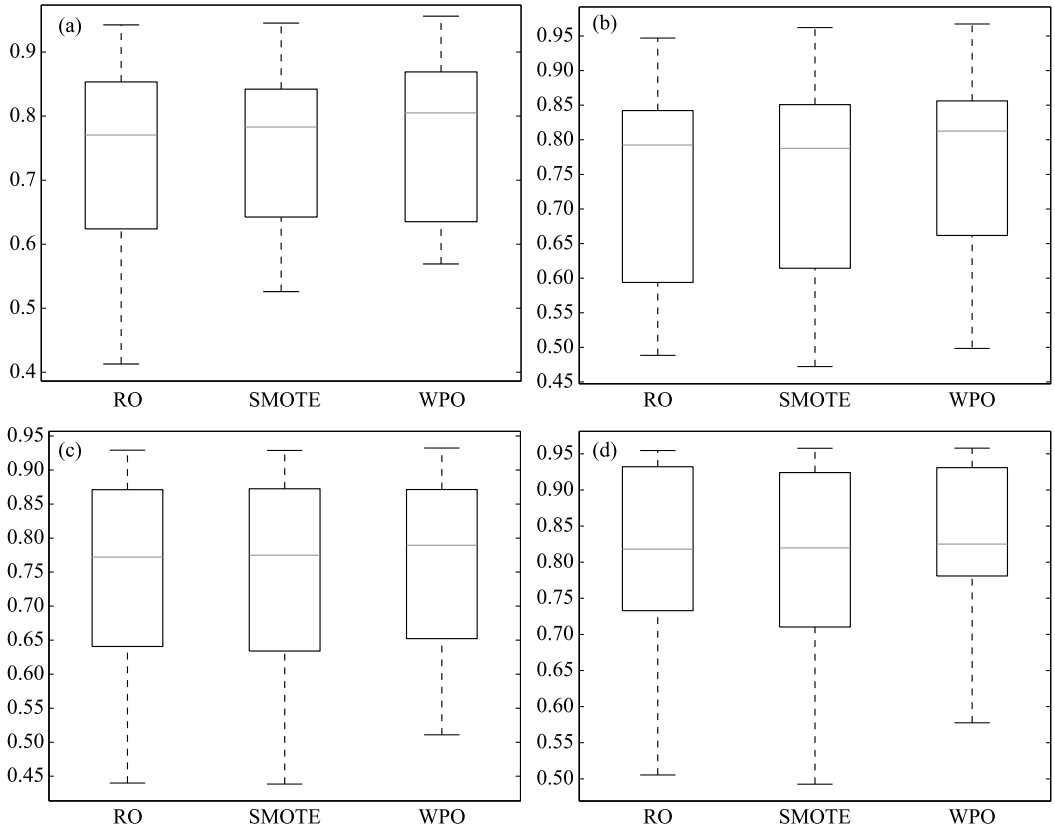


Fig. 2 The boxplot depicting the AUC distribution on sixteen datasets with single classifier. (a) AUC values distribution (CART); (b) AUC values distribution (KNN); (c) AUC values distribution (NB); (d) AUC values distribution (NN)

further demonstrates separate AUC values under five different oversampling rates, which is consistent with the average results of Table 5. To analyze the most significant metric AUC, we also build the non-parametric statistical visual plot, boxplot, averaging the AUC of four datasets.

The grey line in the box of Fig. 2 specifies the statistical median of the corresponding AUC distribution. It is evident that the median of WPO is significant different from RO and SMOTE.

Above results demonstrate the average performances among WPO, SMOTE and RO. Table 7 using the *t*-test at confidence level of 95% to assess whether the means of WPO are statistically different from SMOTE in CART. We set the null hypothesis as that the average corresponding result of WPO is smaller than SMOTE's. The results use the form "Worse-TIE-Better" when comparing WPO with RO or SMOTE. For instance, the first item "2-2-12" of RO in CART indicates that WPO is significantly better than RO in 12 data sets, worse in two data sets and not significant difference in two datasets. It shows that most results from WPO are significant higher than from SMOTE. In summary, these observations imply that integration WPO with single classifiers achieves supe-

riority performance than SMOTE and RO.

Table 7 The *t*-test of three metrics on 16 data sets with significant level at 95%

t-test		F-measure	G-mean	AUC
RO	C	2-2-12	2-1-13	2-2-12
	K	4-2-10	4-1-11	3-2-11
	B	3-2-11	3-1-12	2-3-11
	N	2-3-11	2-2-12	2-1-13
SMOTE	C	2-2-12	2-4-10	1-3-12
	K	4-3-9	3-5-8	3-4-9
	B	2-4-10	3-2-11	3-2-11
	N	3-2-11	4-2-10	3-3-10

Note: C, K, B and N represent the classifier CART, KNN, Naive Bayes and Neural Network respectively. The results use the form "Worse-Tie-Better" when compared WPO with RO or SMOTE

4.3 Evaluating the improvement on ensemble methods

This part will test the effectiveness of WPO in AdaBoost and compare the proposed WPOBoost with other imbalanced-learning strategies.

4.3.1 Performance analysis of WPOBoost and WPOAdaCART

Since single classifiers Neural Network and CART show better performance in the previous experiment, we use them as the base learners respectively in AdaBoost. Figures 3 and 4 show the results of using Neural Network as the basic classifiers, which are also performed 10-fold cross validation.

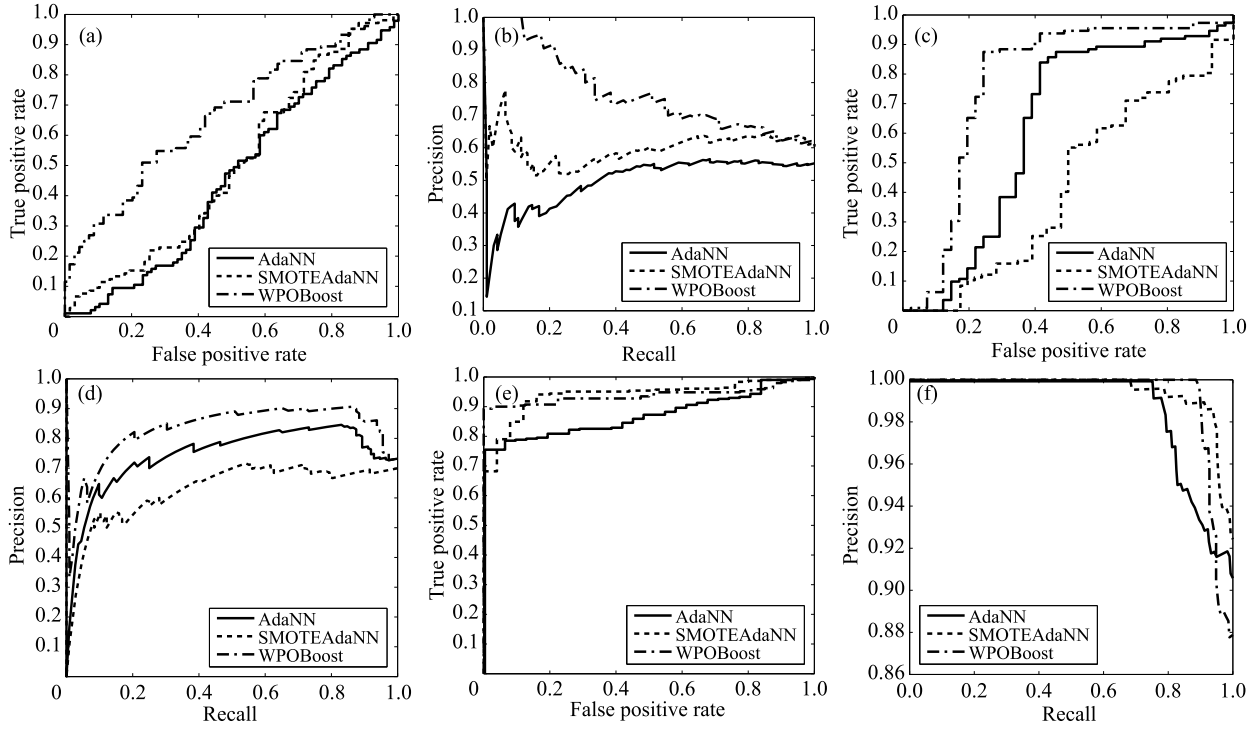


Fig. 3 The PR and ROC curves of AdaNN, SMOTEAdaNN and WPOBoost. (a) ROC curve on Bupa; (b) PR curve on Bupa; (c) ROC curve on Haberman; (d) PR curve on Haberman; (e) ROC curve on Vowel; (f) PR curve on Vowel

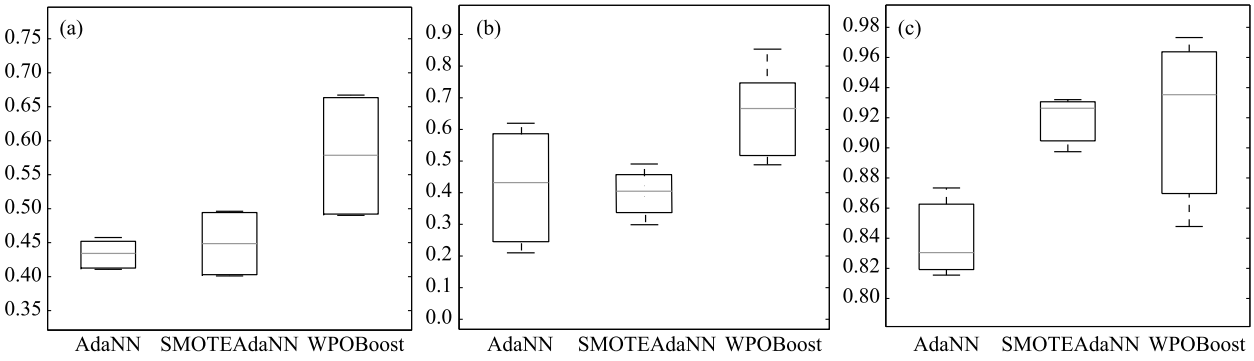


Fig. 4 The boxplot depicting the AUC distribution with AdaNN, SMOTEAdaNN and WPOBoost. (a) AUC values distribution on Bupa; (b) AUC values distribution on Haberman; (c) AUC values distribution on Vowel

All three ROC curves of WPOBoost in Fig. 3(a), Fig. 3(c) and Fig. 3(e) achieve high AUC values, which are also evident in Fig. 4. More specifically, the lines of our method WPOBoost achieves at least 20% improvement over SMOTEAdaNN and AdaNN, as shown in Fig. 3(a) and Fig. 3(c).

Precision-Recall (PR) curves in Fig. 3(b), Fig. 3(d) and Fig. 3(f) present the evaluation in the ROC space. Though Recall and Precision goals are usually conflicting, the curve of WPOBoost in Fig. 3(b) and Fig. 3(d) increases the Recall but sacrifices less *Precision* than *AdaNN* and SMOTEAdaNN. Thus, the Precison of WPOBoost is more steady and less sensitive to the variants of Recall. Hence, we could conclude that WPOBoost achieves effective performance than AdaNN and SMOTEAdaNN in both the ROC and the PR.

Similarly, for CART as the basic learner of AdaBoost, Fig. 5 shows the ROC and PR curves on datasets Pima, Balance and Yeast. It is clear that both WPOAdaCART and SMOTEBoost achieve higher ROC and PR than AdaCART. More specifically, *WPOAdaCART* outperforms *SMOTEBoost* with high PR values in Figs. 5(b), 5(d) and 5(f). Figure 6 depicts the AUC distribution by boxplot and further verifies averagely better performance of WPOAdaCART.

In conclusion, the excellent performance of WPOBoost

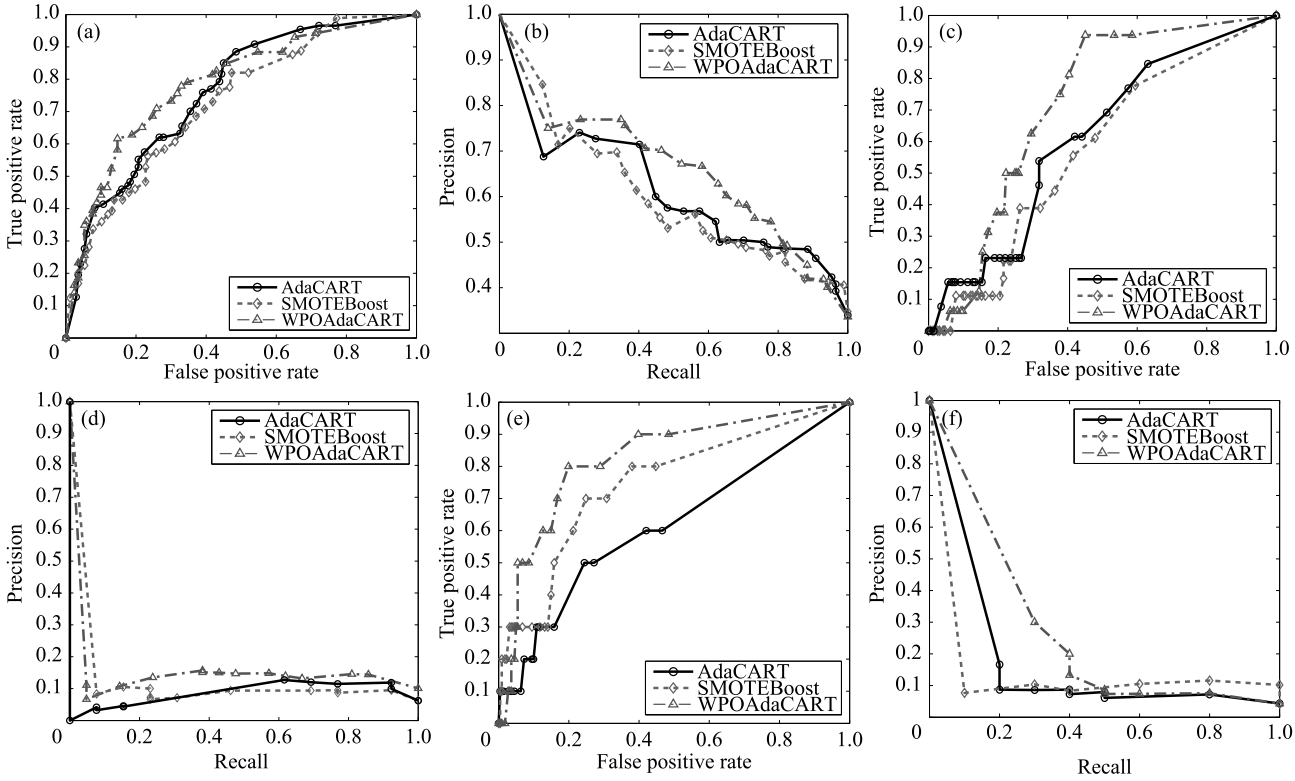


Fig. 5 The PR and ROC curves with AdaCART, SMOTEBoost and WPOAdaCART. (a) ROC curve on Pima; (b) PR curve on Pima; (c) ROC curve on Balance; (d) PR curve on Balance; (e) ROC curve on Yeast; (f) PR curve on Yeast

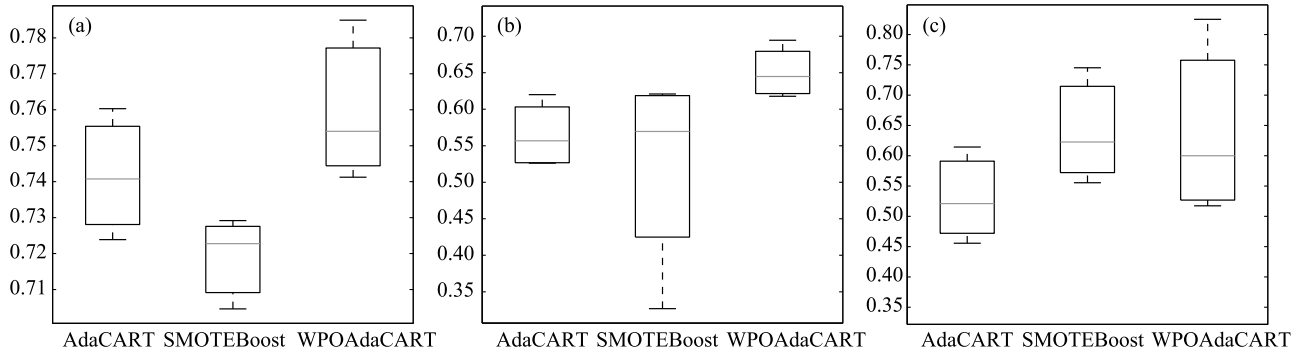


Fig. 6 The boxplot depicting the AUC distribution with AdaCART, SMOTEBoost and WPOAdaCART. (a) AUC values distribution on Pima; (b) AUC values distribution on Balance; (c) AUC values distribution on Yeast

and WPOAdaCART verify that integrating oversampling into *AdaBoost* algorithm can consistently enhance the capability of solving imbalanced issue.

4.3.2 Performance analysis of five state-of-the-art imbalanced-learning approaches

To verify the performance of proposed WPOBoost, we further compare it with other strategies. Figure 7 demonstrates the ROC and PR curves of five state-of-the-art imbalanced-learning approaches on datasets *Glass* and *Ecoli*. Figure 8 depicts the AUC using *boxplot*.

It is apparent that the Bagging performs the worst under the same settings in Figs. 7 and 8. The lost of the potential information of minority class by under-sampling leads to the deteriorated performance of Bagging. In contrast, the corre-

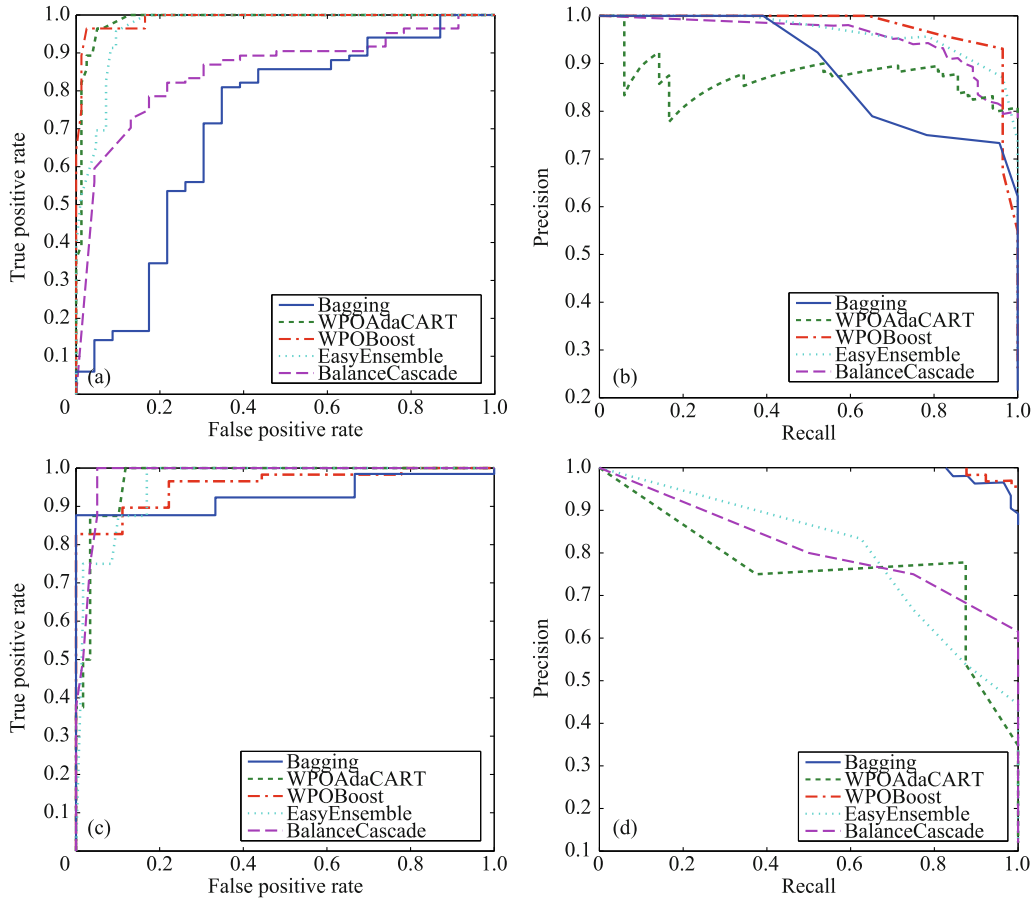


Fig. 7 The PR and ROC curves with five ensemble approaches. (a) ROC curve on Glass; (b) PR curve on Glass; (c) ROC curve on Ecoli; (d) PR curve on Ecoli

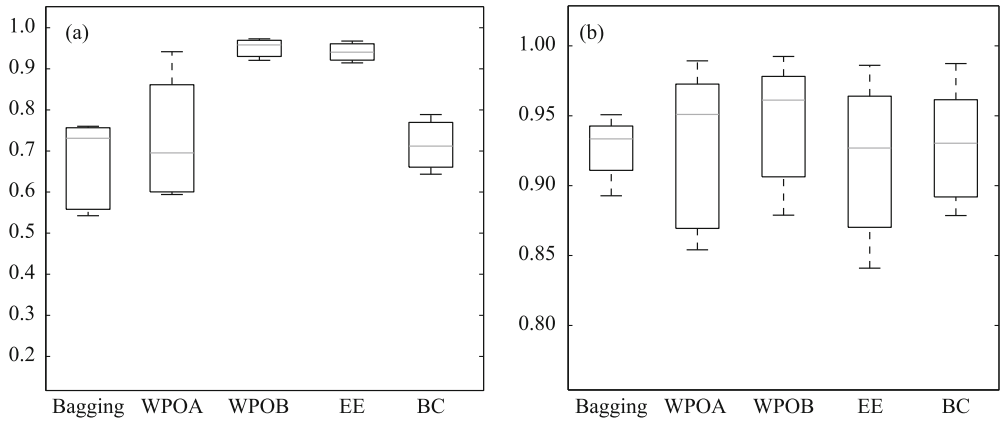


Fig. 8 The boxplot depicting the AUC distribution with five ensemble approaches. From left to right, the histograms represent Bagging, WPOAdaCART, WPOBoost, EasyEnsemble and BalanceCascade. (a) AUC values distribution on Glass; (b) AUC values distribution on Ecoli

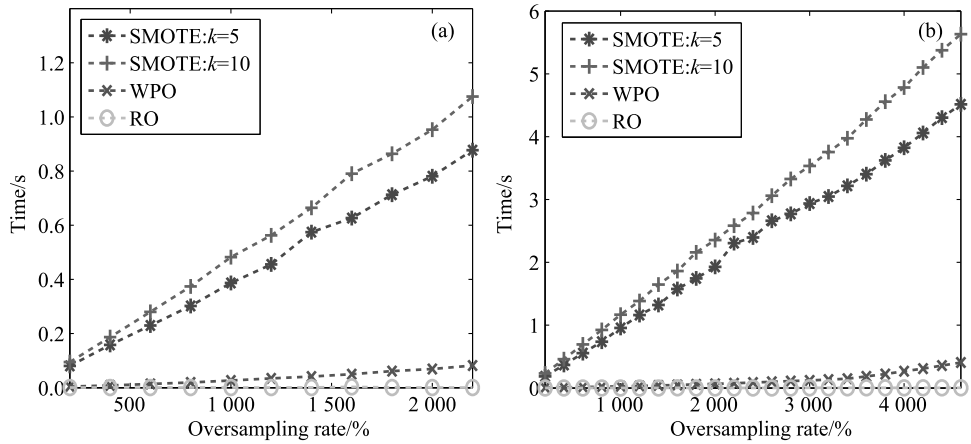


Fig. 9 The oversampling time cost. The size of minority class n is (a) 2 300 and (b) 4 600

sponding point of WPOBoost in ROC is closer to the $(0, 1)$ point than the other, with the average AUC close to 0.95 in Fig. 7(a) for dataset Glass. It is clear that the distribution of AUC values for WPOBoost is much more favorable in Figs. 7(c) and 7(a). EasyEnsemble, BalanceCascade and WPOAdaCART follow the same trend but perform worse than WPOBoost. In addition, Figs. 7(d) and 7(b) show the Precision-Recall curves and clearly show the significance improvement over the other four methods.

From these observations, we can conclude that WPOBoost algorithm that integrates WPO with AdaNN outperforms most prevalent approaches.

4.4 Analysis of oversampling time

Figures 9(a) and 9(b) explore the time cost of RO, SMOTE and WPO, with various oversampling rate. The size of minority class n is 2 300 in Fig. 9(a) and 4 600 in Fig. 9(b). The SMOTE algorithm has a parameter k that determines the searching space when generating new samples. We assign k with two common using values 5 and 10 and compare with RO and WPO under the same settings. In Figs. 9(a) and 9(b), the oversampling rate $r_{\max}$ increases 200% each time.

From Figs. 9(a) and 9(b) it is clear that the time costs of SMOTE and WPO grow as the oversampling rate grows. SMOTE consumes the most time compared with RO and WPO in both figures. The overall complexity in SMOTE is theoretically specified by $k \cdot n + k \cdot r$, which is approximately $O(k \cdot r)$ when n and k are predefined, as shown in Fig. 9(a) and Fig. 9(b). In contrast, RO and WPO consume less time than SMOTE. The RO oversamples the samples by randomly choosing from existing samples, and the running time of RO is irrelevant with the oversampling rate, so it is steady in both Figs. 9(a) and 9(b).

However, the time saved in RO results in deteriorated performance when compared with WPO and SMOTE. As we mentioned previously, the complexity of generating new samples is $O(1)$, which is independent from the number of original samples. Consequently, the oversampling time of WPO increases as the oversampling rate varies from 200 to 4600. In conclusion, With the running time close to RO but much shorter than SMOTE, WPO achieves much superiority capability of tackling imbalanced datasets.

5 Conclusion

Learning from imbalanced data has aroused significant research efforts in recent years. Since undersampling may discard useful information in original dataset, oversampling has been proposed to generate synthetic minority samples by replication or synthesizing. Nevertheless, oversampling constraints the values of new samples within the range of original training dataset, which further results in the tremble of the decision boundary. Moreover, the involved time complexity usually make it inefficient to incorporate oversampling with classifiers.

This paper adopted the *Wiener process*, and developed a novel over-sampling technique Wiener Process Oversampling, which has two advantages:

- Firstly, theoretic proof shows that the new samples generated by WPO conform to the distribution of the minority class while broadening the decision region. This lays the foundation for the unbiased and robust classifier. Experimental results indicate that WPO consistently enhances the capability of classifiers, with improved performance over SMOTE. Especially, by integrating WPO with AdaNN, WPOBoost outperforms all compared ensemble learning approaches.

- Secondly, as WPO can be implemented with linear complexity, WPO is more efficient than common over-sampling. Consequently, it is feasible to efficiently integrate WPO with any classifier for imbalanced data learning.

Acknowledgements This research was partially supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030200), the National Natural Science Foundation of China (Grant Nos. M1552006, 61403369, 61272427, and 61363030), Xinjiang Uygur Autonomous Region Science and Technology Project (201230123), Beijing Key Lab of Intelligent Telecommunication Software, Multimedia (ITSM201502), Guangxi Key Laboratory of Trusted Software (kx201418).

References

1. Zhou Z H, Liu X Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(1): 63–77
2. Liu X Y, Zhou Z H. The influence of class imbalance on cost-sensitive learning: an empirical study. In: Proceedings of the 6th International Conference on Data Mining. 2006, 970–974
3. Yu L, Wang S, Lai K K. Developing an svm-based ensemble learning system for customer risk identification collaborating with customer relationship management. Frontiers of Computer Science in China, 2010, 4(2): 196–203
4. Liu E, Zhao H, Guo F, Liang J, Tian J. Fingerprint segmentation based on an adaboost classifier. Frontiers of Computer Science in China, 2011, 5(2): 148–157
5. Han H, Wang W, Mao B. Over-sampling algorithm based on adaboost in unbalanced data set. Computer Engineering, 2007, 33(10): 207–209 (in Chinese)
6. Chawla N V, Lazarevic A, Hall L O, Bowyer K W. Smoteboost: improving prediction of the minority class in boosting. Lecture Notes in Computer Science, 2003, 2838: 107–119
7. Mease D, Wyner A J, Buja A. Boosted classification trees and class probability/quantile estimation. The Journal of Machine Learning Research, 2007, 8: 409–439
8. Batista G E, Prati R C, Monard M C. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 20–29
9. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-levelsmote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Proceedings of Advances in Knowledge Discovery and Data Mining. 2009, 475–482
10. Chawla N V, Bowyer K W, Hall L O, Kegelmeyer W P. Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 2002, 321–357
11. Yuan B, Liu W. Measure oriented training: a targeted approach to imbalanced classification problems. Frontiers of Computer Science, 2012, 6(5): 489–497
12. Kang P, Cho S. EUS SVMS: ensemble of under-sampled svms for data imbalance problems. In: Proceedings of Neural Information Processing. 2006, 837–846
13. Japkowicz N. The class imbalance problem: significance and strategies. In: Proceedings of International Conference on Artificial Intelligence. 2000
14. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, 2012, 42(4): 463–484
15. Yuan B, Ma X. Sampling+ reweighting: boosting the performance of adaboost on imbalanced datasets. In: Proceedings of International Joint Conference on Neural Networks. 2012, 1–6
16. Hida T. Brownian motion. Springer US, 1980, 11(5): 44–113
17. Dietterich T G. Ensemble methods in machine learning. In: Proceedings of Multiple classifier systems. 2000, 1–15
18. Maloof M A. Learning when data sets are imbalanced and when costs are unequal and unknown. In: Proceedings of ICML-2003 Workshop on Learning from Imbalanced Data Sets II. 2003
19. Chawla N V, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 1–6
20. Han H, Wang W Y, Mao B H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Proceedings of Advances in Intelligent Computing. 2005, 878–887
21. Liu X Y, Wu J, Zhou Z H. Exploratory undersampling for classimbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, 2009, 39(2): 539–550
22. Schapire R E. The boosting approach to machine learning: an overview. Nonlinear Estimation and Classification, 2003, 149–171
23. Schapire R E, Singer Y. Boostexter: a boosting-based system for text categorization. Machine Learning, 2000, 39(2–3): 135–168
24. Li X, Wang L, Sung E. Adaboost with svm-based component classifiers. Engineering Applications of Artificial Intelligence, 2008, 21(5): 785–795
25. Boyd S, Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press, 2004
26. Asuncion A, Newman D. UCI machine learning repository. http://www.ics.uci.edu/mlearn/MLRepository.html, 2007
27. Breiman L, Friedman J, Stone C J, Olshen R A. Classification and Regression Trees. Belmont: Wadsworth International Group, 1984
28. Lewis D D. Naive (Bayes) at forty: the independence assumption in information retrieval. In: Proceedings of Machine Learning: ECML-98. 1998, 4–15
29. Keller J M, Gray M R, Givens J A. A fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man and Cybernetics, 1985 (4): 580–585
30. Breiman L. Bagging predictors. Machine Learning, 1996, 24(2): 123–140

Qian Li received her MS at Shandong University of Computer Software and Theory, China. Now she is a PhD student of Institute of Information Engineering, Chinese Academy of Sciences, China. Her main research interests include machine learning, data mining and services computing.

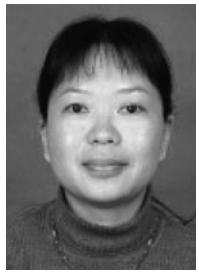
Gang Li is currently a senior lecturer in the School of Information Technology at Deakin University, Australia. His research interest are in the area of data mining, machine learning and multimedia analysis. He served on the Program Committee for over 40 international conferences in artificial intelligence, data mining and machine learning, tourism and hospitality management.

Wenjia Niu is an associate professor in the Institute of Information Engineering, Chinese Academy of Sciences, China. His research interests include Web services, agent, sensor network and data mining. He has served as a regular reviewer for Journal of Network and Computer Applications (JNCA), Knowledge and Information Systems (KAIS), and Journal of Computer Science and Technology (JCST).

Yanan Cao is an associate professor in the Institute of Information Engineering, Academy of Sciences, China. She obtained her PhD in the Institute of Computing Technology in 2012. Her research interests include data mining methodologies, machine learning algorithms and knowledge graph.

Liang Chang received his PhD in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, China in 2008. He is currently a professor in the School of Computer Science and Engineering, Guilin University of Electronic Technology, China. His research interests include knowledge representation and reasoning, formal methods, trusted software and intelligent planning.

Jianlong Tan is a researcher in the Institute of Information Engineering, Chinese Academy of Sciences, China. He is also the chairman of the Intelligent Information Processing Research Center, Institute of Information Engineering, Chinese Academy of Sciences. His research interests are string matching algorithm, algorithm security and information security.



Li Guo is a researcher in the Institute of Information Engineering, Chinese Academy of Sciences, China. Her research interests include data stream management systems and information security.