

## 基于 CSD-ELM 的不平衡数据分类算法

王大飞<sup>a</sup>, 解武杰<sup>b</sup>, 董文瀚<sup>b</sup>

(空军工程大学 a. 研究生院; b. 航空工程学院, 西安 710038)

**摘 要:** 基于代价敏感学习的极限学习机(ELM)算法在处理不平衡数据分类问题时,未考虑不同类别样本的分布特点以及同一类别中各样本的重要性对分类结果的影响。为此,提出基于样本数量比例的错分惩罚因子设置方法,并基于 Mini-batch k-means 聚类与距离测度设计一种类内样本权值确定方案。在此基础上,构建区分正、负类别的隐含层输出矩阵,根据训练样本数与 ELM 隐含层节点数间的关系,分 2 种情况计算 ELM 隐含层与输出层间的连接权值,以降低算法的时间复杂度。实验结果表明,与 ELM、WELM 等算法相比,该算法的 *G-mean*、*F1* 分类性能指标值均较高。

**关键词:** 不平衡数据;极限学习机;代价敏感学习;Mini-batch k-means 聚类;约束优化理论

开放科学(资源服务)标志码(OSID):



**中文引用格式:**王大飞,解武杰,董文瀚. 基于 CSD-ELM 的不平衡数据分类算法[J]. 计算机工程,2019,45(11): 54-61.

**英文引用格式:**WANG Dafei,XIE Wujie,DONG Wenhan. Imbalanced data classification algorithm based on CSD-ELM[J]. Computer Engineering,2019,45(11):54-61.

## Imbalanced Data Classification Algorithm Based on CSD-ELM

WANG Dafei<sup>a</sup>, XIE Wujie<sup>b</sup>, DONG Wenhan<sup>b</sup>

(a. Graduate School; b. College of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China)

**[Abstract]** The Extreme Learning Machine (ELM) based on cost-sensitive learning has its advantages in dealing with imbalanced data classification problems. However, it fails to consider the distribution characteristics of samples in different classes and the importance of each sample in the same class, both of which can have influence on the classification results. Therefore, we propose a setting method for misclassified penalty factor based on the proportion of sample size. Besides, based on Mini-batch k-means clustering and distance measure, we propose a determination method for the weights of samples in the same class. On this basis, we build the output matrix of the hidden layer to distinguish the positive and negative categories. According to the relationship between the size of training samples and the number of nodes in the ELM hidden layer, we calculate the connection weights between the hidden layer and the output layer of ELM in two conditions, thus reducing the time complexity of the algorithm. Experimental results show that compared with ELM, WELM and other algorithms, the proposed algorithm has higher *G-mean* and *F1* classification performance index.

**[Key words]** imbalanced data; Extreme Learning Machine (ELM); cost-sensitive learning; Mini-batch k-means clustering; constrained optimization theory

DOI:10.19678/j.issn.1000-3428.0054988

### 0 概述

极限学习机(Extreme Learning Machine, ELM)是一种单隐含层前馈型神经网络<sup>[1]</sup>。ELM 可随机指定输入层与隐含层间的连接权值以及隐含层各节点的偏置值,并采用解析方法求解隐含层与输出层

间的连接权值。与传统基于梯度下降方法的神经网络相比,ELM 具有更好的泛化能力与更高的计算效率。文献[2]将约束优化理论引入到 ELM 的求解过程中,进一步完善了 ELM 的理论体系。

不平衡数据集是指其中某一类别样本数量远少于其他类别样本数量的数据集。传统 ELM 在设计

基金项目:航空科学基金(20141396012)。

作者简介:王大飞(1984—),男,硕士研究生,主研方向为数据挖掘、机器学习;解武杰、董文瀚,教授、博士。

收稿日期:2019-05-22 修回日期:2019-06-25 E-mail:746970090@qq.com

过程中并未考虑训练数据集的不平衡性给分类器带来的影响<sup>[3-5]</sup>。因此,文献[6-8]借鉴代价敏感学习的思想,从样本加权和惩罚因子调节 2 个方面对 ELM 算法进行改进,以改善 ELM 在不平衡数据集中对少数类样本的分类性能。

样本加权方法在模型训练过程中对样本施以不同的加权值,从而改善少数类样本的分类效果。文献[9]提出一种加权极限学习机(Weighted Extreme Learning Machine, WELM),其在求解 ELM 等价约束优化问题时,通过为不同类别样本赋予与其所属类别数量成反比的权重,来提高少数类样本被错分的代价,从而降低该类样本的错分概率。文献[10]提出一种面向不平衡数据流的自适应加权在线 ELM 算法,其通过设计自动调整实时到达训练样本加权值的惩罚矩阵,使得算法可以在线处理不平衡数据流的分类问题。文献[11]将模糊理论与加权 ELM 相结合,构造基于多数投票机制的集成分类器,以提高 WELM 对不平衡基因表达数据的分类性能。文献[12]提出一种基于指数加权的在线核序列 ELM 算法,并将其应用于混沌动力学系统的动态重构研究。WELM 在设计时虽然对每个样本都做了加权处理,但在计算加权值时仅考虑类别的影响,对同一类别样本都取同样的权值,并未考虑每个类别内部样本不同的重要性。

惩罚因子调节方法通过对不同类别样本赋予不同的错分惩罚因子,从而提高少数类样本的分类效果。文献[13]提出一种类依赖代价调节极限学习机(CCR-ELM),其在求解 ELM 等价约束优化问题时,通过对少数类样本施以较大错分惩罚因子和对多数类样本施以较小错分惩罚因子,以提高少数类样本的错分代价,最终降低该类样本的错分概率。文献[14]采用可变长度的粒子群优化算法自适应调整 CCR-ELM 的超参数,提高了 CCR-ELM 算法的稳定性及其对不平衡数据的分类精度。文献[15]提出了一种基于可变长度头脑风暴优化的自适应 CCR-ELM 算法,其提高了 CCR-ELM 算法在处理不平衡数据时的分类稳定性。CCR-ELM 在类别层面做了错分区别惩罚,但在惩罚因子的取值确定方面,对 2 种类别错分惩罚因子采取独立赋值的方法,并未考虑各类别样本的分布情况对错分惩罚因子取值的影响,且同样未考虑每个类别内部样本不同的重要性。

本文提出一种针对二分类不平衡数据集的类别与样本双依赖极限学习机(CSD-ELM)分类算法。按照两类别样本数量比例设计错分惩罚因子,根据类内样本分布情况给出基于 Mini-batch k-means 聚类的类内样本权值确定方法,并进一步按照聚类所得簇内样本分布情况提出基于类簇中心点距离测度

的簇内样本权值确定方法,每个样本的最终加权值由以上 2 个权值相乘而得。在此基础上,基于约束优化理论,提出区分正、负类别的隐含层输出矩阵,并根据训练样本数与 ELM 隐含层节点数间的关系,在较低的时间复杂度下分 2 种情况计算 ELM 隐含层与输出层间的连接权值。

## 1 算法描述

### 1.1 基于类别样本数量比例的错分惩罚因子设置

设计不平衡数据分类算法的主要目的是提高少数类样本的分类精度,同时尽可能保持多数类样本的分类精度。由于在 ELM 等价约束优化问题的目标函数中,惩罚因子越大,对该类样本分类错误的惩罚就越大,考虑到不同类别样本的分布情况,在 CCR-ELM 算法的基础上,本文提出一种按样本比例对两类别错分惩罚因子进行设置的方法,具体如下:

$$\begin{cases} C_+ = \frac{N}{N_+}C \\ C_- = \frac{N}{N_-}C \end{cases} \quad (1)$$

其中, $N_+$ 、 $N_-$ 分别表示训练集的少数类、多数类样本数量, $N$ 为训练集样本总量, $C_+$ 、 $C_-$ 分别表示少数类、多数类错分惩罚因子, $C$ 为一致惩罚因子。与 CCR-ELM 算法相比,本文算法的惩罚因子数量从 2 个减少为 1 个,这在很大程度上降低了算法在进行参数寻优时的计算复杂度。同时,有:

$$\frac{C_+}{C_-} = \frac{N}{N_+} = IR \quad (2)$$

其中, $IR$ 为训练数据集的类别不平衡比率。在设置一致惩罚因子  $C$  的基础上进行加权, $C_+ > C_-$ ,且  $C_+$  为  $C_-$  的  $IR$  倍,这就使得对少数类样本的错分惩罚大于对多数类样本的错分惩罚,且惩罚放大倍数与训练数据集的不平衡比率直接相关。

### 1.2 类内样本权值确定

Mini-batch k-means 算法是一种改进的 k-means 算法<sup>[16]</sup>,其通过随机抽样产生多个固定规模的小批量数据子集,并使用数据子集进行类簇划分和中心点更新计算。同时,Mini-batch k-means 算法还引入基于当前簇样本数倒数的学习率以提高算法的收敛速度。实验结果表明,上述改进大幅提高了 k-means 算法的运算速度,使算法能够在大数据集上以非常低的时间消耗来产生较好的聚类效果<sup>[16]</sup>。

WELM 算法在样本加权值的设置上仅考虑类别因素,对同一类别内部样本施以同样的权值。CCR-ELM 算法在设置错分惩罚因子时仅对 2 个类别进行区分设置,同样也未考虑每个类别内部样本不同的重要性。在通常情况下,由于每个样本在训

训练特征空间中所处位置的不同,它们对于分类算法训练过程中分类面形成的影响程度往往也不同,因此,应根据类内样本分布的特点对每个样本进行权值设置。本文设计基于 Mini-batch k-means 聚类与距离测度的类内样本权值确定方法。首先使用 Mini-batch k-means 算法对每个类别实例进行聚类,根据聚类所得每个类簇实例数量确定样本的簇加权值,类簇实例数越多,簇加权值越大。然后根据每个类簇内各实例点与簇中心点的欧式距离计算各样本的簇内加权值,距离越近,簇内加权值越大。每个样本的最终加权值由其簇加权值与簇内加权值相乘而得。基于 Mini-batch k-means 聚类与距离测度的类内样本权值确定方法具体描述如下:

**算法 1** 基于 Mini-batch k-means 聚类与距离测度的类内样本权值确定方法

**输入** 训练样本集  $D = \{(\mathbf{x}_i, t_i)\}_{i=1}^N, \mathbf{x}_i \in R^n, t_i \in \{+1, -1\}$ , 少数类实例集  $X_+ = \{\mathbf{x}_i^+ | \mathbf{x}_i^+ \in R^n, t_i^+ = +1, i=1, 2, \dots, N_+\}$ , 多数类实例集  $X_- = \{\mathbf{x}_i^- | \mathbf{x}_i^- \in R^n, t_i^- = -1, i=1, 2, \dots, N_-\}$

**输出** 少数类、多数类样本加权矩阵  $W_+, W_-$

**步骤 1** 使用 Mini-batch k-means 算法对  $X_+, X_-$  进行聚类, 分别获得  $k_+$  个类簇  $\{X_+^1, X_+^2, \dots, X_+^{k_+}\}$  及  $k_-$  个类簇  $\{X_-^1, X_-^2, \dots, X_-^{k_-}\}$ , 对应的类簇中心点分别为  $\{\mu_+^1, \mu_+^2, \dots, \mu_+^{k_+}\}$  及  $\{\mu_-^1, \mu_-^2, \dots, \mu_-^{k_-}\}$ , 簇实例数量分别为  $\{N_+^1, N_+^2, \dots, N_+^{k_+}\}$  及  $\{N_-^1, N_-^2, \dots, N_-^{k_-}\}$ 。

**步骤 2** 根据实例  $\mathbf{x}_i^+$  所属类簇  $X_+^p$  ( $p=1, 2, \dots, k_+$ ) 的实例数量占类别总实例数量的比例, 计算样本  $(\mathbf{x}_i^+, t_i^+)$  的簇加权值:  $(W_{ii}^+)^C = \frac{N_+^p}{N_+}$ 。

**步骤 3** 根据实例  $\mathbf{x}_i^+ \in X_+^p$  与所属类簇中心点的欧式距离计算样本  $(\mathbf{x}_i^+, t_i^+)$  的簇内加权值:  $(W_{ii}^+)^S = \exp(-\|\mathbf{x}_i^+ - \mu_+^p\|)$ 。

**步骤 4** 计算样本  $\mathbf{x}_i^+ \in X_+^p$  的最终加权值:  $W_{ii}^+ = (W_{ii}^+)^C \cdot (W_{ii}^+)^S$ 。

**步骤 5** 令  $i = i + 1$ , 重复步骤 2 ~ 步骤 4, 直至  $i = N_+$ 。

**步骤 6** 返回少数类样本加权矩阵:  $W_+ = \text{diag}(W_{ii}^+)$ 。

**步骤 7** 按步骤 2 ~ 步骤 6 计算并返回多数类样本加权矩阵:  $W_- = \text{diag}(W_{ii}^-)$ 。

### 1.3 CSD-ELM 分类算法

本文基于约束优化理论, 通过设计区分正、负类别的隐含层输出矩阵, 并根据训练样本数与 ELM 隐含层节点数间的关系, 分 2 种情况计算 ELM 隐含层与输出层间的连接权值。具体过程如下:

给定一个类别不平衡的二分类问题训练数据集  $D = \{(\mathbf{x}_i, t_i)\}_{i=1}^N, \mathbf{x}_i \in R^n, t_i \in \{+1, -1\}$ , CSD-ELM

算法的等价约束优化问题可以表示为:

$$\text{Minimize: } \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C_+ \frac{1}{2} \sum_{i=1}^{N_+} W_{ii}^+ (\xi_i^+)^2 + C_- \frac{1}{2} \sum_{i=1}^{N_-} W_{ii}^- (\xi_i^-)^2$$

Subject to:

$$\mathbf{h}(\mathbf{x}_i^+) \boldsymbol{\beta} = t_i^+ - \xi_i^+, i=1, 2, \dots, N_+$$

$$\mathbf{h}(\mathbf{x}_i^-) \boldsymbol{\beta} = t_i^- - \xi_i^-, i=1, 2, \dots, N_- \quad (3)$$

其中,  $(\mathbf{x}_i^+, t_i^+)$ 、 $(\mathbf{x}_i^-, t_i^-)$  分别表示少数类、多数类样本, 其数量分别为  $N_+, N_-$ ,  $\xi^+, \xi^-$  分别表示少数类、多数类样本的训练误差,  $\boldsymbol{\beta}$  表示 ELM 隐含层与输出层间的连接权值,  $\mathbf{h}(\mathbf{x})$  表示 ELM 的隐含层输出函数,  $h_i(\mathbf{x})$  表示 ELM 第  $i$  个隐含层节点的输出函数,  $\mathbf{h}(\mathbf{x}_i^+) = [h_1(\mathbf{x}_i^+), h_2(\mathbf{x}_i^+), \dots, h_L(\mathbf{x}_i^+)]^T$  个隐含层节点的输出函数,  $\mathbf{h}(\mathbf{x}_i^+) = [h_1(\mathbf{x}_i^+), h_2(\mathbf{x}_i^+), \dots, h_L(\mathbf{x}_i^+)]^T$ ,  $\mathbf{h}(\mathbf{x}_i^-) = [h_1(\mathbf{x}_i^-), h_2(\mathbf{x}_i^-), \dots, h_L(\mathbf{x}_i^-)]^T$ 。少数类和多数类样本错分惩罚因子  $C_+$  和  $C_-$  的取值按式(1)设置, 少数类、多数类样本加权因子  $W_{ii}^+, W_{ii}^-$  按算法 1 计算而得。

分别记:

$$\mathbf{t}_+ = [t_1^+, t_2^+, \dots, t_{N_+}^+]^T \quad (4)$$

$$\mathbf{t}_- = [t_1^-, t_2^-, \dots, t_{N_-}^-]^T \quad (5)$$

$$\boldsymbol{\xi}_+ = [\xi_1^+, \xi_2^+, \dots, \xi_{N_+}^+]^T \quad (6)$$

$$\boldsymbol{\xi}_- = [\xi_1^-, \xi_2^-, \dots, \xi_{N_-}^-]^T \quad (7)$$

$$\mathbf{H}_+ = [\mathbf{h}(\mathbf{x}_1^+)^T, \mathbf{h}(\mathbf{x}_2^+)^T, \dots, \mathbf{h}(\mathbf{x}_{N_+}^+)^T]^T \quad (8)$$

$$\mathbf{H}_- = [\mathbf{h}(\mathbf{x}_1^-)^T, \mathbf{h}(\mathbf{x}_2^-)^T, \dots, \mathbf{h}(\mathbf{x}_{N_-}^-)^T]^T \quad (9)$$

将式(3)写为矩阵形式:

$$\text{Minimize: } \frac{1}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} + C_+ \frac{1}{2} \boldsymbol{\xi}_+^T \mathbf{W}_+ \boldsymbol{\xi}_+ + C_- \frac{1}{2} \boldsymbol{\xi}_-^T \mathbf{W}_- \boldsymbol{\xi}_-$$

Subject to:

$$\mathbf{H}_+ \boldsymbol{\beta} = \mathbf{t}_+ - \boldsymbol{\xi}_+, \mathbf{H}_- \boldsymbol{\beta} = \mathbf{t}_- - \boldsymbol{\xi}_- \quad (10)$$

其中,  $\mathbf{W}_+$  与  $\mathbf{W}_-$  根据算法 1 计算而得。

构造式(10)所述约束优化问题的拉格朗日辅助函数:

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\xi}_+, \boldsymbol{\xi}_-, \boldsymbol{\alpha}_+, \boldsymbol{\alpha}_-) &= \frac{1}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} + C_+ \frac{1}{2} \boldsymbol{\xi}_+^T \mathbf{W}_+ \boldsymbol{\xi}_+ + \\ &C_- \frac{1}{2} \boldsymbol{\xi}_-^T \mathbf{W}_- \boldsymbol{\xi}_- - \boldsymbol{\alpha}_+^T (\mathbf{H}_+ \boldsymbol{\beta} - \mathbf{t}_+ + \boldsymbol{\xi}_+) - \\ &\boldsymbol{\alpha}_-^T (\mathbf{H}_- \boldsymbol{\beta} - \mathbf{t}_- + \boldsymbol{\xi}_-) \end{aligned} \quad (11)$$

其中,  $\boldsymbol{\alpha}_+, \boldsymbol{\alpha}_-$  为拉格朗日乘子,  $\boldsymbol{\alpha}_+ = [\alpha_+^1, \alpha_+^2, \dots, \alpha_+^{N_+}]^T \in R^{N_+}$ ,  $\boldsymbol{\alpha}_- = [\alpha_-^1, \alpha_-^2, \dots, \alpha_-^{N_-}]^T \in R^{N_-}$ 。

求解式(11)的 KKT 条件<sup>[17]</sup>为:

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \rightarrow \boldsymbol{\beta} = \mathbf{H}_+^T \boldsymbol{\alpha}_+ + \mathbf{H}_-^T \boldsymbol{\alpha}_- \quad (12)$$

$$\frac{\partial L}{\partial \boldsymbol{\xi}_+} = 0 \rightarrow \boldsymbol{\alpha}_+ = C_+ \mathbf{W}_+ \boldsymbol{\xi}_+ \quad (13)$$

$$\frac{\partial L}{\partial \xi_-} = 0 \rightarrow \alpha_- = C_- W_- \xi_- \quad (14)$$

$$\frac{\partial L}{\partial \alpha_+} = 0 \rightarrow H_+ \beta = t_+ - \xi_+ \quad (15)$$

$$\frac{\partial L}{\partial \alpha_-} = 0 \rightarrow H_- \beta = t_- - \xi_- \quad (16)$$

将式(12)~式(16)写为分块矩阵形式:

$$\beta = \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} = \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} \xi_+ \\ \xi_- \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} H_+ \\ H_- \end{bmatrix} \beta = \begin{bmatrix} t_+ \\ t_- \end{bmatrix} - \begin{bmatrix} \xi_+ \\ \xi_- \end{bmatrix} \quad (19)$$

分以下2种情况进行讨论:

1) 当训练样本数少于或等于隐含层节点数,即  $N \leq L$  时,由式(18)可得:

$$\begin{bmatrix} \xi_+ \\ \xi_- \end{bmatrix} = \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix}^{-1} \begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} \quad (20)$$

将式(17)、式(20)代入式(19)中,可得:

$$\begin{bmatrix} H_+ \\ H_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} = \begin{bmatrix} t_+ \\ t_- \end{bmatrix} - \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix}^{-1} \begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} \quad (21)$$

将式(21)两端左乘  $\begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix}$ , 可得:

$$\begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} = \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix} - \begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} \quad (22)$$

从而,有:

$$\begin{bmatrix} \alpha_+ \\ \alpha_- \end{bmatrix} = \left( \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T + I \right)^{-1} \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix} \quad (23)$$

将式(23)代入式(17)中,可得:

$$\beta = \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \left( \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T + I \right)^{-1} \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix} \quad (24)$$

2) 当训练样本数多于隐含层节点数,即  $N > L$  时,由式(17)、式(18)可得:

$$\beta = \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} \xi_+ \\ \xi_- \end{bmatrix} \quad (25)$$

于是,有:

$$\begin{bmatrix} \xi_+ \\ \xi_- \end{bmatrix} = \left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \right)^\dagger \beta \quad (26)$$

其中,  $(\cdot)^\dagger$  表示矩阵的 Moore-Penrose 广义逆<sup>[18]</sup>。

将式(26)代入式(19)中,整理可得:

$$\left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \right)^\dagger \begin{bmatrix} H_+ \\ H_- \end{bmatrix} \beta = \begin{bmatrix} t_+ \\ t_- \end{bmatrix} \quad (27)$$

由于  $\left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \right)_{L \times N}$  矩阵满行秩,根据 Moore-Penrose 广义逆矩阵的性质<sup>[18]</sup>,可得:

$$\left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \right)^\dagger = I_{L \times L} \quad (28)$$

然后将式(27)的两端同时左乘表达式

$$\left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \right), \text{可得:}$$

$$\left( \left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \right) \begin{bmatrix} H_+ \\ H_- \end{bmatrix} + I \right) \beta = \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix} \quad (29)$$

从而,有:

$$\beta = \left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix} + I \right)^{-1} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix} \quad (30)$$

综合以上2种情况,可以得到式(31)。根据式(31)进一步得到二分类CSD-ELM的输出函数为式(32)。

$$\beta = \begin{cases} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \cdot \left( \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T + I \right)^{-1} \cdot \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix}, N \leq L \\ \left( \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} H_+ \\ H_- \end{bmatrix} + I \right)^{-1} \cdot \begin{bmatrix} H_+ \\ H_- \end{bmatrix}^T \begin{bmatrix} C_+ W_+ & O \\ O & C_- W_- \end{bmatrix} \begin{bmatrix} t_+ \\ t_- \end{bmatrix}, N > L \end{cases} \quad (31)$$

$$f(\mathbf{x}) = \text{sign } \mathbf{h}(\mathbf{x})\boldsymbol{\beta} = \begin{cases} \text{sign } \mathbf{h}(\mathbf{x}) \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}^T \left( \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix} \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix} \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}^T + \mathbf{I} \right)^{-1} \cdot \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix} \begin{bmatrix} \mathbf{t}_+ \\ \mathbf{t}_- \end{bmatrix}, N \leq L \\ \text{sign } \mathbf{h}(\mathbf{x}) \left( \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix} \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix} + \mathbf{I} \right)^{-1} \cdot \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix} \begin{bmatrix} \mathbf{t}_+ \\ \mathbf{t}_- \end{bmatrix}, N > L \end{cases} \quad (32)$$

CSD-ELM 算法描述如下:

#### 算法 2 CSD-ELM 算法

输入 训练样本集  $D = \{(\mathbf{x}_i, t_i)\}_{i=1}^N$ , 其中,  $\mathbf{x}_i \in R^n$ ,  $t_i \in \{+1, -1\}$ , 少数类、多数类样本集  $D_+ = \{(\mathbf{x}_i^+, t_i^+)\}_{i=1}^{N_+}$ ,  $D_- = \{(\mathbf{x}_i^-, t_i^-)\}_{i=1}^{N_-}$ , 其中,  $\mathbf{x}_i^+$ ,  $\mathbf{x}_i^- \in R^n$ ,  $t_i^+ = +1$ ,  $t_i^- = -1$ , ELM 的隐含层节点数  $L$ , 隐含层激活函数  $G(\mathbf{x})$

输出 二分类 CSD-ELM 的输出函数  $f(\mathbf{x})$

步骤 1 根据任意随机概率分布函数产生 ELM 输入层与隐含层间的连接权值  $\{\mathbf{a}_j\}_{j=1}^L$  及隐含层偏置值  $\{b_j\}_{j=1}^L$ , 从而得到隐含层输出函数  $\mathbf{h}(\mathbf{x}) = [G(\mathbf{a}_1, b_1, \mathbf{x}), G(\mathbf{a}_2, b_2, \mathbf{x}), \dots, G(\mathbf{a}_L, b_L, \mathbf{x})]$ 。

步骤 2 根据式(4)、式(5)、式(8)及式(9)分别计算  $\mathbf{t}_+$ 、 $\mathbf{t}_-$ 、 $\mathbf{H}_+$  及  $\mathbf{H}_-$ 。

步骤 3 根据式(1)计算少数类错分惩罚因子  $C^+$  及多数类错分惩罚因子  $C^-$ 。

步骤 4 根据算法 1 计算少数类加权矩阵  $\mathbf{W}^+$  及多数类加权矩阵  $\mathbf{W}^-$ 。

步骤 5 根据式(31)计算 ELM 隐含层与输出层间的连接权值  $\boldsymbol{\beta}$ 。

步骤 6 根据式(32)返回二分类 CSD-ELM 的输出函数  $f(\mathbf{x})$ 。

#### 1.4 算法时间复杂度分析

由式(24)可得  $N \leq L$  时  $\boldsymbol{\beta}$  的计算公式为:

$$\boldsymbol{\beta} = \left( \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix} \right)_{L \times N}^T \left( \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix}_{N \times N} \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}_{N \times L} \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}_{N \times L}^T + \mathbf{I}_{N \times N} \right)^{-1} \cdot \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix}_{N \times N} \begin{bmatrix} \mathbf{t}_+ \\ \mathbf{t}_- \end{bmatrix}_{N \times 1} \quad (33)$$

可得此时计算  $\boldsymbol{\beta}$  的时间复杂度为:

$$O_{N \leq L} = O(N^3 + LN^2 + LN^2 + LN + LN^2 + LN^2) = O(N^3 + 4LN^2 + LN) \quad (34)$$

由式(30)可得  $N > L$  时  $\boldsymbol{\beta}$  的计算公式为:

$$\boldsymbol{\beta} = \left( \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix} \right)_{L \times N}^T \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix}_{N \times N} \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}_{N \times L} + \mathbf{I}_{L \times L} \right)^{-1} \cdot \begin{bmatrix} \mathbf{H}_+ \\ \mathbf{H}_- \end{bmatrix}_{N \times L}^T \begin{bmatrix} \mathbf{C}_+ \mathbf{W}_+ & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_- \mathbf{W}_- \end{bmatrix}_{N \times N} \begin{bmatrix} \mathbf{t}_+ \\ \mathbf{t}_- \end{bmatrix}_{N \times 1} \quad (35)$$

可得此时计算  $\boldsymbol{\beta}$  的时间复杂度为:

$$O_{N > L} = O(L^3 + L^2N + LN^2 + LN + LN^2 + L^2N) = O(L^3 + 2LN^2 + 2L^2N + LN) \quad (36)$$

由于:

$$(N^3 + 4LN^2 + LN) - (L^3 + 2LN^2 + 2L^2N + LN) = N^3 - L^3 + 2LN^2 - 2L^2N = (N-L)(N^2 + L^2 + 3NL) \begin{cases} \leq 0, N \leq L \\ > 0, N > L \end{cases} \quad (37)$$

故有:

$$\begin{cases} O_{N \leq L} \leq O_{N > L}, N \leq L \\ O_{N > L} < O_{N \leq L}, N > L \end{cases} \quad (38)$$

由式(38)可知, CSD-ELM 算法总能在较低的时间复杂度下求解  $\boldsymbol{\beta}$ 。

## 2 实验结果与分析

### 2.1 评价指标

二分类问题的分类性能评价通常建立在表 1 所示的分类结果混淆矩阵基础上<sup>[19]</sup>, 其中, 将少数类规定为正类, 多数类规定为负类。

表 1 分类结果混淆矩阵

分类情况	预测为正类	预测为负类
真实为正类	TP(真正例数)	FN(假负例数)
真实为负类	FP(假正例数)	TN(真负例数)

在表 1 中,  $TP$  和  $FP$  分别表示被分类器预测为正类、真实类别为正类和负类的样本数量,  $TN$  和  $FN$  分别表示被分类器预测为负类、真实类别为负类和正类的样本数量。由上述 4 个量可以定义如下常用的分类性能评价指标:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (39)$$

$$Precision = \frac{TP}{TP + FP} \quad (40)$$

$$Recall = \frac{TP}{TP + FN} \quad (41)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (42)$$

$$G-mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \quad (43)$$

其中,  $ACC$  表示被预测类别正确的样本占总样本的比例,其被称为整体分类精度(简称为分类精度),  $Precision$  称为查准率,表示所有被预测为正类的样本中真实类别为正类的样本所占的比例,  $Recall$  称为查全率,表示所有真实类别为正类的样本中被正确预测为正类的样本所占的比例,  $F1$  为查准率和查全率的调和平均,当查准率和查全率大小相当时  $F1$  较大,  $G-mean$  为正类分类正确率和负类分类正确率的几何平均,  $G-mean$  较高,说明分类器对正类和负类的分类性能都较好。

针对不平衡数据分类的特点,为更全面地评价分类器对多数类与少数类样本的综合分类性能,本文选取  $G-mean$  作为分类器性能的主要评价指标,并将  $F1$ 、 $ACC$  作为分类器性能评价的辅助指标。

## 2.2 实验设置

本文实验的硬件环境为 Intel i5-6200U 4 核 CPU,主频 2.30 GHz,内存 8 GB,操作系统为 Win10 64 位,编译环境为 Python 3.6。

本文从 KEEL 数据库<sup>[20]</sup>中选取 8 个二分类不平衡数据集进行算法对比与性能测试。实验数据集在选取时尽可能地考虑样本数量与不平衡比率的多样性,数据集具体信息如表 2 所示。为综合评价 CSD-ELM 算法的分类性能与计算效率,本文将其与 ELM、WELM、CCR-ELM 等多种算法进行对比。在算法涉及的参数设置上,参照文献[2,10]的方法设置统一的取值范围,并通过网格搜索方法针对具体的分类性能指标寻求最优参数组合,各算法参数设置范围如表 3 所示,其中,隐含层节点激活函数统一使用 Sigmoid 函数。同时,为体现实验的公正性,本文采用 10 折交叉检验的方法,取 10 次结果的平均值及标准差作为最终实验结果。

表 2 实验数据集具体信息

序号	数据集	样本数	特征数	少数类样本数	多数类样本数	类不平衡比率
1	vehicle1	846	18	217	629	2.90
2	segment0	2 308	19	329	1 979	6.02
3	pageblocks0	5 472	10	559	4 913	8.79
4	yeast2vs4	514	8	51	463	9.08
5	shuttleC0vsC4	1 829	9	123	1 706	13.87
6	pageblocks13vs4	472	10	28	444	15.86
7	glass5	214	9	9	205	22.78
8	yeast6	1 484	8	35	1 449	41.40

表 3 各算法参数设置

算法	参数	取值范围
ELM	惩罚因子 $C$	$\{2^{-10}, 2^{-9}, \dots, 2^{50}\}$
	隐含层节点数 $L$	$\{10, 20, \dots, 1\ 000\}$
WELM	惩罚因子 $C$	$\{2^{-10}, 2^{-9}, \dots, 2^{50}\}$
	隐含层节点数 $L$	$\{10, 20, \dots, 1\ 000\}$
CCR-ELM	少数类惩罚因子 $C_+$	$\{2^{-10}, 2^{-9}, \dots, 2^{50}\}$
	多数类惩罚因子 $C_-$	$\{2^{-10}, 2^{-9}, \dots, 2^{50}\}$
	隐含层节点数 $L$	$\{10, 20, \dots, 1\ 000\}$
CSD-ELM	惩罚因子 $C$	$\{2^{-10}, 2^{-9}, \dots, 2^{50}\}$
	隐含层节点数 $L$	$\{10, 20, \dots, 1\ 000\}$

## 2.3 结果分析

从表 4 可以看出,与原始 ELM 算法相比, WELM、CCR-ELM、CSD-ELM 算法在  $G-mean$ 、 $F1$

两项分类性能指标上多数都有所提升,但  $ACC$  指标提升幅度普遍较低,甚至在某些数据集上还有所下降。造成该现象的原因在于:在未采取任何不平衡数据处理技术的情况下,因为样本数量的悬殊差距,传统的 ELM 算法必然会产生倾向于多数类样本的分类结果,最终使其总体分类精度  $ACC$  较高;而改进 ELM 算法由于使用了样本加权或者类别区分惩罚技术,因此其少数类样本的分类精度得到了有效提升,这些算法的  $G-mean$  及  $F1$  值也较高。在 WELM、CCR-ELM、CSD-ELM 3 种算法的对比中,WELM 及 CSD-ELM 都表现出了较强的稳定性,而 CCR-ELM 算法则在一些数据集(尤其是不平衡度较高的 glass5、yeast6 数据集)上出现了一定程度的性能波动。本文 CSD-ELM 算法不仅同时

使用了样本加权和类别区分惩罚技术,并且针对不同数据集的类别分布特点设计了与其不平衡比率相关联的正负类别惩罚因子,使用 Mini-batch k-means 聚类算法并基于聚类后各簇中心点的距离

测度对同一类别内部样本进行区分加权,上述设计都有助于其更精准地进行分类。从实验结果也可以看出,CSD-ELM 算法在多数数据集上都取得了最高的  $G\text{-mean}$  及  $F1$  值。

表 4 各算法分类性能对比结果

数据集	算法	评价指标			参数设置	
		$G\text{-mean}$	$F1$	$ACC$	$L$	$C(C_+, C_-)$
glass5	ELM	0.926 9 ± 0.054 5	0.470 0 ± 0.021 0	0.968 2 ± 0.035 9	200	2 <sup>45</sup>
	WELM	0.963 0 ± 0.020 1	0.656 7 ± 0.021 9	0.931 8 ± 0.037 4	650	2 <sup>35</sup>
	CCR-ELM	0.915 6 ± 0.035 3	0.455 2 ± 0.027 2	0.900 9 ± 0.064 5	220	2 <sup>34</sup> , 2 <sup>5</sup>
	CSD-ELM	0.983 0 ± 0.042 1	0.785 4 ± 0.037 1	0.967 3 ± 0.022 5	580	2 <sup>23</sup>
pageblocks 13 vs 4	ELM	0.935 3 ± 0.046 6	0.824 3 ± 0.025 2	0.975 0 ± 0.019 1	390	2 <sup>15</sup>
	WELM	0.963 1 ± 0.051 2	0.876 1 ± 0.069 0	0.962 5 ± 0.014 1	220	2 <sup>6</sup>
	CCR-ELM	0.951 5 ± 0.079 3	0.894 6 ± 0.015 6	0.972 9 ± 0.026 1	790	2 <sup>26</sup> , 2 <sup>7</sup>
	CSD-ELM	0.994 4 ± 0.003 5	0.932 1 ± 0.045 2	0.969 6 ± 0.006 6	160	2 <sup>11</sup>
yeast2 vs 4	ELM	0.859 7 ± 0.056 1	0.829 4 ± 0.010 7	0.966 0 ± 0.021 4	50	2 <sup>14</sup>
	WELM	0.913 9 ± 0.054 8	0.805 4 ± 0.094 2	0.969 8 ± 0.022 1	140	2 <sup>21</sup>
	CCR-ELM	0.875 6 ± 0.059 6	0.828 7 ± 0.014 6	0.962 3 ± 0.034 4	940	2 <sup>18</sup> , 2 <sup>2</sup>
	CSD-ELM	0.939 6 ± 0.035 0	0.866 2 ± 0.056 5	0.947 2 ± 0.019 5	100	2 <sup>24</sup>
vehicle1	ELM	0.763 9 ± 0.046 8	0.676 6 ± 0.055 4	0.841 2 ± 0.027 9	660	2 <sup>5</sup>
	WELM	0.813 2 ± 0.056 9	0.725 0 ± 0.073 1	0.854 1 ± 0.038 5	870	2 <sup>11</sup>
	CCR-ELM	0.789 6 ± 0.062 9	0.696 1 ± 0.026 7	0.844 1 ± 0.036 1	230	2 <sup>15</sup> , 2 <sup>4</sup>
	CSD-ELM	0.858 5 ± 0.047 3	0.750 7 ± 0.070 9	0.843 5 ± 0.050 2	730	2 <sup>13</sup>
yeast6	ELM	0.718 7 ± 0.015 7	0.476 8 ± 0.010 0	0.935 6 ± 0.018 2	740	2 <sup>17</sup>
	WELM	0.754 4 ± 0.017 2	0.537 1 ± 0.027 4	0.948 3 ± 0.005 5	840	2 <sup>34</sup>
	CCR-ELM	0.684 8 ± 0.051 8	0.459 1 ± 0.041 9	0.916 2 ± 0.011 1	910	2 <sup>41</sup> , 2 <sup>8</sup>
	CSD-ELM	0.866 4 ± 0.040 2	0.696 5 ± 0.052 0	0.905 1 ± 0.018 8	80	2 <sup>12</sup>
shuttleC0 vs C4	ELM	0.922 0 ± 0.064 0	0.907 2 ± 0.069 4	0.989 7 ± 0.008 3	80	2 <sup>-7</sup>
	WELM	0.976 3 ± 0.027 8	0.975 7 ± 0.028 7	0.996 7 ± 0.003 8	600	2 <sup>-10</sup>
	CCR-ELM	0.956 0 ± 0.040 5	0.954 3 ± 0.042 7	0.994 0 ± 0.005 4	160	2 <sup>5</sup> , 2 <sup>-8</sup>
	CSD-ELM	0.999 7 ± 0.000 9	0.996 3 ± 0.011 7	0.999 5 ± 0.001 7	80	2 <sup>29</sup>
segment0	ELM	0.936 9 ± 0.033 8	0.934 4 ± 0.036 5	0.982 7 ± 0.008 9	920	2 <sup>7</sup>
	WELM	0.992 4 ± 0.008 0	0.992 3 ± 0.008 1	0.997 8 ± 0.002 3	190	2 <sup>4</sup>
	CCR-ELM	0.972 6 ± 0.011 5	0.899 1 ± 0.027 7	0.968 4 ± 0.010 2	990	2 <sup>35</sup> , 2 <sup>6</sup>
	CSD-ELM	0.988 5 ± 0.001 8	0.981 1 ± 0.010 3	0.987 4 ± 0.003 0	810	2 <sup>22</sup>
pageblocks0	ELM	0.834 5 ± 0.036 4	0.782 0 ± 0.043 1	0.960 0 ± 0.007 2	140	2 <sup>14</sup>
	WELM	0.891 3 ± 0.022 5	0.778 0 ± 0.036 1	0.968 4 ± 0.004 8	540	2 <sup>34</sup>
	CCR-ELM	0.890 1 ± 0.027 4	0.829 5 ± 0.037 8	0.966 2 ± 0.007 2	690	2 <sup>42</sup> , 2 <sup>8</sup>
	CSD-ELM	0.922 3 ± 0.014 3	0.838 8 ± 0.026 3	0.944 9 ± 0.010 8	330	2 <sup>27</sup>

由表 5 可以看出,各算法的训练时间基本上都随训练数据集样本数量的增多而增加。在同一训练数据集中,ELM 算法的平均训练时间最短,其他 3 种算法的训练时间都要高于原始 ELM 算法。而在 WELM、CCR-ELM、CSD-ELM 3 种算法的对比

中,CCR-ELM 算法的训练时间普遍较低,CSD-ELM 算法的训练时间最高。造成该现象的原因在于,WELM 增加了加权矩阵的计算,而 CSD-ELM 则要额外基于聚类与距离测度来计算每个样本的类内加权值等。

表5 各算法训练时间对比

数据集	样本数	算法训练时间/s			
		ELM	WELM	CCR-ELM	CSD-ELM
glass5	214	0.095 1 ± 0.000 1	0.155 8 ± 0.000 2	0.105 7 ± 0.000 6	0.306 1 ± 0.001 2
pageblocks13vs4	472	0.145 8 ± 0.006 6	0.262 2 ± 0.007 5	0.167 7 ± 0.008 4	0.472 9 ± 0.005 6
yeast2vs4	514	0.134 9 ± 0.008 1	0.397 1 ± 0.006 5	0.141 5 ± 0.005 1	0.865 1 ± 0.000 4
vehicle1	846	0.184 1 ± 0.008 2	0.673 4 ± 0.007 4	0.205 9 ± 0.001 7	1.581 9 ± 0.001 7
yeast6	1 484	0.317 9 ± 0.000 7	1.286 9 ± 0.006 1	0.581 5 ± 0.001 9	4.518 1 ± 0.006 2
shuttleC0vsC4	1 829	0.437 6 ± 0.006 6	1.577 2 ± 0.015 2	0.581 3 ± 0.000 1	5.007 2 ± 0.001 8
segment0	2 308	0.617 8 ± 0.020 9	2.074 9 ± 0.038 6	0.771 9 ± 0.001 4	6.124 8 ± 0.010 4
pageblocks0	5 472	1.054 1 ± 0.001 7	10.517 8 ± 0.005 7	1.356 1 ± 0.007 8	32.619 7 ± 0.008 1

### 3 结束语

本文提出一种不平衡数据分类算法 CSD-ELM。分别设计错分惩罚因子与类内样本权值的确定方法,在此基础上,利用约束优化理论,根据训练样本数与 ELM 隐含层节点数间的关系,分 2 种情况求解 ELM 隐含层与输出层间的连接权值。实验结果表明,相比于 ELM、WELM、CCR-ELM 3 种算法,CSD-ELM 算法的 *G-mean*、*F1* 指标值较高,分类性能较好,但其平均训练时间高于 3 种对比算法。因此,下一步将通过增量式训练方法,在提高 CSD-ELM 算法分类性能的同时缩短其训练时间。

### 参考文献

- [1] HUANG Guangbin, ZHU Qinyu, SIEW C K. Extreme learning machine: theory and applications [J]. *Neurocomputing*, 2006, 70: 489-501.
- [2] HUANG Guangbin, ZHOU Hongming, DING Xiaojian, 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.
- [3] 于化龙. 不平衡学习: 理论与算法[M]. 北京: 清华大学出版社, 2017.
- [4] JANAKIRAMAN V M, NGUYEN X L, STERNIAK 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.
- [5] JANAKIRAMAN V M, NGUYEN X L, ASSANIS D. Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines[J]. *Neurocomputing*, 2016, 177: 304-316.
- [6] ELKAN C. The foundations of cost-sensitive learning [C]//*Proceedings of the 17th International Joint Conference on Artificial Intelligence*. Washington D. C., USA: IEEE Press, 2001: 973-978.
- [7] 赵永彬, 陈硕, 刘明, 等. 基于置信度代价敏感的支持向量机不均衡数据学习[J]. *计算机工程*, 2015, 41(10): 177-180.
- [8] 陈博深. 代价敏感的多分类恶意网页识别系统研究与实现[D]. 北京: 北京邮电大学, 2019.
- [9] ZONG Weiwei, HUANG Guangbin, CHEN Yiqiang. Weighted extreme learning machine for imbalance learning[J]. *Neurocomputing*, 2013, 101: 229-242.
- [10] 梅颖, 卢诚波. 面向不平衡数据流的自适应加权在线超限学习机算法[J]. *模式识别与人工智能*, 2019, 32(2): 144-150.
- [11] WANG Yang, WANG Anna, AI Qing, et al. Ensemble based fuzzy weighted extreme learning machine for gene expression classification[J]. *Applied Intelligence*, 2019, 49(3): 1161-1171.
- [12] 李军, 后新燕. 基于指数加权-核在线序列极限学习机的混沌系统动态重构研究[J]. *物理学报*, 2019, 68(10): 27-39.
- [13] XIAO Wendong, ZHANG Jie, LI Yanjiao, et al. Class-specific cost regulation extreme learning machine for imbalanced classification [J]. *Neurocomputing*, 2017, 261: 70-82.
- [14] GUO Yinan, ZHANG Pei, CUI Ning, et al. VPSO-based CCR-ELM for imbalanced classification [C]//*Proceedings of the 9th International Conference on Swarm Intelligence*. Berlin, Germany: Springer, 2018: 361-369.
- [15] CHENG Jian, CHEN Jingjing, GUO Yinan, et al. Adaptive CCR-ELM with variable-length brain storm optimization algorithm for class-imbalance learning[J]. *Natural Computing*, 2019(2): 1-12.
- [16] SCULLEY D. Web-scale K-means clustering [C]//*Proceedings of the 19th International Conference on World Wide Web*. New York, USA: ACM Press, 2010: 1177-1178.
- [17] FLETCHER R. Practical methods of optimization [M]. New York, USA: Wiley, 2013.
- [18] SERRE D. Matrices: theory and applications [M]. Berlin, Germany: Springer, 2002.
- [19] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.
- [20] ALCALÁ FDEZ J, FERNÁNDEZ A, LUENGO J, et al. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework[J]. *Journal of Multiple-Valued Logic and Soft Computing*, 2011, 17: 255-287.

编辑 吴云芳