面对类别不平衡的增量在线序列极限学习机

左鹏玉， 周洁， 王士同

引用本文:
左鹏玉, 周洁, 王士同. 面对类别不平衡的增量在线序列极限学习机[J]. 智能系统学报, 2020, 15(3): 520–527.
ZUO Pengyu, ZHOU Jie, WANG Shitong. Incremental online sequential extreme learning machine for imbalanced data[J]. *CAAI Transactions on Intelligent Systems*, 2020, 15(3): 520–527.

在线阅读 View online: https://dx.doi.org/10.11992/tis.201904040

DOI: 10.11992/tis.201904040

面对类别不平衡的增量在线序列极限学习机

左鹏玉[1], 周洁[1], 王士同[1,2]

(1. 江南大学 数字媒体学院, 江苏 无锡 214122; 2. 江苏省媒体设计与软件设计重点实验室, 江苏 无锡 214122)

摘 要: 针对在线序列极限学习机对于类别不平衡数据的学习效率低、分类准确率差的问题, 提出了面对类别不平衡的增量在线序列极限学习机 (IOS-ELM)。该算法根据类别不平衡比例调整平衡因子, 利用分块矩阵的广义逆矩阵对隐含层节点数进行寻优, 提高了模型对类别不平衡数据的在线处理能力, 最后通过 14 个二类和多类不平衡数据集对该算法有效性和可行性进行验证。实验结果表明: 该算法与同类其他算法相比具有更好的泛化性和准确率, 适用于类别不平衡场景下的在线学习。

关键词: 类别不平衡学习; 增量; 无逆矩阵; 在线学习; 极限学习机; 分类; 多类不平衡; 神经网络

中图分类号: TP181 **文献标志码:** A **文章编号:** 1673−4785(2020)03−0520−08

中文引用格式: 左鹏玉, 周洁, 王士同. 面对类别不平衡的增量在线序列极限学习机 [J]. 智能系统学报, 2020, 15(3): 520–527.

英文引用格式: ZUO Pengyu, ZHOU Jie, WANG Shitong. Incremental online sequential extreme learning machine for imbalanced data[J]. CAAI transactions on intelligent systems, 2020, 15(3): 520–527.

Incremental online sequential extreme learning machine for imbalanced data

ZUO Pengyu[1], ZHOU Jie[1], WANG Shitong[1,2]

(1. College of Digital Media, Jiangnan University, Wuxi 214122, China; 2. Jiangsu Province Key Lab. of Media Design & Software Technologies, Wuxi 214122, China)

Abstract: In this paper, an incremental online sequential extreme learning machine (IOS-ELM) is proposed to solve the problems of low efficiency and poor classification accuracy of OS-ELM for class imbalance learning. The basic idea is to adjust the balance factor according to the category imbalance ratio in an imbalanced dataset and then determine an optimal number of hidden nodes using the generalized inverse of the block matrix, thereby improving the online learning ability of IOS-ELM. The experiments on the effectiveness and feasibility of 14 binary-class and multi-class imbalanced datasets show that the proposed IOS-ELM has better generalization capability and classification performance than other comparative methods.

Keywords: class imbalance; incremental learning; inverse-free matrix; online learning; extreme learning machine; classification; multi-class imbalanced; neural network

近年来, 极限学习机 (extreme learning machine, ELM) 已经得到了广泛的研究和应用。ELM 是基于前馈神经网络 (single hidden-layer feedforward neural network, SLFN) 的最小二乘算法, 同时具有最小的训练误差和最小的权重范数, 可应用于回归问题和分类问题[1]。固定型 ELM 为了获得较好的学习能力, 通常采用高维的网络结构, 学习规模较大, 因此寻找最优隐节点个数和有效控制网络结构复杂性成为急需解决的问题。Huang 等[2] 提出了增量型极限学习机 (incremental extreme learning machine, I-ELM), 通过增加隐含层节点数减少训练误差, 但是其使用增量式策略后得到的新输出权重与具有同样隐含层参

收稿日期: 2019−04−17.
基金项目: 国家自然科学基金项目 (61170122).
通信作者: 左鹏玉. E-mail: 1253712018@qq.com.

数的标准 ELM 求得的输出权重结果不同。文献 [3] 提出了不同的增量式策略, 根据分块矩阵的广义逆矩阵分析确定输出权重, 且其具有 ELM 的最优性。以上所述均为批量学习算法, 只能将数据一次性输入给训练模型。而现实生活中, 很多数据都不是一次性获得的。数据依次加入到训练模型中, 批量学习算法需将旧的数据和新的数据一起重新训练, 需要花费大量的时间[4-5]。文献 [6] 提出了在线序列极限学习机 (online sequence extreme learning machine, OS-ELM), 可以将训练数据逐个或多个地加入到训练模型中, 丢掉已经训练过的数据以减少空间消耗。文献 [7] 提出了一种基于增量平均加权的在线序贯极限学习机算法 (incremental weighted average based online sequential extreme learning machine, IWOS-ELM), 利用原始数据来弱化增量数据的波动, 使在线学习机具有良好的稳定性。然而, 现实生活中存在着大量的不平衡数据, 例如生物医学应用和网络入侵等, ELM 并不适用于此类不平衡数据。在不平衡数据中, 分为多数类和少数类, 一般学习算法中大多数类将分离边界推向少数类, 以期望获得更好的自身分类效果[8-10]。文献 [11]提出了应用于不平衡数据的 W-ELM 算法。此算法增加了权数, 使得数据具有新的平衡程度。对于在线学习, 文献 [12] 在 OS-ELM 的基础上提出了加权在线序列极限学习机 (weighted online sequential extreme learning machine, WOS-ELM), 设置适当的权重, 使得不平衡分类的学习性能更好。但是这些在线学习算法有着和 ELM 一样的问题——如何寻找最优隐含层节点数。

本文提出了针对类别不平衡问题的增量在线序列极限学习机 (incremental and online sequence extreme learning machine for class imbalance learning, IOS-ELM)。首先根据类别不平衡比率调整平衡因子, 增大少数类样本的平衡因子使得分离超平面靠近多类样本。再根据分类误差大小决定是否增加隐节点数, 通常情况下隐节点数小于训练样本, 利用 Schur complement 公式增加隐节点; 当隐节点数较大时利用 Sherman-Morrison 公式增加隐节点。寻找到最优隐节点数后, 可逐个或多个地加入新训练样本获得更好的训练模型。

1　相关工作

1.1　极限学习机

ELM 随机产生隐含层参数且不需要进行调整, 通过最小二乘法直接确定隐含层的输出权重, 极大地提高了运行速度且具有良好的泛化性能[13]。ELM 是批量学习算法, 训练样本数是固定的。2006 年, Huang 等[1] 正式提出极限学习机的理论及应用。

$$\boldsymbol{y}_j=\sum_{i=1}^{l}\boldsymbol{\beta}_i g(\boldsymbol{w}_i,\boldsymbol{e}_i,\boldsymbol{x}_j),\ \ j=1,2,\cdots,t \tag{1}$$

式中: $\boldsymbol{y}_j$ 是第 j 个训练样本的输出值; $\boldsymbol{w}_i$ 为第 i 个隐含层节点的输入权重; $\boldsymbol{e}_i$ 为第 i 个隐含层节点的偏差; $\boldsymbol{x}_j$ 为第 j 个输入节点。由式 (1) 可推出输出权重 $\boldsymbol{\beta}$ 为

$$\boldsymbol{\beta}=\begin{cases}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})^{-1}\boldsymbol{Y}, & \text{样本数隐} < \text{节点数}\\ (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{Y}, & \text{隐节点数} < \text{样本数}\end{cases}$$

ELM 没有迭代调整的过程, 相对于传统的前馈神经网络极大地提高了学习速度。

1.2　在线序列极限学习机

OS-ELM 是 ELM 增加训练样本而得的一种在线学习算法, 具有 ELM 所有的优点。OS-ELM 包括一个初始的 ELM 批量学习过程和在线序列学习过程。在初始化阶段, 根据广义逆矩阵的公式, 初始的输出权重 $\boldsymbol{\beta}_0$ 的计算公式为

$$\boldsymbol{\beta}_0=\boldsymbol{K}_0^{-1}\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{Y}_0$$

式中: $\boldsymbol{H}_0$ 是隐含层输出矩阵; $\boldsymbol{K}_0=\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0$。当增加一个训练样本后, 隐含层输出矩阵 $\boldsymbol{H}_{t+1}$ 和训练样本的期望输出值 $\boldsymbol{Y}_{t+1}$ 分别为

$$\boldsymbol{H}_{t+1}=\begin{bmatrix}\boldsymbol{H}_t\\ \boldsymbol{h}_{t+1}\end{bmatrix},\ \ \boldsymbol{Y}_{t+1}=\begin{bmatrix}\boldsymbol{Y}_t\\ \boldsymbol{y}_{t+1}\end{bmatrix}$$

式中: $\boldsymbol{h}_{t+1}$ 为新增节点的隐含层输出矩阵; $\boldsymbol{y}_{t+1}$ 为新增训练样本的期望输出向量。当第 t+1 块数据集加入到训练样本时, 输出矩阵可由第 t 块数据集加入到训练样本后的输出矩阵分析求得, 计算公式为

$$\boldsymbol{\beta}_{t+1}=\boldsymbol{\beta}_t+\boldsymbol{K}_{t+1}^{-1}\boldsymbol{h}_{t+1}^{\mathrm{T}}(\boldsymbol{Y}_{t+1}-\boldsymbol{h}_{t+1}\boldsymbol{\beta}_t)$$

$$\boldsymbol{K}_{t+1}^{-1}=\boldsymbol{K}_t^{-1}-\boldsymbol{K}_t^{-1}\boldsymbol{h}_{t+1}^{\mathrm{T}}(\boldsymbol{I}+\boldsymbol{h}_{t+1}\boldsymbol{K}_t^{-1}\boldsymbol{h}_{t+1}^{\mathrm{T}})^{-1}\boldsymbol{h}_{t+1}\boldsymbol{K}_t^{-1}$$

1.3　无逆矩阵极限学习机

在极限学习机的模型中, 训练误差随着隐含层节点数的增加而减小。但在实验中, 考虑到计算复杂度, 应尽量减少隐含层节点数。为了平衡训练误差与计算复杂度这两个因素, 寻找隐含层节点数的最优值成为迫切需要解决的问题。无逆矩阵极限学习机 (IF-ELM) 应运而生。该算法采用了隐节点增加策略, 具有 l+1 个隐含层节点的输出权重可由原 l 个隐含层节点的输出权重求出, 而不需要重新计算所有的隐含层节点输出权重。

当增加一个隐含层节点时, 输入权重 $\boldsymbol{W}^{l+1}$ 和偏值 $\boldsymbol{E}^{l+1}$ 更新为如下形式:

$$\boldsymbol{W}^{l+1}=\left[\begin{array}{c}\boldsymbol{W}^{l}\\ \boldsymbol{w}_{l+1}\end{array}\right],\ \boldsymbol{E}^{l+1}=\left[\begin{array}{c}\boldsymbol{E}^{l}\\ \boldsymbol{e}_{l+1}\end{array}\right]$$

式中 $\boldsymbol{W}^{l}$ 和 $\boldsymbol{E}^{l}$ 是具有 l 个隐含层节点数的输入权重和偏差。$\boldsymbol{w}_{l+1}$ 为新增隐含层节点输出权重，$\boldsymbol{e}_{l+1}$ 为新增隐含层节点偏差，两者均为随机选取的参数。则具有 l+1 个隐含层节点的 ELM 的隐含层输出矩阵 $\boldsymbol{H}^{l+1}$ 为

$$\begin{aligned}&\boldsymbol{H}^{l+1}=g(\boldsymbol{W}^{l+1}\boldsymbol{X}+\boldsymbol{E}^{l+1})=\\&g\left(\left[\begin{array}{cc}\boldsymbol{W}^{l} & \boldsymbol{w}_{l+1}\end{array}\right]\boldsymbol{X}+\left[\begin{array}{cc}\boldsymbol{E}^{l} & \boldsymbol{e}_{l+1}\end{array}\right]\right)=[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]\end{aligned}\tag{2}$$

具有 l+1 个隐含层节点的输出权重 $\boldsymbol{\beta}^{l+1}$ 计算公式为

$$\begin{aligned}&\boldsymbol{\beta}^{l+1}=(\boldsymbol{H}^{l+1})^{\dagger}\boldsymbol{Y}=\\&([\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}])^{-1}[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{U}\boldsymbol{Y}\end{aligned}$$

式中：$(\boldsymbol{H}^{l+1})^{\dagger}$ 是 $\boldsymbol{H}^{l+1}$ 的广义逆矩阵[14]；$\boldsymbol{U}=([\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}\cdot[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}])^{-1}[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}$。为了避免产生过拟合现象，加入正则化项 a，则

$$\begin{aligned}&\boldsymbol{U}=(a^2\boldsymbol{I}_{l+1}+[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}])^{-1}[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}=\\&\left[\begin{array}{cc}a^2\boldsymbol{I}_l+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} & \boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}\\ \boldsymbol{h}^{\mathrm{T}}\boldsymbol{H} & a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h}\end{array}\right]^{-1}\left[\begin{array}{c}\boldsymbol{H}\\ \boldsymbol{h}\end{array}\right]=\\&\left[\begin{array}{cc}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{C} & \boldsymbol{D}\end{array}\right]\left[\begin{array}{c}\boldsymbol{H}\\ \boldsymbol{h}\end{array}\right]=\left[\begin{array}{c}\boldsymbol{A}\boldsymbol{H}+\boldsymbol{B}\boldsymbol{h}\\ \boldsymbol{C}\boldsymbol{H}+\boldsymbol{D}\boldsymbol{h}\end{array}\right]\end{aligned}$$

由 Schur complement 公式可得

$$\begin{aligned}&\boldsymbol{A}=\left[a^2\boldsymbol{I}_l+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}\right]^{-1}\\&\boldsymbol{B}=-\boldsymbol{A}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}\\&\boldsymbol{C}=-(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{A}\\&\boldsymbol{D}=-\boldsymbol{C}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}+(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}\end{aligned}$$

式中 $\boldsymbol{A}$ 可由 Sherman-Morrison 公式求得：

$$\begin{aligned}&\boldsymbol{A}=(a^2\boldsymbol{I}_l+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}+(a^2\boldsymbol{I}_l+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}\cdot\\&[\boldsymbol{I}+(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}(a^2\boldsymbol{I}_l+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{h}]^{-1}\cdot\\&(a^2+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{h})^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}(a^2\boldsymbol{I}_l+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\end{aligned}$$

2 基于类别不平衡的增量在线 ELM

2.1 算法思想

OS-ELM 算法通过不断地增加训练样本更好地反应数据模型。现实生活中有很多类别不平衡数据，为了获得更高的分类准确率 OS-ELM 算法将分离超平面推向少数类，降低了少类的识别率。此外，隐节点个数太少降低了分类准确率，但隐含层节点个数太多使网络结构变得复杂。OS-ELM 算法只是逐个增加训练样本个数，并未对隐含层节点个数进行调整。

本文提出了面向类别不平衡的无逆矩阵在线序列极限学习机。所提算法首先利用参数 λ 平衡类别不平衡数据中分离边界的距离和训练误差之间的关系，然后通过增加隐节点来调整网络结构，最后使用在线学习方式，训练模型在线的加入训练数据以更好地反映数据模型。

2.2 算法推导

现实生活中有很多类别不平衡现象，例如欺诈交易识别中，绝大多数交易属于正常交易，只有少数交易为欺诈交易，这就形成了类别不平衡现象[15]。在经典 ELM 中，为了获得更高的分类准确率，ELM 将分离超平面推向少数类，降低了少数类的识别率[16-17]。本文为了提高少数类的识别率，为每个不同的类别设置不同的 λ 值，少数类的 λ 值比多数类的 λ 大，将分离超平面推向多数类。新的 $\boldsymbol{\lambda}$ 值为 $k\times k$ 的矩阵，设置如下：

$$\lambda_{ii}=\frac{100}{\text{第}i\text{类样本个数}}$$

本文所提算法分为两种情况：一种为隐含层节点数小于训练样本数，另一种为训练样本数小于隐含层节点数。

1) 隐含层节点数小于训练样本数是比较常见的情况。将 $\boldsymbol{\lambda}$ 设置为矩阵后，具有 l 个隐含层节点的输出权重 $\boldsymbol{\beta}^{l}$ 由式 (1) 可得：

$$\boldsymbol{\beta}^{l}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{Y}\tag{3}$$

初始化阶段增加隐含层节点数后，隐含层输出权重 $\boldsymbol{H}^{l+1}$ 更新如式 (2) 所示，将式 (2) 代入到式 (3) 可得到新的输出权重 $\boldsymbol{\beta}^{l+1}$：

$$\begin{aligned}&\boldsymbol{\beta}^{l+1}=([\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]^{\mathrm{T}}\boldsymbol{\lambda}[\begin{array}{cc}\boldsymbol{H} & \boldsymbol{h}\end{array}]+\boldsymbol{I})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{Y}=\\&\left[\begin{array}{cc}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I}_{l\times l} & \boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}\\ \boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H} & \boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1\end{array}\right]^{-1}\left[\begin{array}{c}\boldsymbol{H}^{\mathrm{T}}\\ \boldsymbol{h}^{\mathrm{T}}\end{array}\right]\boldsymbol{\lambda}\boldsymbol{Y}=\\&\left[\begin{array}{cc}\boldsymbol{A}_1 & \boldsymbol{B}_1\\ \boldsymbol{C}_1 & \boldsymbol{D}_1\end{array}\right]\left[\begin{array}{c}\boldsymbol{H}^{\mathrm{T}}\\ \boldsymbol{h}^{\mathrm{T}}\end{array}\right]\boldsymbol{\lambda}\boldsymbol{Y}\end{aligned}$$

其中：

$$\begin{aligned}&\boldsymbol{A}_1=[\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I}_{l\times l}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}]^{-1}\\&\boldsymbol{B}_1=-\boldsymbol{A}_1\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}\\&\boldsymbol{C}_1=-(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}\boldsymbol{A}_1\\&\boldsymbol{D}_1=-\boldsymbol{C}_1\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}+(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}\end{aligned}$$

式中 $\boldsymbol{A}_1$ 可由 Sherman-Morrison 公式求得：

$$\begin{aligned}&\boldsymbol{A}_1=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I}_{l\times l})^{-1}+(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I}_{l\times l})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}\cdot\\&[\boldsymbol{I}+(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I}_{l\times l})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}]^{-1}\cdot\\&(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{h}+1)^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\lambda}\boldsymbol{H}+\boldsymbol{I}_{l\times l})^{-1}\end{aligned}$$

在线学习阶段，增加隐含层节点数以减小训练误差，当隐含层节点数与训练误差都具有合适的值的时候，再继续增加训练样本数，更多的样本以更好地反映数据模型。当增加样本时，参数 $\boldsymbol{\lambda}$、隐含层输出矩阵 $\boldsymbol{H}$ 和预期输出 $\boldsymbol{Y}$ 分别为

$$\boldsymbol{\lambda}_{t+1}=\left[\begin{array}{cc}\boldsymbol{\lambda}_t & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\lambda}_{t+1}\end{array}\right],\boldsymbol{H}_{t+1}=\left[\begin{array}{c}\boldsymbol{H}_t\\ \boldsymbol{h}_{t+1}\end{array}\right],\boldsymbol{Y}_{t+1}=\left[\begin{array}{c}\boldsymbol{Y}_t\\ \boldsymbol{y}_{t+1}\end{array}\right]$$

故在线学习的输出权重为

$$\begin{aligned}&\boldsymbol{\beta}_{t+1}=\boldsymbol{\beta}_t+\boldsymbol{K}_{t+1}^{-1}\boldsymbol{h}_{t+1}^{\mathrm{T}}\boldsymbol{\lambda}_{t+1}(\boldsymbol{Y}_{t+1}-\boldsymbol{h}_{t+1}\boldsymbol{\beta}_t)\\&\boldsymbol{K}_{t+1}^{-1}=\boldsymbol{K}_t^{-1}-\boldsymbol{K}_t^{-1}\boldsymbol{h}_{t+1}^{\mathrm{T}}\left(\boldsymbol{I}+\boldsymbol{\lambda}_{t+1}\boldsymbol{h}_{t+1}\boldsymbol{K}_t^{-1}\boldsymbol{h}_{t+1}{}^{\mathrm{T}}\right)^{-1}\boldsymbol{\lambda}_{t+1}\boldsymbol{h}_{t+1}\boldsymbol{K}_t^{-1}\end{aligned}$$

2) OS-ELM 和 IF-ELM 中都只讨论了隐含层

节点数小于训练样本数的情况，而当隐含层节点数大于训练样本数时，这两种算法都不符合最小二乘定律。有些数据结构比较复杂，数据之间的关系或是属性较多，此时需要较多的隐含层节点数。接下来将讨论隐含层节点数大于训练样本数的情况，初始化阶段，将 $\boldsymbol{\lambda}$ 设置为矩阵后，由式 (1) 可得具有 l 个隐含层节点的输出权重 $\boldsymbol{\beta}^l$ 为

$$\boldsymbol{\beta}^l = \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1}\boldsymbol{Y}$$

设

$$\boldsymbol{Q}^l = (\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1} \tag{4}$$

增加隐含层节点后，新的输出权重 $\boldsymbol{\beta}^{l+1}$ 为

$$\boldsymbol{\beta}^{l+1} = [\ \boldsymbol{H}\quad \boldsymbol{h}\ ]^{\mathrm{T}}([\ \boldsymbol{H}\quad \boldsymbol{h}\ ][\ \boldsymbol{H}\quad \boldsymbol{h}\ ]^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1}\boldsymbol{Y} = [\ \boldsymbol{H}\quad \boldsymbol{h}\ ]^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{h}\boldsymbol{h}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1}\boldsymbol{Y} = [\ \boldsymbol{H}\quad \boldsymbol{h}\ ]^{\mathrm{T}}\boldsymbol{Q}^{l+1}\boldsymbol{Y}$$

式中 $\boldsymbol{Q}^{l+1} = (\boldsymbol{I} + \boldsymbol{C}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{C}\boldsymbol{h}\boldsymbol{h}^{\mathrm{T}})^{-1}$，根据 Sherman-Morrison 公式可得

$$\boldsymbol{Q}^{l+1} = (\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1} - (\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1}\boldsymbol{h}\cdot [\boldsymbol{I} + \boldsymbol{h}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1}\boldsymbol{h}]^{-1}\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\lambda}^{-1})^{-1} \tag{5}$$

将式 (4) 代入到式 (5) 中得

$$\boldsymbol{Q}^{l+1} = \boldsymbol{Q}^l - \boldsymbol{Q}^l\boldsymbol{h}(\boldsymbol{I} + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{Q}^l\boldsymbol{h})^{-1}\boldsymbol{h}^{\mathrm{T}}\boldsymbol{Q}^l$$

在线学习阶段增加新训练样本后，参数 $\boldsymbol{\lambda}$、隐含层输出矩阵 $\boldsymbol{H}$ 和预期输出 $\boldsymbol{Y}$ 的变化与隐含层节点数小于训练样本数的情况相同，增加训练样本后新的输出权重为

$$\boldsymbol{\beta}^{t+1} = \begin{bmatrix} \boldsymbol{H}_t \\ \boldsymbol{h}_{t+1} \end{bmatrix}^{\mathrm{T}}\left(\begin{bmatrix} \boldsymbol{H}_t \\ \boldsymbol{h}_{t+1} \end{bmatrix}\begin{bmatrix} \boldsymbol{H}_t \\ \boldsymbol{h}_{t+1} \end{bmatrix}^{\mathrm{T}} + \begin{bmatrix} \boldsymbol{\lambda}_t & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\lambda}_{t+1} \end{bmatrix}^{-1}\right)^{-1}\begin{bmatrix} \boldsymbol{Y}_t \\ \boldsymbol{y}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_t \\ \boldsymbol{h}_{t+1} \end{bmatrix}^{\mathrm{T}}\begin{pmatrix} \boldsymbol{H}_t\boldsymbol{H}_t^{\mathrm{T}} + \boldsymbol{\lambda}_t & \boldsymbol{H}_t\boldsymbol{h}_{t+1}^{\mathrm{T}} \\ \boldsymbol{h}_{t+1}\boldsymbol{H}_t^{\mathrm{T}} & \boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1} \end{pmatrix}^{-1}\begin{bmatrix} \boldsymbol{Y}_t \\ \boldsymbol{y}_{t+1} \end{bmatrix}$$

由 Schur complement 公式可得：

$$\begin{bmatrix} \boldsymbol{O} & \boldsymbol{P} \\ \boldsymbol{R} & \boldsymbol{S} \end{bmatrix} = \begin{pmatrix} \boldsymbol{H}_t\boldsymbol{H}_t^{\mathrm{T}} + \boldsymbol{\lambda}_t & \boldsymbol{H}_t\boldsymbol{h}_{t+1}^{\mathrm{T}} \\ \boldsymbol{h}_{t+1}\boldsymbol{H}_t^{\mathrm{T}} & \boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1} \end{pmatrix}^{-1}$$

其中：

$$\boldsymbol{O} = \left[\boldsymbol{H}_t\boldsymbol{H}_t^{\mathrm{T}} + \boldsymbol{\lambda}_t - \boldsymbol{H}_t\boldsymbol{h}_{t+1}^{\mathrm{T}}(\boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1})^{-1}\boldsymbol{h}_{t+1}\boldsymbol{H}_t^{\mathrm{T}}\right]^{-1}$$

$$\boldsymbol{P} = -\boldsymbol{O}\boldsymbol{H}_t\boldsymbol{h}_{t+1}^{\mathrm{T}}(\boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1})^{-1}$$

$$\boldsymbol{R} = -(\boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1})^{-1}\boldsymbol{h}_{t+1}\boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{O}$$

$$\boldsymbol{S} = -\boldsymbol{R}\boldsymbol{H}_t\boldsymbol{h}_{t+1}^{\mathrm{T}}(\boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1})^{-1} + (\boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1})^{-1}$$

设 $\boldsymbol{\gamma} = (\boldsymbol{H}_t\boldsymbol{H}_t^{\mathrm{T}} + \boldsymbol{\lambda}_t)^{-1}$，$\boldsymbol{\iota} = (\boldsymbol{h}_{t+1}\boldsymbol{h}_{t+1}^{\mathrm{T}} + \boldsymbol{\lambda}_{t+1})^{-1}\boldsymbol{h}_{t+1}\boldsymbol{H}_t^{\mathrm{T}}$，$\boldsymbol{\eta} = \boldsymbol{H}_t\boldsymbol{h}_{t+1}^{\mathrm{T}}$，根据 Sherman-Morrison 公式，可得

$$\boldsymbol{O} = \boldsymbol{\gamma} + \boldsymbol{\gamma}\boldsymbol{\eta}(\boldsymbol{I} + \boldsymbol{\iota}\boldsymbol{\gamma}\boldsymbol{\eta})^{-1}\boldsymbol{\iota}\boldsymbol{\gamma}$$

3　实验结果

为了验证本文所提 IOS-ELM 算法的有效性，利用 keel 数据集和 UCI 数据集对 W-ELM、IF-ELM-SMOTE、OS-ELM- SMOTE、EWOS-ELM 和所提 IOS-ELM 算法进行测试。实验数据集分别是：Dermatology-6、Abalone9-18、Yeast1、Shuttle-c0-vs-c4、Segment0、Abalone19、Pageblocks0、Pdspeechfeaters、Vehicle1、Vehicle3、Biodge、DNA、Satimage、USPS，具体描述如表 1 所示。本文所有实验均在同一环境下完成，采用在 Windows 10 环境下搭建系统，计算机处理器配置为 Intel® CoreTM i5-8400 CPU@2.8 GHz，内存 12 GB，MATLAB2016b 下完成。

表 1　实验数据集

Table 1　Experimental datasets

数据集	样本数	类别不平衡率	特征数	类别数
Dermatology-6	358	16.9	34	2
Abalone9-20	731	16.4	8	2
Shuttle-c0-vs-c4	1 829	13.87	9	2
Segment0	2 308	6.02	19	2
Abalone19	4 174	129.44	8	2
Pageblocks0	5 472	8.79	10	2
Pdspeechfeaters	756	2.94	754	2
Vehicle1	846	2.9	20	2
Vehicle3	846	2.99	20	2
Biodge	1 055	1.96	41	2
DNA	2 000	2.25	200	3
Satimage	6 435	2.38	36	6
USPS	9 298	2.19	256	10

实验中，将所有数据归一化到 [−1,1] 区间中。ELM 网络的激活函数均为 Sigmoid 函数，为了保证实验的有效性，实验使用五折交叉验证法，每组数据进行 20 次实验，最终结果为 20 次实验结果的平均值。为了确保 IF-ELM 和 IOS-ELM 算法网络结构不会无休止增长，隐含层节点最大增长个数为 50。IF-ELM、OS-ELM 和 WEOS-ELM 算法使用 SMOTE 作为过采样方法[15]。SMOTE 中的 k 值设置为 5，若少数类样本数量较少，则 k 值相应地减小。本文采用类别不平衡领域中的常用评价性能指标几何平均数 (geometric_mean, G-mean) 来比较各个算法的分类性能[17]。对于多类问题，本文将多类划分为多个二类问题，求出每个二类问题的 G-mean 值，取其平均值作为多类分类最终实验结果。

表 2 给出了隐节点数小于训练样本数的不同 ELM 算法二分类实验结果。大部分的二分类实验中本文所提出的 IOS-ELM 算法的 G-mean 值最

高且训练时间也较少。以 Dermatology6 数据集为例，初始的隐含层节点数为 5，误差终止条件为 tempmean=0.98。IOS-ELM 算法的 G-mean=0.96，训练时间为 0.078 9 s，分类准确率明显高于其他 3 种算法。表 3 给出了隐含层节点数大于训练样本的结果。在隐节点数大于训练样本时，初始时隐含层节点数较多，增加隐节点数对实验结果影响较小。隐节点数过大也导致训练时间较多。表 4 给出了多类分类实验结果，证明 IOS-ELM 算法对多类分类实验也有很好的学习性能。

表 2　隐节点数大于训练样本数的二分类实验结果

Table 2　Two-class experimental results with the number of hidden nodes more than the number of training samples

数据集	算法	几何平均数	时间/s	训练样本几何平均数	初始隐节点数	初始训练样本数	每次新增训练样本数
Dermatology6	IOSELM	0.942 9	0.065 6	0.98	5	200	1
	WELM	0.835 3	0.023 1	—	5	200	1
	IFELM-SMOTE	0.897 3	118.417 9	0.98	5	200	1
	OSELM-SMOTE	0.839 1	0.031 2	—	5	200	1
	EWOSELM-SMOTE	0.853 7	0.478 9	—	5	200	1
Abalone9-18	IOSELM	0.869 2	1.262 5	0.86	5	400	1
	WELM	0.758 6	0.070 7	—	5	400	1
	IFELM-SMOTE	0.849 8	477.331 2	0.86	5	400	1
	OSELM-SMOTE	0.808 4	0.036 7	—	5	400	1
	EWOSELM-SMOTE	0.728 6	0.719 5	—	5	400	1
Yeast1	IOSELM	0.704 3	0.485 9	0.71	10	700	10
	WELM	0.679 6	0.069 6	—	10	700	10
	IFELM-SMOTE	0.702 1	104.203 1	0.71	10	700	10
	OSELM-SMOTE	0.697 6	0.025 1	—	10	700	10
	EWOSELM-SMOTE	0.700 1	0.739 1	—	10	700	10
Shuttle-c0-vs-c4	IOSELM	0.995 4	0.078 1	0.99	10	1 000	10
	WELM	0.995 7	0.121 2	—	10	1 000	10
	IFELM-SMOTE	0.994 9	4.834 3	0.99	10	1 000	10
	OSELM-SMOTE	0.996 6	0.035 9	—	10	1 000	10
	EWOSELM-SMOTE	0.996 7	1.629 6	—	10	1 000	10
Segment0	IOSELM	0.918 6	6.684 3	0.95	10	1 000	10
	WELM	0.761 8	0.281 6	—	10	1 000	10
	IFELM-SMOTE	0.893 2	3 425.397 6	0.95	10	1 000	10
	OSELM-SMOTE	0.865 2	0.052 3	—	10	1 000	10
	EWOSELM-SMOTE	0.896 0	2.970 3	—	10	1 000	10
Abalone19	IOSELM	0.755 3	19.647 6	0.75	10	2 000	10
	WELM	0.667 2	1.291 7	—	10	2 000	10

续表 2

数据集	算法	几何平均数	时间/s	训练样本几何平均数	初始隐节点数	初始训练样本数	每次新增训练样本数
	IFELM-SMOTE	0.647 6	637.358 5	0.75	10	2 000	10
	OSELM-SMOTE	0.664 1	0.093 7	—	10	2 000	10
	EWOSELM-SMOTE	0.564 2	2.291 4	—	10	2 000	10
Pageblocks0	IOSELM	0.849 4	26.216 4	0.88	10	2 000	10
	WELM	0.801 8	3.220 0	—	10	2 000	10
	IFELM-SMOTE	0.871 4	12 476.750 1	0.88	10	2 000	10
	OSELM-SMOTE	0.826 6	0.104 6	—	10	2 000	10
	EWOSELM-SMOTE	0.819 0	4.114 8	—	10	2 000	10

表 3　训练样本数大于隐节点数的二分类实验结果

Table 3　Two-class experimental results with the number of training samples more than the number of hidden nodes

数据集	算法	几何平均数	时间/s	训练样本几何平均数	初始隐节点数	初始训练样本数	每次新增训练样本数
Pdspeechfeaters	IOSELM	0.683 6	38.064 8	0.99	700	400	1
	WELM	0.562 5	4.513 8	—	700	400	1
	IFELM-SMOTE	0.526 6	3 184.181 2	0.99	700	400	1
	OSELM-SMOTE	0.656 7	14.027 3	—	700	400	1
	EWOSELM-SMOTE	0.648 7	286.821 8	—	700	400	1
Vehicle1	IOSELM	0.815 5	65.702 3	0.9	700	400	1
	WELM	0.774 6	4.374 2	—	700	400	1
	IFELM-SMOTE	0.646 6	46.496 0	0.9	700	400	1
	OSELM-SMOTE	0.765 8	22.901 5	—	700	400	1
	EWOSELM-SMOTE	0.767 3	457.938 2	—	700	400	1
Vehicle3	IOSELM	0.811 4	60.600 7	0.84	700	400	1
	WELM	0.757 4	4.387 3	—	700	400	1
	IFELM-SMOTE	0.657 9	46.917 1	0.84	700	400	1
	OSELM-SMOTE	0.764 4	22.427 3	—	700	400	1
	EWOSELM-SMOTE	0.769 9	451.362 5	—	700	400	1
Biodeg	IOSELM	0.842 0	8.399 2	0.89	700	600	10
	WELM	0.893 8	0.640 9	—	700	600	10
	IFELM-SMOTE	0.699 8	4.734 3	0.89	700	600	10
	OSELM-SMOTE	0.524 8	2.888 2	—	700	600	10
	EWOSELM-SMOTE	0.516 3	210.161 7	—	700	600	10

表 4　多类分类实验结果

Table 4　Experimental results of the multi-class classification

数据集	算法	几何平均数	时间/s	训练样本几何平均数	初始训练隐含层节点数	初始训练样本数	每次新增训练样本数
DNA	IOSELM	0.787 7	8.223 4	0.86	20	1 000	10
	WELM	0.645 3	0.210 6	—	20	1 000	10
	IFELM-SMOTE	0.824 3	1 731.592 9	0.86	20	1 000	10
	OSELM-SMOTE	0.656 1	0.046 8	—	20	1 000	10
	EWOSELM-SMOTE	0.657 1	2.383 5	—	20	1 000	10
Satimage	IOSELM	0.884 8	16.314 0	0.9	20	3 000	10
	WELM	0.855 3	2.548 9	—	20	3 000	10
	IFELM-SMOTE	0.863 9	14 529.904 6	0.9	20	3 000	10
	OSELM-SMOTE	0.855 6	0.085 1	—	20	3 000	10
	EWOSELM-SMOTE	0.845 1	1.681 2	—	20	3 000	10
USPS	IOSELM	0.895 6	118.898 4	0.9	20	6 000	10
	WELM	0.775 5	8.163 4	—	20	6 000	10
	IFELM-SMOTE	0.727 2	99 448.959 3	0.9	20	6 000	10
	OSELM-SMOTE	0.796 8	0.226 5	—	20	6 000	10
	EWOSELM-SMOTE	0.786 4	6.308 5	—	20	6 000	10

4　结束语

本文针对类别不平衡环境下的增量学习问题，提出了面对类别不平衡的增量在线极限学习机算法，即 IOS-ELM 算法。ISO-ELM 算法利用 Schur complement 公式增加隐含层节点获得连接权重的最优解。再引入在线学习思想，使训练样本可以逐个或多个地加入到训练模型中。最后调节惩罚因子的大小使其适用于类别不平衡环境下的学习。针对隐含层节点数小于或大于训练样本数两种情况，本文分别给出了理论推理。实验证明，与对比算法相比 IOS-ELM 算法具有较好的泛化性能和在线预测能力。

参考文献：

[1] HUANG Guangbin, ZHU Qinyu, SIEW C K. Extreme learning machine: theory and applications[J]. Neurocomputing, 2006, 70(1/2/3): 489–501.

[2] HUANG Guangbin, CHEN Lei, SIEW C K. Universal approximation using incremental constructive feedforward networks with random hidden nodes[J]. IEEE transactions on neural networks, 2006, 17(4): 879–892.

[3] LI Shuai, YOU Zhuhong, GUO Hongliang, et al. Inverse-free extreme learning machine with optimal information updating[J]. IEEE transactions on cybernetics, 2016, 46(5): 1229–1241.

[4] HUANG Shan, WANG Botao, CHEN Yuemei, et al. An efficient parallel method for batched OS-ELM training using MapReduce[J]. Memetic computing, 2017, 9(3): 183–197.

[5] KIM Y, TOH K A, TEOH A B J, et al. An online learning network for biometric scores fusion[J]. Neurocomputing, 2013, 102: 65–77.

[6] LIANG Nanying, HUANG Guangbin, SARATCHANDRAN P, et al. A fast and accurate online sequential learning algorithm for feedforward networks[J]. IEEE transactions on neural networks, 2006, 17(6): 1411–1423.

[7] 张明洋, 闻英友, 杨晓陶, 等. 一种基于增量加权平均的在线序贯极限学习机算法 [J]. 控制与决策, 2017, 32(10): 1887–1893.
ZHANG Mingyang, WEN Yingyou, YANG Xiaotao, et al. An incremental weighted average based online sequential extreme learning machine algorithm[J]. Control and decision, 2017, 32(10): 1887–1893.

[8] DOUZAS G, BACAO F, LAST F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE[J]. Information sciences, 2018, 465: 1–20.

[9] BATUWITA R, PALADE V. Class imbalance learning methods for support vector machines[M]//HE Haibo, MA Yunqian. Imbalanced Learning: Foundations, Algorithms, and Applications. New York: John Wiley & Sons, Inc., 2013: 145–168.

[10] XIA Shixiong, MENG Fanrong, LIU Bing, et al. A Kernel Clustering-based possibilistic fuzzy extreme learning machine for class imbalance learning[J]. Cognitive computation, 2015, 7(1): 74–85.

[11] ZONG Weiwei, HUANG Guangbin, CHEN Yiqiang. Weighted extreme learning machine for imbalance learning[J]. Neurocomputing, 2013, 101: 229–242.

[12] MIRZA B, LIN Zhiping, TOH K A. Weighted online sequential extreme learning machine for class imbalance learning[J]. Neural processing letters, 2013, 38(3): 465–486.

[13] 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.

[14] RAO C R, MITRA S K. Generalized inverse of a matrix and its applications[C]//Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics. Berkeley, : University of California Press, 1972: 601–620.

[15] BATUWITA R, PALADE V. FSVM-CIL: fuzzy support vector machines for class imbalance learning[J]. IEEE transactions on fuzzy systems, 2010, 18(3): 558–571.

[16] DING Shuya, MIRZA B, LIN Zhiping, et al. Kernel based online learning for imbalance multiclass classification[J]. Neurocomputing, 2017, 277: 139–148.

[17] HE H, GARCIA E A. Learning from imbalance data[J]. IEEE transactions on knowledge and data engineering, 2009, 21(9): 1263–1284.

作者简介：

左鹏玉，硕士研究生，主要研究方向为人工智能、模式识别。

周洁，博士研究生，主要研究方向为人工智能、模式识别、机器学习。

王士同，教授，博士生导师，CCF 会员，主要研究方向为人工智能、模式识别。作为第一作者发表学术论文百余篇。