Composite reduced-kernel weighted extreme learning machine for imbalanced data classification

Dafei Wang*, Wujie Xie, and Wenhan Dong
Aeronautics Engineering College, Air Force Engineering University, Xi’an, Shanxi, 710038, China
*Corresponding author’s e-mail: 746970090@qq.com

Abstract. In order to solving the problem that the weighted extreme learning machine based on the ensemble learning method enhances the classification performance while increasing the running time of the algorithm, starting from the perspective of multi-core learning, a weighted extreme learning machine based on composite kernel functions and reduced-kernel technique is proposed. The composite kernel function based on Gaussian kernel and Polynomial kernel weighted combination is designed, which effectively improves the classification performance of weighted extreme learning machine. Meanwhile, based on the sample distribution characteristics of the imbalanced dataset, the balanced input sub-matrix is designed to reduce the computational cost of the composite kernel method. The eight binary classification imbalanced datasets of KEEL dataset repository were used for testing. The experimental results show that compared with the original weighted extreme learning machine algorithm, the G-mean and AUC classification performance indicators of the composite reduced-kernel weighted extreme learning machine algorithm are improved in each dataset, and the computation cost is effectively reduced.

1. Introduction

Extreme learning machine (ELM) is a kind of single hidden layer feedforward neural networks (SLFNs) proposed by Huang et al. [1] in 2004. ELM randomly initializes the connection weights between the input layer and the hidden layer as well as the offset values of each hidden layer node, and use analysis method to calculate the connection weights between the hidden layer and the output layer, which makes it obviously faster than the traditional neural network based on the gradient descent methods[2]. Huang et al.[3] proposed an optimized version of ELM based on constrained optimization theory in 2012, and introduced the kernel method into the solution of ELM problem, which further developing and perfecting the theory of ELM. In recent years, more and more extensive and in-depth theoretical and applied research on ELM has made it an efficient tool for handling classification and regression problems in the field of machine learning and data mining[4].

The traditional extreme learning machine algorithm does not consider the influence of the class imbalance of the training dataset on the classification problem[5], Zong et al. [6] proposed a weighted extreme learning machine(WELM) algorithm based on the idea of cost-sensitive learning, the method of changing the penalty factor by assigning different weights to different classes of samples during the model training process, effectively reducing the probability of minority samples being misclassified [7]. Later, many improved versions of the WELM algorithm were proposed for more efficient resolution of imbalanced data classification problems. Among them, the boosting weighted extreme learning machine(BWELM) algorithm proposed by Li et al. [8] improves the performance of WELM
through the integration method. BWELM is based on an improved AdaBoost framework and uses WELM as the base classifier to generate a sequential method similar to AdaBoost. Unlike the traditional AdaBoost algorithm, the base classifier re-adjusts the weight of each class sample at the end of each iteration to make the total weighting in the update setting of each sample weight assignment. The allocation is balanced and finally the weighted voting is used to calculate the integrated output. However, BWELM has greatly increased the computational cost while improving the performance of WELM classification. It has been shown in [8] that the training time of BWELM algorithm is about 10-50 times of WELM. DEBEWELM (differential evolution based ensemble weighted extreme learning machine) algorithm Zhang et al. [9] uses the heterogeneous integration method to improve the classification performance of WELM. It uses WELM based on different activation functions as the base classifier. The weight distribution is calculated by the differential evolution method, and the final integrated output result is also obtained by weighted voting. The DEBEWELM algorithm is prone to information redundancy and also greatly increases computational cost.

The WELM algorithm based on integrated learning generally increases the time consumption while improving the classification performance. To design an algorithm that can improve the performance of WELM classification without greatly increasing the computational cost, this paper attempts to learn from the perspective of multi-core learning. Starting with Gaussian kernel and polynomial kernel weighted synthesis method to improve the performance of WELM algorithm, combined with the characteristics of imbalanced dataset, designed a method to construct a balanced training subset using random downsampling algorithm, and then performing reduced-kernel technique to reduce the computation time increase caused by the composite kernel method.

2. Related work

2.1. Kernelized ELM

Given the training dataset $D = \{(x_i, t_i) | x_i \in \mathbb{R}^n, t_i \in \mathbb{R}, i = 1, 2, \ldots, N\}$, the number of hidden layer nodes $L$ of the ELM, and the hidden layer activation function $G(x)$, define the following symbols:

$$\beta_L = [\beta_{11}, \beta_{12}, \ldots, \beta_{1m}]^T$$

$$h(x) = [h_1(x), h_2(x), \ldots, h_L(x)]$$

$$T = \begin{bmatrix} t_{11}^T \\ \vdots \\ t_{N1} \end{bmatrix} = \begin{bmatrix} t_{11} \\ \vdots \\ t_{1m} \\ \vdots \\ \vdots \\ t_{N1} \ldots t_{Nm} \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_{11} \\ \vdots \\ \beta_{1L} \\ \vdots \\ \vdots \\ \beta_{L1} \ldots \beta_{Lm} \end{bmatrix}$$

$$X = \begin{bmatrix} x_{11}^T \\ \vdots \\ x_{N1} \end{bmatrix} = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1m} \\ \vdots \\ \vdots \\ x_{N1} \ldots x_{Nm} \end{bmatrix}$$

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) \\ \vdots \\ h_L(x_1) \\ \vdots \\ \vdots \\ h_1(x_N) \ldots h_L(x_N) \end{bmatrix}$$
Here, $X$ is the input matrix of the training dataset, $T$ is the label matrix of the training dataset, $\beta_j=(\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm})^T$ represents the connection weight between the $j$th hidden layer node and the output layer, $\beta$ represents the connection weight between the hidden layer and the output layer; $h_j(x) = G(a_j \cdot x + b_j)$ represents the output function of the $j$th hidden layer node, $a_j$ represents the connection weight of the input layer and the $j$th hidden layer node, $b_j$ represents the offset value of the $j$th hidden layer node, $h(x)$ represents the output function of the output layer, $H$ represents the output matrix of the ELM hidden layer to the training set input matrix $X$.

According to [2], the output function of ELM can be expressed as

$$f(x) = \sum_{j=1}^{L} \beta_j h_j(x) = h(x) \beta$$  \hspace{1cm} (7)$$

Since the connection weight between ELM’s input layer and the hidden layer as well as the offset value of each hidden layer node can be randomly initialized according to any continuous probability distribution function [2], the central problem of ELM training becomes the computation of the connection weight matrix between the hidden layer and the output layer. In [3], a method for transforming this problem into a constrained optimization problem is proposed. By using Lagrangian dual method and solving the KKT condition, we have

$$\beta = H^T \left( \frac{I}{C} + HH^T \right)^{-1} T$$  \hspace{1cm} (8)$$

Here, $C$ is a regularization coefficient, which is used to enhance the generalization performance and stability of ELM, and $I$ is an identity matrix.

Bringing Eq. (8) into Eq. (7), the output function of ELM can be further expressed as

$$f(x) = h(x)H^T \left( \frac{I}{C} + HH^T \right)^{-1} T$$  \hspace{1cm} (9)$$

Introduce the kernel method in Eq. (9), let $K(x,z) = h(x) \cdot h(z)$, define the kernel matrix of the ELM as follow

$$\Omega_{ELM} = HH^T = \begin{bmatrix}
h(x_1) \cdot h(x_1) & \cdots & h(x_1) \cdot h(x_N) \\
\vdots & \ddots & \vdots \\
h(x_N) \cdot h(x_1) & \cdots & h(x_N) \cdot h(x_N)
\end{bmatrix}$$

Then the output function of kernelized extreme learning machine (KELM) can be written as

$$f(x) = \begin{bmatrix}
K(x,x_1) \\
\vdots \\
K(x,x_N)
\end{bmatrix}^T \left( \frac{I}{C} + \Omega_{ELM} \right)^{-1} T$$  \hspace{1cm} (11)$$

2.2. Weighted extreme learning machine

Neither ELM nor KELM considers the impact of class imbalance in training datasets on classifiers. Zong et al. [6] proposed a WELM algorithm that uses weighting methods to deal with imbalanced dataset classification problems. In the case of binary class problem, because the label value of training sample becomes a scalar $t_i \in \{+1,-1\}$, ELM only needs one output node ($m=1$). According to the literature [6], the output function of the binary class WELM can be expressed in kernel form as
\[ f(x) = \text{sign} \left[ K(x, x_1)^T \cdots K(x, x_N) \right] = \left( \frac{I}{C + \Omega_{\text{ELM}}} \right)^{-1} W^T \]

Here, \( W \) is an \( N \times N \) weighted diagonal matrix, document [6] provides two weight allocation schemes

\[
W_1: W_{ii} = \frac{1}{N_i}, \quad \frac{1}{N_i}, N_i > N_{\text{avg}}
\]

\[
W_2: W_{ii} = \begin{cases} 
0.618, & N_i \leq N_{\text{avg}} \\ 
\end{cases}
\]

Here, \( N_i \) represents the number of training samples of class \( i \) corresponding to instance \( x_i \), \( N_{\text{avg}} \) is the average number of training samples for all classes. The basic design principle of the two weight distribution schemes is to increase the weight of the minority sample and reduce the weight of the majority of the sample, so as to alleviate the impact of class imbalance during the training of the ELM model.

3. Proposed composite reduced-kernel weighted extreme learning machine

3.1. Composite-kernel method

Different kernel functions have their own advantages and disadvantages, as well as their respective applicability. The traditional kernelized weighted extreme learning machine uses single-kernel method to train the model. In recent years, theoretical and applied research has proved that using multi-kernel instead of single-kernel can obtain a better performance model. The composite-kernel method is a basic method of multi-kernel learning. It combines the kernel functions with different characteristics to obtain the advantages of multi-class kernel functions, and thus obtains better mapping performance [10]. This paper attempts to transform the Gaussian kernel in the traditional kernelized weighted extreme learning machine algorithm into a composite-kernel method that is weighted combined by Gaussian kernel function and polynomial kernel function.

\[ K = \mu K_G + (1 - \mu) K_p = \begin{bmatrix} \mu \\ 1 - \mu \end{bmatrix} \begin{bmatrix} K_G \\ K_p \end{bmatrix} \]

Here, \( K_G(x, z) = \exp(-\gamma \|x - z\|^2) \) is Gaussian kernel function, \( K_p(x, z) = (x \cdot z + 1)^p \) is polynomial kernel function, \( \gamma \) and \( p \) are the control parameters of the Gaussian kernel function and the polynomial kernel function, respectively. \( \mu \) is the weighted synergy factor between the two kernel functions, with a value range of \([0, 1]\). Gaussian kernel function and polynomial kernel function are two commonly used kernel functions. It is easy to see from the form that Gaussian kernel function has strong local mapping ability relative to the kernel center, and polynomial kernel function has strong global mapping. The weighted combination of these two kernel functions can more effectively exploit the advantages of the two kernel functions, and the feature space mapping that is more conducive to the classification effect is improved for the input space characteristics of different training datasets. However, while the composite-kernel method increases the diversity and effectiveness of spatial mapping, it also increases the computational complexity of the algorithm. As shown in Eq. (15), compared with the single-kernel method, the composite-kernel method should perform the calculation of an additional kernel matrix and the weighted combination of the two kernel matrices, which inevitably increases the time cost of the algorithm.
3.2. Reduced-kernel method designing based on composite-kernel technique

In order to solve the problem of increasing computational complexity caused by the composite-kernel method in 3.1, this paper proposed an algorithm of composite reduced-kernel weighted extreme learning machine (CRKWELM). Given an imbalanced training dataset of a binary class classification problem $D = \{(x_i, t_i) | x_i \in \mathbb{R}^n, t_i \in \{+1,-1\}, i = 1,2,\ldots,N\}$. Here, $N^-$ represents the number of the majority class, $N^+$ represents the number of the minority class, class imbalance ratio $IR = N^-/N^+$. The CRKWELM algorithm uses a random downsampling method to randomly select $N^+$ samples from $N^-$ majority samples and mix them with all minority samples to generate a balanced training subset $D'$. All instances in $D'$ form a class balanced sub-input matrix $X'=[x_1,\ldots,x_n]^T$. Obviously, $N' < N$, after that, in the central point selection process of the kernel matrix calculation, the CRKWELM algorithm uses the sub-input matrix $X'$ instead of $X$ to reduce the computational complexity of the algorithm and improve the classification effect of the minority samples. The specific derivation process of CRKWELM algorithm is as follows.

By removing the regularization term $I/C$ from the output function of KWELM, write kernel matrix $\Omega_{ELM}$ as the kernel function form of the training set input matrix

$$\Omega_{ELM} = K(X, X) = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \cdots & K(x_n, x_n) \end{bmatrix}$$

Then Eq. (12) can be rewritten as

$$f(x) = \text{sign} \left[ \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_n) \end{bmatrix}^T (WK(X, X))^{-1} WT \right]$$

(17)

Here, the computation method of the weight matrix $W$ uses the weight distribution scheme $W_1$ provided in [6]. Substituting the input matrix with a class-balanced sub-input matrix as the center point of the kernel function calculation, we have

$$f(x) = \text{sign} \left[ \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_n) \end{bmatrix}^T (WK(X', X'))_{N\times N}^{-1} WT \right]$$

(18)

Here, $(WK(X, X'))^T$ represents the Moore-Penrose generalized inverse matrix of $WK(X, X')$. Calculate $(WK(X, X'))^T$ according to the method provided in [11], we have

$$(WK(X, X'))^T = \left( (WK(X, X'))^T WK(X, X') \right)^{-1} (WK(X, X'))^T = \left( K(X, X')^T W_2 K(X, X') \right)^{-1} K(X, X')^T W$$

(19)

Bringing Eq. (19) into Eq. (18)
Referring to the ridge regression method [12], in order to enhance the stability and generalization performance of the model, a regularization term $\lambda I$ is added to the inverse matrix part in Eq. (20), and let $W' = W^2$, then Eq. (20) becomes

$$f(x) = \text{sign} \left[ \mu \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_N) \end{bmatrix}^{T} \left( \lambda I + \begin{bmatrix} \mu \\ 1-\mu \end{bmatrix}^{T} \begin{bmatrix} K_G^T \\ K_P^T \end{bmatrix} \right)^{-1} \begin{bmatrix} K_G^{T} \\ K_P^{T} \end{bmatrix} \right] W' \begin{bmatrix} \mu \\ 1-\mu \end{bmatrix}^{T} \begin{bmatrix} \mu \\ 1-\mu \end{bmatrix}^{T} \begin{bmatrix} K_G^T \\ K_P^T \end{bmatrix} W T$$ (23)

The specific steps of CRKWELM algorithm are as follows:

1. According to any continuous probability distribution function, the connection weight $a_j$ between each node of the input layer and the hidden layer and the offset value $b_j$ of each node of the hidden layer are randomly initialized, thereby obtaining the output function $h_i(x) = G(a_j, x + b_j)$ of each hidden layer node and the hidden layer output function $h(x) = [h_1(x), h_2(x), \ldots, h_L(x)]$.

2. Calculate the weighted matrix $W$ and $W' = W^2$ according to Eq. (13).

3. The random downsampling method is used to randomly select $N'^+$ samples from $N^-$ majority samples and mix them with all the minority samples to generate the balanced training subset $D'$. All the samples in $D'$ form a class balanced sub-input matrix $X'= [x_1, \ldots, x_{N'}]^T$.

4. Calculated $k_G, k_P, K_G, K_P$ according to Eq. (15), (16) and (22).

5. Return the connection weight between the ELM hidden layer and the output layer according to Eq. (24)

$$\beta = \left( \lambda I + \begin{bmatrix} \mu \\ 1-\mu \end{bmatrix}^{T} \begin{bmatrix} K_G^T \\ K_P^T \end{bmatrix} \right)^{-1} \begin{bmatrix} K_G^{T} \\ K_P^{T} \end{bmatrix} W' \begin{bmatrix} \mu \\ 1-\mu \end{bmatrix}^{T} \begin{bmatrix} \mu \\ 1-\mu \end{bmatrix}^{T} \begin{bmatrix} K_G^T \\ K_P^T \end{bmatrix} W T$$ (24)
(6) Return the output function of the binary class CRKWELM according to Eq. (25)

\[
f(x) = \text{sign} \left[ \mu \left( k_G + k_P \right)^T \beta \right]
\]

(25)

4. Experimental setup and result analysis

4.1. Experimental setup

In this paper, eight binary imbalanced datasets randomly selected from the KEEL data depositary [13] are used to compare and test the algorithm. Most of the selected datasets have different sample numbers, feature numbers and class imbalance ratio. The specific information of the selected dataset is shown in Table 1.

| No. | Dataset       | #Samples | #Features | IR    |
|-----|---------------|----------|-----------|-------|
| 1   | glass2        | 214      | 9         | 11.59 |
| 2   | yeast1vs7     | 459      | 7         | 14.3  |
| 3   | wisconsin     | 683      | 9         | 1.86  |
| 4   | pima          | 768      | 8         | 1.87  |
| 5   | yeast1289vs7  | 947      | 8         | 30.57 |
| 6   | yeast4        | 1484     | 8         | 28.1  |
| 7   | segment0      | 2308     | 19        | 6.02  |
| 8   | abalone19     | 4174     | 8         | 129.4 |

The hardware environment of this experiment is Intel i5-6200U 4 core CPU, the main frequency is 2.30GHz, the memory is 8GB, the operating system is Win10 64 bit, the compilation environment is Python 3.6.

In order to comprehensively evaluate the classification performance and computational efficiency of the proposed algorithm, it is compared with ELM, WELM, RKWELM (reduced-kernel weighted extreme learning machine), CKWELM (composite-kernel weighted extreme learning machine). For the parameter setting, refer to the method of [6], set a uniform value range and use the grid search method to find the optimal parameter combination for the specific classification performance index. The specific parameter setting range is shown in Table 2. Meanwhile, in order to reflect the fairness of the experiment, a 10-fold cross-check method was used and the average of 10 results was taken as the final experimental result.

| Parameter   | Description                              | Range                                      |
|-------------|------------------------------------------|--------------------------------------------|
| L           | ELM hidden layer nodes                   | \{10, 20, ..., 1000\}                      |
| \(\lambda\) | ELM regularization factor                | \{2^{-25}, 2^{-24}, ..., 2^{24}, 2^{25}\}  |
| \(\gamma\) | Gaussian kernel function parameter       | \{2^{-25}, 2^{-24}, ..., 2^{24}, 2^{25}\}  |
| \(\mu\)    | Composite-kernel weighted factor         | \{0, 0.02, ..., 0.98, 1\}                 |
| \(p\)      | Polynomial kernel function parameter     | \{0, 0.25, ..., 9.75, 10\}                |
According to the characteristics of the classification performance evaluation index and the imbalanced data classification problem, referring to the [14], this paper uses G-mean and AUC as the main indicators of the algorithm classification performance evaluation, and the overall classification accuracy Accuracy as the reference index. At the same time, this paper also uses training time as an important indicator to evaluate the performance of the algorithm.

4.2. Experimental results

![Figure 1. Comparison of various algorithms on Accuracy index.](image1)

![Figure 2. Comparison of various algorithms on G-mean index.](image2)

![Figure 3. Comparison of various algorithms on AUC index.](image3)
It can be seen from Fig. 1 to Fig. 3 that compared with the original extreme learning machine algorithm, the Accuracy indicators of various weighted extreme learning machine algorithms are reduced in the experimentally selected datasets, especially the use of only the kernel reduction technology. The RKWELM algorithm is seriously degraded. The main reason is that the traditional ELM algorithm is mainly based on the minimization of the overall classification accuracy. The weighted method focuses on pursuing the balance of classification performance in different categories, especially the method of reducing the kernel due to the random drop of most samples. Sampling will inevitably reduce the overall classification accuracy of the sample. While the G-mean and AUC indicators of various weighted extreme learning machine algorithms are improved on each dataset, which indicates that the weighted method can offset the impact of imbalanced distribution of different class to some extent.

In the comparison of G-mean and AUC classification performance indexes of several weighted extreme learning machine algorithms on each dataset, compared with the original weighted extreme learning machine algorithm, CKWELM algorithm using only composite kernel technology has the largest amplitude of the classification performance, RKWELM algorithm using only the reduced-kernel technology has a degraded classification performance. At the same time, the classification performance of the CRKWELM algorithm using the two techniques has improved to a certain extent on most datasets.

| No. | Datasets       | Training time (s) |
|-----|----------------|-------------------|
|     |                | ELM   | WELM  | RKWELM | CKWELM | CRKWELM |
| 1   | glass2         | 0.002 | 0.008 | 0.0008 | 0.032  | 0.0025  |
| 2   | yeast1vs7      | 0.011 | 0.027 | 0.0013 | 0.109  | 0.0072  |
| 3   | wisconsin      | 0.029 | 0.078 | 0.0027 | 0.295  | 0.0087  |
| 4   | pima           | 0.031 | 0.08  | 0.0034 | 0.334  | 0.0072  |
| 5   | yeast1289vs7   | 0.056 | 0.175 | 0.0051 | 0.672  | 0.0091  |
| 6   | yeast4         | 0.132 | 0.492 | 0.0143 | 2.056  | 0.0256  |
| 7   | segment0       | 0.498 | 1.669 | 0.0412 | 7.042  | 0.091   |
| 8   | abalone19      | 1.602 | 8.938 | 0.1795 | 40.72  | 0.4513  |

In summary, compared with the original WELM algorithm, the CRKWELM algorithm proposed in this paper has different degrees of improvement in the common performance evaluation indexes of G-
mean and AUC on imbalanced data classification, and And effectively reduce the computational time increase caused by the composite kernel method.

5. Conclusion
In this paper, a composite reduced-kernel weighted extreme learning machine (CRKWELM) algorithm is proposed. A composite kernel method based on polynomial kernel function and Gaussian kernel function is designed. Weighted synergy factor is introduced to adjust the weight distribution of two kernel functions. It can better utilize the advantages of the two kernel functions and has better feature space mapping ability. To reduce the time consumption of the algorithm caused by the composite kernel method, this paper proposes a reduced-kernel method based on the sub-input matrix of the balancing class. The experimental results show that, in most cases, the G-mean and AUC indexes of the CRKWELM algorithm are better than the original WELM algorithm, and the time consumption of the algorithm is also reduced to some extent. However, the algorithm involves more parameters, how to carry out more efficient and accurate parameter optimization is the main direction of our further work.

References
[1] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: a new learning scheme of feedforward neural networks[C] //Proceedings of the 2004 IEEE International Joint Conference, 2004: 985-990.
[2] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: theory and applications[J]. Neurocomputing, 2006, 70(13): 489-501.
[3] Huang G B, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(2): 513-529.
[4] Huang G, Huang G B, Song S, et al. Trends in extreme learning machines: A review[J]. Neural Networks, 2015, 61: 32-48.
[5] Janakiraman V M, Nguyen X L, Sterngi J, et al. Identification of the dynamic operating envelope of HCCI engines using class imbalance learning[J]. IEEE transactions on neural networks and learning systems, 2015, 26(1): 98-112.
[6] Zong W, Huang G B, Chen Y. Weighted extreme learning machine for imbalance learning[J]. Neurocomputing, 2013, 101: 229-242.
[7] Yu H L, Qi Y S, Yang X B, et al. Research on class imbalance fuzzy weighted extreme learning machine algorithm[J]. Journal of Frontiers of Computer Science and Technology, 2017, 11(4): 619-632.
[8] Li K, Kong X, Lu Z, et al. Boosting weighted ELM for imbalanced learning[J]. Neurocomputing, 2014, 128: 15-21.
[9] Zhang Y, Liu B, Cai J, et al. Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution[J]. Neural Computing and Applications, 2017, 28(1): 259-267.
[10] Wang H Q, Sun F C, Cai Y N, et al. On Multiple Kernel Learning Methods[J]. ACTA AUTOMATICA SINICA, 2010, 36(8): 1037-1050.
[11] Courri€e P. Fast computation of Moore-Penrose inverse matrices[J]. arXiv preprint: 0804.4809, 2008.
[12] Hoerl A E, Kennard R W. Ridge regression: biased estimation for nonorthogonal problems[J]. Technometrics, 2000, 42(1): 80-86.
[13] Alcalá-Fdez J, Fernández A, Luengo J, et al. Keel data-mining software tool: dataset repository, integration of algorithms and experimental analysis framework[J]. Journal of Multiple-Valued Logic & Soft Computing, 2011, 17.
[14] Yu H L. Class imbalance learning: theory and algorithm[M]. Beijing: Tsinghua University Press, 2017.