

# 控制与决策

Control and Decision

## 基于M-estimator的鲁棒宽度学习系统

郭威, 徐涛

引用本文:

郭威, 徐涛. 基于M-estimator的鲁棒宽度学习系统[J]. 控制与决策, 2023, 38(4): 1039–1046.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.1479>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 融合块对角约束的鲁棒低秩多核聚类

Low-rank robust multiple kernel clustering with block diagonal constraints

控制与决策. 2022, 37(11): 2977–2983 <https://doi.org/10.13195/j.kzyjc.2021.0566>

#### 不完备数据的鲁棒多视角图学习及其聚类应用

Robust multiview graph learning with applications to clustering for incomplete data

控制与决策. 2022, 37(12): 3251–3258 <https://doi.org/10.13195/j.kzyjc.2021.0827>

#### 基于数据分布特性的代价敏感宽度学习系统

Data distribution-based cost-sensitive broad learning system

控制与决策. 2021, 36(7): 1686–1692 <https://doi.org/10.13195/j.kzyjc.2019.1484>

#### 一种基于池计算的宽度学习系统

A broad learning system based on reservoir computing

控制与决策. 2021, 36(9): 2203–2210 <https://doi.org/10.13195/j.kzyjc.2019.1729>

#### 基于深度强化学习与迭代贪婪的流水车间调度优化

Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method

控制与决策. 2021, 36(11): 2609–2617 <https://doi.org/10.13195/j.kzyjc.2020.0608>

# 基于 M-estimator 的鲁棒宽度学习系统

郭威<sup>1,2†</sup>, 徐涛<sup>3</sup>

- 盐城师范学院 江苏省心理与认知科学大数据重点建设实验室, 江苏 盐城 224002;
- 盐城师范学院 信息工程学院, 江苏 盐城 224002;
- 南京航空航天大学 计算机科学与技术学院, 南京 210016)

**摘要:** 宽度学习系统(BLS)是最近提出的一种准确且高效的新兴机器学习算法,已在分类、回归等问题中展现出优越的学习性能.然而,传统 BLS 以最小二乘作为学习准则,易受到离群值的干扰从而生成不准确的学习模型.鉴于此,提出一种基于 M-estimator 的鲁棒宽度学习系统(RBLS).与 BLS 不同,RBLS 在学习模型中使用具有鲁棒特性的 M-estimator 代价函数替代传统的最小二乘代价函数,并采用拉格朗日乘子法和迭代加权最小二乘方法进行优化求解.在迭代学习过程中,正常样本和离群值样本将根据其训练误差的大小而被逆向赋予不同的权重,从而有效地抑制或消除离群值误差对学习模型的不利影响.实验结果表明,作为一种统一的鲁棒学习框架,RBLS 可以融合使用不同的 M-estimator 加权策略,且能够取得更好的泛化性能和鲁棒性.

**关键词:** 宽度学习系统; 离群值; 鲁棒性; M 估计; 迭代加权最小二乘; 拉格朗日乘子法

中图分类号: TP183

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1479

开放科学(资源服务)标识码(OSID):



引用格式: 郭威,徐涛.基于 M-estimator 的鲁棒宽度学习系统[J].控制与决策,2023,38(4):1039-1046.

## M-estimator-based robust broad learning system

GUO Wei<sup>1,2†</sup>, XU Tao<sup>3</sup>

- Jiangsu Provincial Key Constructive Laboratory for Big Data of Psychology and Cognitive Science, Yancheng Teachers University, Yancheng 224002, China;
- College of Information Engineering, Yancheng Teachers University, Yancheng 224002, China;
- College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract:** The broad learning system (BLS) is an accurate and efficient machine learning algorithm proposed recently, which has shown excellent performance in classification, regression and other problems. However, the traditional BLS takes least squares as learning criterion, which is prone to be affected by outliers and thus generates inaccurate learning models. To solve this problem, this paper proposes a robust broad learning system (RBLS) based on an M-estimator. Different from the BLS, the RBLS uses a robust M-estimator cost function to replace the traditional least squares cost function in the learning model, and adopts the Lagrange multiplier method and the iteratively reweighted least squares method to seek for an optimal solution. In the iterative learning process, the normal sample and the outlier sample will be reversely assigned different weights according to the size of their training errors, so as to effectively suppress or eliminate the adverse effects of the outlier residual on the learning model. Experimental results show that, as a unified robust learning framework, the RBLS can combine different M-estimator weighting strategies and achieve better generalization performance and robustness than the comparison algorithms.

**Keywords:** broad learning system; outliers; robustness; M-estimator; iteratively reweighted least squares; Lagrange multiplier

## 0 引言

近年来,深度学习作为人工智能领域最热门的研究方向,在计算机视觉、语音识别、自然语言处理等诸

多领域取得了重大突破和广泛应用<sup>[1]</sup>.尽管深度学习具有强大的学习能力,但由于其模型结构复杂,且涉及大量超参数的迭代调整,导致深度学习算法的训练

收稿日期: 2021-08-23; 录用日期: 2022-01-28.

基金项目: 国家自然科学基金项目(61603326); 江苏省心理与认知科学大数据重点建设实验室开放基金项目(72591962004G).

责任编辑: 阳春华.

†通讯作者. E-mail: weigu031@163.com.

过程非常耗时.此外,对于需要进行增量学习的流数据或动态数据,深度学习模型的动态更新较为困难.

宽度学习系统(broad learning system, BLS)是Chen等<sup>[2]</sup>于2018年提出的一种全新的神经网络学习框架.与深度学习模型相比,BLS通过扩展神经网络的宽度而非深度以实现数据拟合,具有网络结构简单、模型参数少等优点,并且网络的输出权值可通过岭回归算法直接计算得到而无需迭代调整,因此学习速度极快.此外,BLS具有良好的扩展性,可根据实际需要实现网络模型的快速增量重构,以满足大规模时变数据和流式数据的学习需求.鉴于其优秀的学习特性,BLS自提出以来受到学者们的广泛关注,并在理论研究和应用研究中得到了快速发展.文献[3]从数学方面验证了BLS具有通用的学习逼近能力,并给出BLS的几种级联变型结构.文献[4]将Takagi-Sugeno模糊系统融合至BLS,提出一种模糊宽度学习系统(FBLS)并用于多视图高动态范围图像合成<sup>[5]</sup>.为处理包含大量无标记样本和少数有标记样本的数据集,文献[6]基于流形正则化框架对BLS进行扩展并提出一种半监督BLS.针对不均衡数据集的模式分类问题,文献[7]提出一种基于数据分布特性的代价敏感BLS.为进一步提高BLS的学习精度,文献[8]将BLS从宽度结构向深度结构扩展,提出一种宽深结合的Stacked BLS模型.此外,BLS在许多现实问题中也获得了成功应用,如疫情防控中的口罩佩戴检测<sup>[9]</sup>、足球比赛可视化分析<sup>[10]</sup>、工业过程建模<sup>[11]</sup>、文本分类<sup>[12]</sup>、时间序列预测<sup>[13-15]</sup>等.

尽管传统BLS及其改进方法在标杆数据集和实际应用数据集展现出良好的泛化性能和实用效果,但均是在训练样本不含噪声或仅包含高斯噪声的条件下得到的.然而在实际应用中,由于设备退化、人为错误等原因,实际采集的数据样本中不可避免地会存在少量的离群值.离群值的存在会严重影响BLS学习建模的准确性并生成过拟合的学习模型,从而导致其泛化性能严重退化.为了增强BLS对于离群值的抗干扰能力和鲁棒性,本文提出一种基于M-estimator的鲁棒宽度学习系统(robust broad learning system, RBLs),主要内容如下.

1) 针对离群值环境下的数据建模问题,将M-estimator鲁棒学习技术引入BLS,提出一种统一的鲁棒宽度学习框架RBLs.该框架通过融合使用不同的M-estimator代价函数及其对应的加权策略对训练样本进行逆向加权计算,以减轻或消除离群值误差对学习模型的不利影响,从而有效地提高BLS对离群值

的抗干扰能力和鲁棒性.

2) 以RBLs统一学习框架为基础,并结合Huber、Bisquare、Cauchy这3种典型的M-estimator函数,对RBLs的抗离群值能力和鲁棒性进行理论分析和实证研究,并得出一些有意义的结果和结论,为RBLs的进一步拓展与应用提供理论支撑和实践指导.

## 1 宽度学习系统

宽度学习系统(BLS)是最近提出的一种新型随机参数神经网络,为分类、回归等问题提供了一种有效、高效且无需深层结构的宽度学习框架<sup>[2]</sup>.

对于训练集 $\{(\mathbf{X}, \mathbf{Y}) | \mathbf{X} \in \mathbf{R}^{N \times d}, \mathbf{Y} \in \mathbf{R}^{N \times c}\}$ ,  $N$ 为训练样本总数,  $d$ 和 $c$ 分别为输入和输出的维度.具有 $n$ 个特征映射窗口的BLS首先将输入数据 $\mathbf{X}$ 映射到 $n$ 组随机的特征空间,有

$$\mathbf{F}_i = \phi_i(\mathbf{X}\mathbf{W}_{f_i} + \mathbf{b}_{f_i}), \quad i = 1, 2, \dots, n. \quad (1)$$

其中: $\phi_i(\cdot)$ 为特征层激活函数; $\mathbf{W}_{f_i} \in \mathbf{R}^{d \times k}$ 和 $\mathbf{b}_{f_i} \in \mathbf{R}^{N \times k}$ 为随机生成的第 $i$ 个特征映射组的权值和偏置, $k$ 为每个特征映射组中特征节点个数.将所有 $n$ 个映射特征串接构成完整的特征空间 $\mathbf{F}^n \equiv [\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_n]$ ,然后以 $\mathbf{F}^n \in \mathbf{R}^{N \times nk}$ 作为输入并通过非线性变换将其转换为 $m$ 组增强节点,即

$$\mathbf{E}_j = \xi_j(\mathbf{F}^n \mathbf{W}_{e_j} + \mathbf{b}_{e_j}), \quad j = 1, 2, \dots, m. \quad (2)$$

其中: $\xi_j$ 为增强层激活函数; $\mathbf{W}_{e_j} \in \mathbf{R}^{nk \times q}$ 和 $\mathbf{b}_{e_j} \in \mathbf{R}^{N \times q}$ 为随机生成的第 $j$ 组增强节点的权值和偏置, $q$ 为每个增强组中增强节点个数.类似地,将所有 $m$ 组增强节点串接为 $\mathbf{E}^m \equiv [\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_m]$ .

最后,将所有特征节点 $\mathbf{F}^n$ 与增强节点 $\mathbf{E}^m$ 串联并连接到输出层,则BLS模型为

$$\mathbf{Y} = \mathbf{A}\mathbf{W}. \quad (3)$$

其中: $\mathbf{A} = [\mathbf{F}^n | \mathbf{E}^m]$ ,  $\mathbf{W}$ 为连接特征节点和增强节点到输出层的输出权值.考虑到 $\mathbf{A}$ 有可能为病态矩阵,因此,求解输出权值 $\mathbf{W}$ 通常需要优化如下带正则化项的目标函数:

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{A}\mathbf{W}\|_2^2 + \lambda \|\mathbf{W}\|_2^2, \quad (4)$$

其中 $\lambda$ 为正则化参数.根据岭回归算法, $\mathbf{W}$ 最终可表示为

$$\mathbf{W} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{Y}. \quad (5)$$

## 2 基于M-estimator的鲁棒宽度学习系统

### 2.1 RBLs学习模型

如式(4)所示,原始BLS采用最小二乘(least squares, LS)作为学习准则,追求训练误差平方和的最小化,式(4)可重写为

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{i=1}^N (e_i)^2 + \lambda \|\mathbf{W}\|_2^2; \\ \text{s.t.} \quad & e_i = y_i - \mathbf{A}_i \mathbf{W}. \end{aligned} \quad (6)$$

其中:  $e_i$  为实际输出与网络输出之间的误差,  $\mathbf{A}_i$  为矩阵  $\mathbf{A}$  的第  $i$  个行向量. 然而, LS 学习准则对于离群值的天然敏感性使其易于生成过拟合的学习模型, 从而导致 BLS 在离群值环境下的泛化性将严重下降. 直观上分析, 当某个训练样本为离群值时, 其对应的训练误差将严重偏离正常误差范围, 而 LS 方法仍然对其进行平方计算, 在此情形下模型的目标函数是有偏的, 据此训练得到的学习模型也将是不准确的.

为了增强 BLS 对于离群值的抗干扰能力, 本文引入具有鲁棒特性的 M-estimator 技术替代传统的 LS 学习准则, 提出一种基于 M-estimator 的鲁棒宽度学习模型 (RBLS), 其目标函数如下:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{i=1}^N \rho(e_i) + \frac{1}{2} \lambda \|\mathbf{W}\|_2^2; \\ \text{s.t.} \quad & e_i = y_i - \mathbf{A}_i \mathbf{W}. \end{aligned} \quad (7)$$

其中  $\rho(\cdot)$  为 M-estimator 代价函数.  $\rho(\cdot)$  通常采用一些比较柔和的分段函数, 当训练误差明显偏大时, 其增长趋势较 LS 估计量更加平缓甚至保持不变, 以减轻或避免对离群值误差进行错误反馈, 从而有效地抑制或消除离群值对学习模型的负面影响. 显然, 当  $\rho(e) = e^2/2$  时, RBLS 退化为 BLS, 即 BLS 为 RBLS 在其 M-estimator 函数取 LS 时的特例.

## 2.2 RBLS 的优化求解

RBLS 模型可采用拉格朗日乘子法进行优化求解. 根据 KKT 定理, 求解式 (7) 等价于求解如下对偶优化问题:

$$L(\mathbf{W}, e, \alpha) = \sum_{i=1}^N \rho(e_i) + \frac{1}{2} \lambda \|\mathbf{W}\|_2^2 + \sum_{i=1}^N \alpha_i (y_i - \mathbf{A}_i \mathbf{W} - e_i), \quad (8)$$

其中  $\alpha_i (i = 1, 2, \dots, N)$  为式 (7) 中第  $i$  个等式约束所对应的拉格朗日乘子. 将拉格朗日函数  $L(\mathbf{W}, e, \alpha)$  分别对  $\mathbf{W}$ 、 $e_i$  和  $\alpha_i$  求导并令导数为 0, 得到如下 KKT 优化条件:

$$\begin{cases} \frac{\partial L}{\partial \mathbf{W}} = 0 \Rightarrow \mathbf{W} = \frac{1}{\lambda} \sum_{i=1}^N \alpha_i \mathbf{A}_i^T; & (9a) \\ \frac{\partial L}{\partial e_i} = 0 \Rightarrow \alpha_i = \frac{\partial \rho(e_i)}{\partial e_i} = \varphi(e_i), & (9b) \\ i = 1, 2, \dots, N; \\ \frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow y_i - \mathbf{A}_i \mathbf{W} - e_i = 0, & (9c) \\ i = 1, 2, \dots, N, \end{cases}$$

其中  $\varphi(e_i)$  为  $\rho(\cdot)$  对于误差  $e_i$  的梯度. 定义

$$\theta(e_i) = \varphi(e_i)/e_i \quad (10)$$

为关于误差  $e_i$  的加权函数, 令  $\theta_i = \theta(e_i)$ , 则式 (9) 可重写为如下矩阵(向量)形式:

$$\begin{cases} \lambda \mathbf{W} = \mathbf{A}^T \boldsymbol{\alpha}, & (11a) \\ \boldsymbol{\alpha} = \boldsymbol{\theta} \mathbf{e}, & (11b) \\ \mathbf{A} \mathbf{W} - \mathbf{Y} + \mathbf{e} = \mathbf{0}. & (11c) \end{cases}$$

其中:  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$  为拉格朗日乘子向量,  $\mathbf{e} = [e_1, e_2, \dots, e_N]^T$  为误差向量,  $\boldsymbol{\theta} = \text{diag}(\theta_1, \theta_2, \dots, \theta_N)$  为加权矩阵.

由式 (11c), 有

$$\mathbf{A}^T \boldsymbol{\theta} \mathbf{A} \mathbf{W} - \mathbf{A}^T \boldsymbol{\theta} \mathbf{Y} + \mathbf{A}^T \boldsymbol{\theta} \mathbf{e} = \mathbf{0}. \quad (12)$$

将式 (11b) 代入 (11a), 得到

$$\lambda \mathbf{W} = \mathbf{A}^T \boldsymbol{\theta} \mathbf{e}. \quad (13)$$

然后将式 (13) 代入 (12), 得到

$$\mathbf{A}^T \boldsymbol{\theta} \mathbf{A} \mathbf{W} - \mathbf{A}^T \boldsymbol{\theta} \mathbf{Y} + \lambda \mathbf{W} = \mathbf{0}. \quad (14)$$

最终得到输出权值  $\mathbf{W}$  为

$$\mathbf{W} = (\mathbf{A}^T \boldsymbol{\theta} \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \boldsymbol{\theta} \mathbf{Y}. \quad (15)$$

比较式 (15) 与 (5) 容易看出, RBLS 与 BLS 的主要区别在于 RBLS 的解析表达式中新加入了一个加权矩阵  $\boldsymbol{\theta}$ , 其本质是根据样本的训练误差赋予每个样本不同的权值, 以反映其对于学习模型的不同贡献. 根据迭代加权最小二乘法<sup>[16]</sup>, 输出权值  $\mathbf{W}$  需要经过多轮迭代计算达到收敛. 具体地, 在每轮迭代学习过程中, 首先基于当前训练误差向量计算对应的加权矩阵  $\boldsymbol{\theta}$ , 然后采用式 (15) 对  $\mathbf{W}$  进行更新计算, 如此反复直至迭代结束. 详细执行过程见第 2.4 节算法描述.

## 2.3 鲁棒性分析

如前文所述, M-estimator 代价函数的引入及其相应的加权策略是 RBLS 具有良好鲁棒性的关键. 目前常用的 M-estimator 函数主要包括 Huber、Bisquare、Cauchy 等, 表 1 为这 3 种典型 M-estimator 所对应的代价函数、加权函数以及默认调节参数 tuning. 下面以 Huber 为例进行鲁棒性分析, 为便于描述, 图 1 为 Huber 的加权函数变化曲线 ( $k = 2$ ). 由图 1 和表 1 可以分析得到, 若训练误差  $e$  的绝对值小于等于阈值  $k$ , 表明当前训练样本为正常样本, 则此时 Huber 的代价函数等价于普通的 LS 代价函数, 且样本对应的权重为 1; 若训练误差  $e$  的绝对值大于阈值  $k$ , 表明当前训练样本为潜在的离群值, 则其对应的权重小于 1, 且权重的值随误差的增加而单调递减, 即误差越大, 权重越小, 从而能够对离群值所产生的较大扰动误差起到

表1 M-estimator函数

名称	代价函数 $\rho$	加权函数 $\theta$	默认 tuning
Huber	$\begin{cases} \frac{1}{2}e^2, &  e/k  \leq 1; \\ k e  - \frac{1}{2}k^2, &  e/k  > 1 \end{cases}$	$\begin{cases} 1, &  e/k  \leq 1; \\ \frac{k}{ e }, &  e/k  > 1 \end{cases}$	1.345
Bisquare	$\begin{cases} \frac{k^2}{6} \left(1 - \left(1 - \left(\frac{e}{k}\right)^2\right)^3\right), &  e/k  \leq 1; \\ \frac{k^2}{6}, &  e/k  > 1 \end{cases}$	$\begin{cases} \left(1 - \left(\frac{e}{k}\right)^2\right)^2, &  e/k  \leq 1 \\ 0, &  e/k  > 1 \end{cases}$	4.685
Cauchy	$\frac{k^2}{2} \log \left(1 + \left(\frac{e}{k}\right)^2\right)$	$\frac{1}{1 + \left(\frac{e}{k}\right)^2}$	2.385

很好的抑制作用,以降低学习模型对于离群值的敏感性.其他M-estimator函数的鲁棒性与Huber类似,只是在具体加权策略上有所不同,限于篇幅,不再赘述.

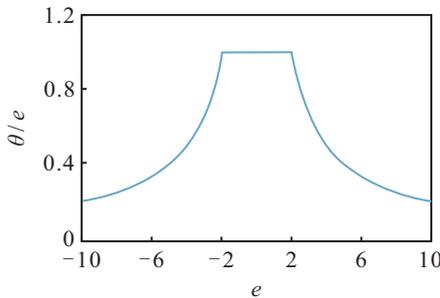


图1 Huber的加权函数曲线

就函数的几何意义而言,阈值  $k$  可理解为 M-estimator 函数对于训练误差的容忍程度,其值对于 RBLS 的鲁棒性和泛化性能具有重要的影响:较小的  $k$  值能够对离群值产生更强的抵抗能力,但同时也使得正常样本被误判为离群值的概率增加,导致模型的学习不够充分;较大的  $k$  值则增加了离群值被误判为正常样本的风险,影响建模的准确性.根据鲁棒回归理论<sup>[17]</sup>,阈值  $k$  通常采用下式进行计算:

$$k = \text{tuning} \times \sigma. \tag{16}$$

其中: tuning 为调节参数,其默认取值见表1;  $\sigma$  为训练误差的标准差,其在离群值环境下的一个鲁棒估计量为

$$\sigma = \text{MAR}/0.6745 = \text{med}(|e|)/0.6745. \tag{17}$$

其中: med( $\cdot$ ) 表示取中位数,常数 0.6745 的引入是为了保证估计量  $\sigma$  在高斯误差条件下的无偏性.

### 2.4 RBLS算法描述

综上所述, RBLS 的算法如下.

**算法1** 给定包含  $N$  个训练样本的数据集合  $(\mathbf{X}, \mathbf{Y})$ , 每个特征映射组中特征节点个数为  $k$ , 特征映射组数为  $n$ , 每个增强节点组中增强节点个数为  $q$ , 增强节点组数为  $m$ , 正则化参数为  $\lambda$ , 最大迭代步数为  $t_{\max}$ .

step 1: 随机生成特征映射层权值  $\mathbf{W}_{fi}$  和偏置  $\mathbf{b}_{fi}$ , 根据式(1)计算映射特征  $\mathbf{F}_i$ .

step 2: 设置特征映射组合  $\mathbf{F}^n \equiv [\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_n]$ .

step 3: 随机生成增强层权值  $\mathbf{W}_{ej}$  和偏置  $\mathbf{b}_{ej}$ , 根据式(2)计算增强节点  $\mathbf{E}_j$ .

step 4: 设置增强节点组合  $\mathbf{E}^m \equiv [\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_m]$ .

step 5: 设置  $\mathbf{A} = [\mathbf{F}^n | \mathbf{E}^m]$ , 根据式(5)计算初始输出权值  $\mathbf{W}$ .

step 6: 设置初始迭代步  $t = 1$ .

step 7: 计算当前迭代中所有训练样本的误差向量  $\mathbf{e} = \mathbf{Y} - \mathbf{AW}$ ,  $\mathbf{e} = [e_1, e_2, \dots, e_N]^T$ .

step 8: 根据式(17)计算训练误差  $\mathbf{e}$  的标准差  $\sigma$ .

step 9: 根据式(16)计算阈值  $k$ .

step 10: 根据式(10)所对应的加权函数计算每个样本的权重  $\theta_i = \theta(e_i/k)$ , 并得到加权矩阵  $\boldsymbol{\theta} = \text{diag}(\theta_1, \theta_2, \dots, \theta_N)$ .

step 11: 根据式(15)更新计算输出权值  $\mathbf{W}$ .

step 12: 设置  $t = t + 1$ , 若  $t > t_{\max}$ , 则算法终止; 否则返回至 step 7 继续执行.

## 3 实验与分析

为了验证 RBLS 算法的有效性, 选择 9 个不同规模大小的回归数据集和 2 个混沌时间序列数据集进行实验测试, 并将实验结果与原始 BLS<sup>[2]</sup> 以及最近提出的 FBLS<sup>[4]</sup> 进行比较和分析. 由于 RBLS 是一个基于 M-estimator 的统一学习框架, 本文使用 Huber、Bisquare 和 Cauchy 这 3 种典型的 M-estimator 函数进行实证研究, 相应的算法分别记作 Huber-RBLS、Bisquare-RBLS 和 Cauchy-RBLS.

### 3.1 回归数据集测试

#### 3.1.1 数据集与实验设计

实验所用的数据集是取自 UCI 数据库的 9 个真实的回归数据集, 如表 2 所示. 为了评估各种算法对

于离群值的鲁棒性,在各个数据集的训练集合中,分别随机选择10%、20%和30%的训练样本添加离群值,以构造出不同离群值水平的噪声数据集,离群值添加方式为在所选训练样本的目标输出 $y$ 上加入 $[y_{\min}, y_{\max}]$ 区间范围内的随机数。

表2 回归数据集描述

数据集	训练样本	测试样本	属性数目
Abalone	2 784	1 393	8
Basketball	64	32	4
Bodyfat	168	84	14
Cleveland	202	101	13
Housing	337	169	13
Mortgage	699	350	15
Pyrim	49	25	27
Strike	416	209	6
Weather Izmir	974	487	9

对于生成的具有不同离群值水平的各个数据集,分别采用BLS、FBLS、Huber-RBLS、Bisquare-RBLS以及Cauchy-RBLS这5种算法进行训练和测试。算法参数设置如下:对于BLS和所提出的Huber-RBLS、Bisquare-RBLS、Cauchy-RBLS这3种RBLS算法,需要设置的共同参数主要包括:每个特征映射组中特征节点个数 $k$ ,特征映射组数 $n$ ,每个增强节点组中增强节点个数 $q$ ,增强节点组数 $m$ ,正则化参数 $\lambda$ 。参照文献[3],首先设置 $m=1$ ,其他4个参数则采用网格搜索法确定,即使得算法在不同 $(k, n, q, \lambda)$ 参数组合下迭代运行,具有最小测试误差的最优参数组合将被选择,这里 $k$ 、 $n$ 、 $q$ 和 $\lambda$ 的搜索范围分别为 $\{2, 4, \dots, 18, 20\}$ 、 $\{2, 4, \dots, 18, 20\}$ 、 $\{10, 20, \dots, 190, 200\}$ 和 $\{10^{-10}, 10^{-9}, \dots, 10^4, 10^5\}$ 。此外,在3种RBLS算法中,最大迭代步数 $t_{\max}$ 统一设置为20。对于FBLS,参照文献[4],其主要参数也采用类似的网格搜索方法确定,其中每个模糊子系统中模糊规则个数 $N_r$ ,模糊子系统个数 $N_t$ ,增强节点个数 $N_e$ ,正则化参数 $\lambda$ 的搜索范围分别为 $\{1, 2, \dots, 19, 20\}$ 、 $\{1, 2, \dots, 9, 10\}$ 、 $\{1, 2, \dots, 99, 100\}$ 和 $\{10^{-10}, 10^{-9}, \dots, 10^4, 10^5\}$ 。

为保证实验结果的可靠性,对每个测试实例分别进行10次独立实验并对实验结果进行统计分析,算法性能评价标准采用均方根误差(RMSE)。

### 3.1.2 实验结果比较与分析

表3为各种算法在回归数据集上的训练时间和测试RMSE,其中最优的测试RMSE以粗体表示。对于参数组合列,BLS、Huber-RBLS、Bisquare-RBLS以及Cauchy-RBLS对应的是 $(k, n, q, \lambda)$ ,FBLS对应的是 $(N_r, N_t, N_e, \lambda)$ ,且其中的 $\lambda$ 值实际为正则化参数 $10^\lambda$ 。由表3可以得到如下结果。

1)当数据集不包含离群值( $P=0\%$ )时,在所有9个数据集上,最小的测试RMSE由FBLS和RBLS取得,其中FBLS共5次,RBLS共4次,而BLS为0次,这表明所提出的RBLS算法对于非离群值环境下的正常数据集的建模也是有效的。

2)当数据集包含离群值时,随着离群值水平的增加,BLS和FBLS的测试RMSE快速增加,而本文3种RBLS的测试RMSE仅有少量的增加,增幅明显更低,甚至在某些测试实例上基本保持不变,展现出了更好的抗离群值能力和鲁棒性。此外,对于不同离群值水平下的所有测试实例,除了 $P=10\%$ 下的Basketball数据集,最优的测试RMSE均由所提出RBLS取得,具有显著的优越性。

3)横向比较Huber-RBLS、Bisquare-RBLS以及Cauchy-RBLS这3种融合不同M-estimator函数的RBLS算法可以发现,它们在大部分测试实例中的学习性能基本相当,且分别能够在不同测试实例中取得最优的测试结果,其中Huber-RBLS共7次,Bisquare-RBLS共11次,Cauchy-RBLS共15次,但Huber-RBLS在部分测试实例中的测试结果不够稳定,尤其是当离群值水平较大时。因此,综合而言,Cauchy-RBLS的学习性能最优,Bisquare-RBLS次之,Huber-RBLS再次之。

4)关于训练时间,与原始BLS相比,尽管所提出RBLS由于在学习过程中需要迭代优化输出权值从而使得算法的训练时间有所增加,但其训练时间通常仅需零点几秒,仍然具有较高的学习效率。

综上所述,RBLS在保持BLS良好学习效率的同时,算法的鲁棒性得到了显著提升,且在离群值和非离群值环境下均可使用,是一种更加实用有效的鲁棒学习框架。

### 3.2 混沌时间序列预测

本节采用Mackey-Glass和Rossler这两个经典的混沌时间序列作为实验数据集,对BLS、FBLS以及RBLS的抗离群值能力和鲁棒性进行进一步的实验验证。为评估各种算法对于离群值的鲁棒性,在各时间序列的训练集中根据其规模大小添加了一定数量的离群值。离群值由随机数发生器随机生成,生成范围为训练集的最小值与最大值之间的区间。表4为混沌时间序列数据集描述,其中 $d, \tau$ 分别为对时间序列进行相空间重构的嵌入维数和时间延迟。需要说明的是,与第3.1节回归数据集中仅训练样本的目标输出包含离群值不同,这里的时间序列经过相空间重构后得到的训练集合中,训练样本的输入和输出均可能包含离群值。

表3 各种算法在回归数据集上的训练时间和测试RMSE

算法	P = 0%		P = 10%		P = 20%		P = 30%	
	参数组合	训练 测试 RMSE 时间 均值 ± 标准 (s) 差 / 10 <sup>-2</sup>	参数组合	训练 测试 RMSE 时间 均值 ± 标准 (s) 差 / 10 <sup>-2</sup>	参数组合	训练 测试 RMSE 时间 均值 ± 标准 (s) 差 / 10 <sup>-2</sup>	参数组合	训练 测试 RMSE 时间 均值 ± 标准 (s) 差 / 10 <sup>-2</sup>
Abalone								
BLS	(12, 2, 200, -2) 0.05	7.58 ± 0.02	(12, 2, 140, -1) 0.02	8.83 ± 0.03	(2, 8, 40, 4)	0.02 10.16 ± 0.15	(2, 20, 180, 5)	0.06 10.34 ± 0.08
FBLs	(2, 10, 78, -10) 0.17	<b>7.55 ± 0.02</b>	(19, 5, 3, -4)	0.22 8.80 ± 0.02	(20, 3, 9, 4)	0.13 9.52 ± 0.06	(19, 1, 37, 4)	0.05 9.47 ± 0.15
Huber-RBLS	(12, 2, 200, -2) 0.19	7.65 ± 0.02	(12, 2, 140, -2) 0.13	<b>7.61 ± 0.04</b>	(14, 2, 160, -1) 0.15	7.92 ± 0.02	(16, 2, 110, -1)	0.10 8.93 ± 0.04
Bisquare-RBLS	(12, 2, 200, -2) 0.21	7.83 ± 0.02	(12, 2, 90, -3)	0.08 7.77 ± 0.03	(12, 2, 90, -3)	0.08 <b>7.65 ± 0.04</b>	(12, 2, 140, -2)	0.13 <b>7.61 ± 0.03</b>
Cauchy-RBLS	(12, 2, 200, -2) 0.20	7.67 ± 0.02	(12, 2, 140, -2) 0.13	7.62 ± 0.03	(14, 2, 160, -1) 0.15	7.69 ± 0.02	(18, 2, 170, -1)	0.18 8.24 ± 0.04
Basketball								
BLS	(14, 4, 20, -1) 0.01	7.48 ± 0.28	(2, 6, 180, 1)	0.01 7.71 ± 0.10	(18, 20, 10, 3)	0.03 8.55 ± 0.11	(18, 14, 10, 3)	0.03 8.85 ± 0.19
FBLs	(1, 6, 1, -10) 0.02	<b>6.87 ± 0.05</b>	(1, 9, 1, -10)	0.03 <b>7.26 ± 0.11</b>	(1, 1, 9, -4)	0.01 10.34 ± 0.26	(1, 1, 57, -5)	0.01 12.22 ± 0.54
Huber-RBLS	(14, 4, 20, -1) 0.01	7.56 ± 0.29	(16, 16, 10, 0)	0.03 7.58 ± 0.26	(18, 20, 10, 2)	0.04 8.29 ± 0.06	(12, 18, 10, 2)	0.03 8.38 ± 0.09
Bisquare-RBLS	(16, 14, 10, -1) 0.03	7.63 ± 0.24	(16, 16, 10, 0)	0.03 7.63 ± 0.20	(6, 4, 10, -1)	0.01 <b>7.73 ± 0.15</b>	(4, 4, 40, 0)	0.01 <b>7.54 ± 0.25</b>
Cauchy-RBLS	(14, 4, 20, -1) 0.01	7.60 ± 0.30	(16, 16, 10, 0)	0.03 7.55 ± 0.23	(8, 14, 10, 0)	0.03 8.17 ± 0.24	(18, 18, 10, 2)	0.03 8.26 ± 0.08
Bodyfat								
BLS	(4, 6, 10, -10) 0.01	0.47 ± 0.04	(20, 4, 10, 0)	0.01 4.93 ± 0.28	(6, 20, 20, 1)	0.03 8.75 ± 0.39	(18, 18, 10, 1)	0.03 13.47 ± 0.15
FBLs	(20, 8, 79, -4) 0.07	0.46 ± 0.05	(7, 10, 86, -1)	0.06 4.93 ± 0.19	(2, 10, 56, -1)	0.04 8.84 ± 0.65	(1, 10, 76, -2)	0.03 12.99 ± 0.40
Huber-RBLS	(4, 6, 40, -7) 0.02	0.19 ± 0.03	(4, 6, 10, -8)	0.01 0.32 ± 0.05	(4, 6, 10, -5)	0.01 0.38 ± 0.06	(6, 20, 10, -1)	0.04 1.68 ± 0.28
Bisquare-RBLS	(4, 6, 40, -7) 0.02	0.20 ± 0.03	(4, 8, 40, -5)	0.02 0.24 ± 0.06	(4, 6, 30, -9)	0.01 <b>0.23 ± 0.06</b>	(10, 16, 10, -4)	0.04 0.40 ± 0.06
Cauchy-RBLS	(4, 4, 50, -10) 0.01	<b>0.18 ± 0.02</b>	(4, 8, 40, -6)	0.02 <b>0.21 ± 0.04</b>	(4, 6, 30, -10)	0.01 <b>0.23 ± 0.06</b>	(10, 16, 10, -5)	0.04 <b>0.39 ± 0.06</b>
Cleveland								
BLS	(4, 14, 190, 2) 0.03	12.11 ± 0.06	(4, 14, 190, 2)	0.03 12.41 ± 0.10	(4, 14, 20, 3)	0.02 12.66 ± 0.15	(2, 6, 40, 3)	0.01 12.67 ± 0.09
FBLs	(3, 7, 73, -1) 0.04	<b>12.06 ± 0.27</b>	(3, 2, 72, -2)	0.01 12.38 ± 0.34	(16, 3, 4, 2)	0.03 12.70 ± 0.11	(20, 3, 2, 2)	0.03 12.64 ± 0.15
Huber-RBLS	(2, 10, 50, -1) 0.02	12.21 ± 0.16	(4, 12, 90, 1)	0.03 12.06 ± 0.11	(4, 12, 90, 1)	0.03 12.41 ± 0.10	(4, 20, 40, 3)	0.04 12.63 ± 0.10
Bisquare-RBLS	(4, 16, 90, 1) 0.04	12.33 ± 0.13	(4, 12, 180, 1)	0.04 12.08 ± 0.13	(4, 16, 140, 1)	0.04 <b>12.27 ± 0.18</b>	(4, 4, 120, 3)	0.01 12.63 ± 0.20
Cauchy-RBLS	(4, 16, 90, 1) 0.03	12.22 ± 0.12	(4, 12, 180, 1)	0.04 <b>12.00 ± 0.13</b>	(4, 12, 90, 1)	0.03 12.36 ± 0.09	(4, 4, 120, 3)	0.01 <b>12.61 ± 0.20</b>
Housing								
BLS	(6, 12, 190, -1) 0.02	8.42 ± 0.24	(20, 2, 200, 0)	0.01 10.21 ± 0.39	(20, 2, 140, 1)	0.01 14.21 ± 0.30	(16, 10, 190, 2)	0.02 16.59 ± 0.31
FBLs	(19, 9, 56, -3) 0.09	<b>8.02 ± 0.33</b>	(1, 10, 67, -7)	0.04 10.38 ± 0.61	(2, 9, 62, -4)	0.05 13.97 ± 0.55	(3, 9, 31, -3)	0.06 16.46 ± 0.51
Huber-RBLS	(8, 8, 150, -2) 0.05	8.73 ± 0.52	(12, 2, 200, -2) 0.05	<b>9.23 ± 0.69</b>	(20, 12, 80, 1)	0.08 10.80 ± 0.42	(18, 8, 200, 1)	0.10 11.38 ± 0.48
Bisquare-RBLS	(8, 8, 130, -3) 0.05	10.30 ± 0.85	(8, 4, 180, -2)	0.05 10.41 ± 0.71	(14, 18, 200, 1)	0.14 10.64 ± 0.41	(10, 10, 190, -1)	0.08 <b>10.27 ± 0.76</b>
Cauchy-RBLS	(8, 8, 150, -2) 0.05	9.58 ± 0.78	(8, 8, 150, -2)	0.05 9.74 ± 0.95	(18, 6, 120, 1)	0.05 <b>10.50 ± 0.35</b>	(10, 4, 140, -1)	0.03 10.48 ± 0.91
Mortgage								
BLS	(16, 6, 200, -3) 0.02	0.50 ± 0.02	(6, 10, 10, 1)	0.02 5.22 ± 0.06	(20, 18, 10, 3)	0.05 8.27 ± 0.06	(20, 4, 10, 3)	0.01 12.05 ± 0.16
FBLs	(19, 8, 62, -6) 0.14	0.52 ± 0.02	(2, 10, 17, -2)	0.07 5.26 ± 0.09	(1, 7, 100, -2)	0.04 8.43 ± 0.08	(2, 2, 1, 2)	0.02 10.33 ± 1.29
Huber-RBLS	(18, 6, 170, -3) 0.11	<b>0.49 ± 0.04</b>	(14, 2, 140, -4) 0.04	0.55 ± 0.04	(20, 2, 180, -3) 0.07	0.59 ± 0.02	(10, 4, 140, -1)	0.05 1.06 ± 0.04
Bisquare-RBLS	(18, 6, 170, -3) 0.12	0.53 ± 0.04	(16, 2, 200, -4) 0.08	0.57 ± 0.07	(20, 2, 180, -4) 0.08	0.54 ± 0.03	(6, 10, 160, -4)	0.08 0.63 ± 0.03
Cauchy-RBLS	(18, 6, 170, -3) 0.11	<b>0.49 ± 0.04</b>	(20, 2, 180, -4) 0.07	<b>0.49 ± 0.04</b>	(16, 6, 200, -3) 0.11	<b>0.51 ± 0.04</b>	(18, 6, 170, -3)	0.12 <b>0.56 ± 0.04</b>
Pyrim								
BLS	(2, 10, 70, -3) 0.02	9.24 ± 2.12	(14, 18, 180, 3)	0.03 13.17 ± 0.37	(16, 4, 150, 3)	0.01 12.94 ± 0.53	(14, 2, 120, 3)	0.01 13.27 ± 0.43
FBLs	(3, 4, 84, -5) 0.02	<b>6.99 ± 1.23</b>	(1, 6, 93, -7)	0.02 12.19 ± 2.50	(15, 10, 5, 2)	0.06 15.56 ± 0.38	(12, 9, 5, 2)	0.05 15.33 ± 0.51
Huber-RBLS	(2, 10, 70, -4) 0.02	9.31 ± 2.20	(18, 2, 10, -3)	0.01 11.36 ± 2.28	(8, 6, 130, 2)	0.01 12.65 ± 0.64	(14, 18, 180, 3)	0.04 13.03 ± 0.31
Bisquare-RBLS	(2, 12, 10, -5) 0.02	9.90 ± 2.70	(4, 6, 180, 0)	0.01 11.56 ± 0.72	(4, 8, 190, 0)	0.02 <b>11.41 ± 0.55</b>	(4, 4, 170, 0)	0.01 <b>11.40 ± 0.77</b>
Cauchy-RBLS	(2, 10, 70, -9) 0.02	9.66 ± 1.86	(18, 2, 10, -3)	0.01 <b>10.56 ± 2.84</b>	(16, 4, 100, 0)	0.01 11.77 ± 0.57	(4, 14, 190, 0)	0.03 11.50 ± 1.43
Strike								
BLS	(4, 12, 200, 0) 0.02	10.13 ± 0.06	(4, 18, 30, 4)	0.03 10.62 ± 0.01	(2, 4, 50, 4)	0.01 10.60 ± 0.03	(4, 12, 200, 5)	0.02 10.61 ± 0.02
FBLs	(5, 7, 79, -7) 0.05	10.16 ± 0.09	(8, 1, 14, 3)	0.01 10.83 ± 0.04	(14, 2, 94, 4)	0.02 10.83 ± 0.01	(4, 1, 4, 3)	0.01 10.83 ± 0.05
Huber-RBLS	(4, 14, 180, -3) 0.06	<b>9.99 ± 0.08</b>	(4, 10, 120, -2) 0.04	<b>10.01 ± 0.07</b>	(4, 14, 200, -1) 0.07	<b>9.97 ± 0.04</b>	(6, 16, 190, 1)	0.08 10.31 ± 0.08
Bisquare-RBLS	(4, 8, 200, -4) 0.06	10.08 ± 0.13	(6, 18, 190, -2) 0.09	10.11 ± 0.12	(4, 14, 200, -3) 0.08	10.11 ± 0.16	(4, 12, 130, -2)	0.06 10.08 ± 0.06
Cauchy-RBLS	(4, 14, 200, -3) 0.08	10.03 ± 0.09	(4, 12, 150, -3) 0.06	10.02 ± 0.11	(6, 14, 190, -2) 0.09	10.00 ± 0.06	(4, 12, 200, -1)	0.07 <b>9.96 ± 0.06</b>
Weather Izmir								
BLS	(6, 2, 160, -3) 0.01	1.93 ± 0.02	(8, 6, 180, 2)	0.02 5.51 ± 0.07	(6, 18, 200, 3)	0.04 9.81 ± 0.07	(4, 14, 190, 4)	0.03 12.35 ± 0.63
FBLs	(20, 9, 87, -4) 0.17	1.93 ± 0.01	(3, 10, 2, -1) 0.10	5.49 ± 0.08	(8, 7, 3, 3)	0.10 9.28 ± 0.31	(3, 8, 1, 3)	0.08 9.59 ± 0.45
Huber-RBLS	(6, 4, 200, -2) 0.08	<b>1.92 ± 0.02</b>	(8, 6, 190, -1) 0.08	1.94 ± 0.02	(6, 4, 200, -1) 0.08	2.12 ± 0.03	(8, 2, 180, -1)	0.08 2.54 ± 0.06
Bisquare-RBLS	(8, 4, 150, -2) 0.06	1.93 ± 0.02	(6, 4, 200, -2) 0.08	1.94 ± 0.02	(6, 4, 200, -2) 0.09	<b>1.94 ± 0.02</b>	(6, 2, 130, -3)	0.05 <b>1.93 ± 0.02</b>
Cauchy-RBLS	(6, 4, 200, -2) 0.08	<b>1.92 ± 0.02</b>	(6, 4, 200, -2) 0.09	<b>1.92 ± 0.02</b>	(8, 2, 160, -2) 0.06	1.95 ± 0.02	(8, 2, 160, -2)	0.06 1.95 ± 0.02

表4 混沌时间序列数据集描述

时间序列	训练样本	测试样本	离群值个数	$(d, \tau)$
Mackey-Glass	1 000	500	8	(17, 1)
Rosler	1 500	500	13	(1, 5)

对于上述包含离群值的时间序列数据集, 分别使用BLS、FBLS和RBLS算法对训练样本进行学习建模, 并在测试集上对其预测性能进行比较. 各种算法的实验参数设置与第3.1节完全相同, 其最终得到的最优参数组合和所对应的训练时间的平均值、预测RMSE的平均值和标准差如表5所示. 由表5可见, 对于包含离群值的混沌时间序列数据集, 原始BLS的预测结果最差, 采用模糊方法的FBLS的预测结果稍好, 所提出的3种RBLS取得了最好的预测性能, 且与前面两种对比方法相比优势非常明显, 展现出更强的抗离群值能力和鲁棒性. 此外, 本节实验结果表明, 对于训练样本的输入和输出均包含离群值的情形, 所提出的RBLS算法也是适用有效的.

表5 各种算法在时间序列数据集上的训练时间和测试RMSE

时间序列	算法	参数组合	训练时间 (s)	测试RMSE 均值 ± 标准差 ( $10^{-3}$ )
Mackey-Glass	BLS	(12, 2, 90, -4)	0.01	7.71 ± 0.64
	FBLS	(20, 10, 90, -4)	0.24	7.12 ± 0.33
	Huber-RBLS	(16, 2, 190, -10)	0.09	1.21 ± 0.19
	Bisquare-RBLS	(12, 14, 30, -7)	0.09	<b>0.69 ± 0.21</b>
	Cauchy-RBLS	(8, 8, 190, -5)	0.11	0.77 ± 0.11
Rosler	BLS	(10, 18, 10, 0)	0.04	6.46 ± 1.56
	FBLS	(1, 9, 98, -6)	0.04	5.23 ± 0.36
	Huber-RBLS	(10, 18, 10, -1)	0.06	<b>2.76 ± 0.15</b>
	Bisquare-RBLS	(12, 18, 10, 0)	0.07	2.79 ± 0.14
	Cauchy-RBLS	(14, 20, 10, -1)	0.10	2.78 ± 0.32

### 3.3 收敛性分析

如第2.4节所述, RBLS在学习过程中需要多次迭代计算以得到优化的输出权值. 为了直观分析RBLS在迭代学习过程中的收敛特性, 以Abalone、Bodyfat、Mortgage、Strike这4个回归数据集为例(离群值水平 $P=20\%$ ), 给出了3种RBLS算法的收敛, 分别如图2~图5所示. 由图2~图5可见, 在所有4个数据集上, 3种RBLS算法的测试RMSE在经过较少的迭代步数后均能够快速收敛至一个稳定值, 收敛速度快, 收敛结果稳定, 具有良好的收敛性.

与原始BLS相比, RBLS算法仅新增了最大迭代步数 $t_{max}$ 这一个参数. 而根据上述收敛性分析可知, RBLS的学习性能对于 $t_{max}$ 并不敏感, 即 $t_{max}$ 可以在一个较为宽松的范围内进行取值而不会对RBLS的学习性能造成影响, 如在本文的所有测试实例中, RBLS中的 $t_{max}$ 统一设置为20, 并均取得了良好的

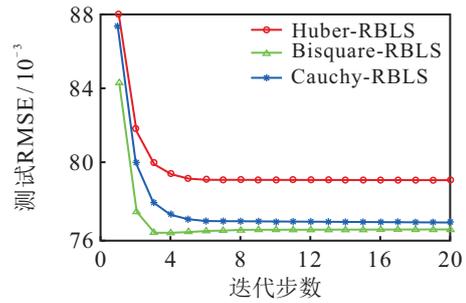


图2 RBLS在Abalone上的收敛

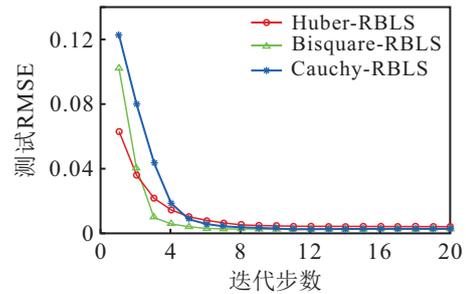


图3 RBLS在Bodyfat上的收敛

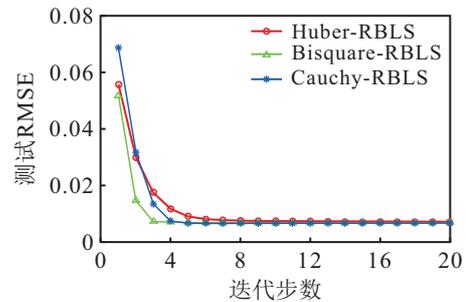


图4 RBLS在Mortgage上的收敛

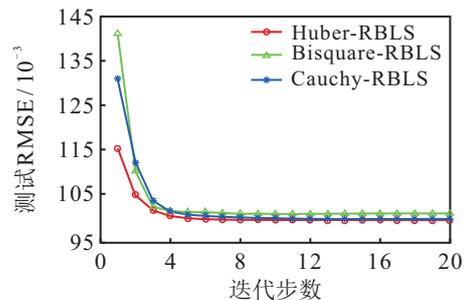


图5 RBLS在Strike上的收敛

测试结果. 事实上, 为了提升RBLS的学习效率, 将 $t_{max}$ 设置为一个较小的值也是可行的. 因此从收敛性分析的角度, RBLS中新引入的参数并未增加算法的使用难度.

## 4 结论

为解决离群值环境下的数据建模问题, 本文提出了一种基于M-estimator的鲁棒宽度学习系统(RBLS). RBLS以BLS模型为基础, 通过引入具有鲁棒特性的M-estimator函数作为代价函数, 并采用迭代加权方法对学习样本进行逆向加权计算, 以对离群值样本所产生的扰动误差进行反向抑制, 从而有效

降低了学习模型对于离群值的敏感性. 9个回归数据集和2个时间序列数据集上的建模实验表明,在多种离群值水平和不同离群值类型下,融合使用3种不同M-estimator函数的RBLS算法均能够取得较BLS更优的学习性能,展现出更强的离群值抵抗能力. 此外,对于不包括离群值的干净数据集, RBLS同样能够取得与BLS相当或更好的测试精度,具有广泛的适用性. 本文主要研究了离线批处理模式下含离群值数据集的学习问题,并提出了相应的鲁棒学习框架和具体实现算法,如何将其拓展至在线学习场景将是下一步的研究工作.

### 参考文献(References)

- [1] Sengupta S, Basak S, Saikia P, et al. A review of deep learning with special emphasis on architectures, applications and recent trends[J]. Knowledge-Based Systems, 2020, 194: 105596.
- [2] Chen C L P, Liu Z L. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(1): 10-24.
- [3] Chen C L P, Liu Z L, Feng S. Universal approximation capability of broad learning system and its structural variations[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(4): 1191-1204.
- [4] Feng S, Chen C L P. Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification[J]. IEEE Transactions on Cybernetics, 2020, 50(2): 414-424.
- [5] Guo H B, Sheng B, Li P, et al. Multiview high dynamic range image synthesis using fuzzy broad learning system[J]. IEEE Transactions on Cybernetics, 2021, 51(5): 2735-2747.
- [6] Zhao H M, Zheng J J, Deng W, et al. Semi-supervised broad learning system based on manifold regularization and broad network[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2020, 67(3): 983-994.
- [7] 徐鹏飞, 王敏, 刘金平, 等. 基于数据分布特性的代价敏感宽度学习系统[J]. 控制与决策, 2021, 36(7): 1686-1692.  
(Xu P F, Wang M, Liu J P, et al. Data distribution-based cost-sensitive broad learning system[J]. Control and Decision, 2021, 36(7): 1686-1692.)
- [8] Liu Z L, Chen C L P, Feng S, et al. Stacked broad learning system: From incremental flattened structure to deep model[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(1): 209-222.
- [9] Wang B S, Zhao Y, Chen C L P. Hybrid transfer learning and broad learning system for wearing mask detection in the COVID-19 era[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-12.
- [10] Sheng B, Li P, Zhang Y H, et al. GreenSea: Visual soccer analysis using broad learning system[J]. IEEE Transactions on Cybernetics, 2021, 51(3): 1463-1477.
- [11] Chu F, Liang T, Chen C L P, et al. Weighted broad learning system and its application in nonlinear industrial process modeling[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(8): 3017-3031.
- [12] Du J, Vong C M, Chen C L P. Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification[J]. IEEE Transactions on Cybernetics, 2021, 51(3): 1586-1597.
- [13] Xu M L, Han M, Chen C L P, et al. Recurrent broad learning systems for time series prediction[J]. IEEE Transactions on Cybernetics, 2020, 50(4): 1405-1417.
- [14] Han M, Feng S B, Chen C L P, et al. Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(9): 1809-1821.
- [15] Han M, Li W J, Feng S B, et al. Maximum information exploitation using broad learning system for large-scale chaotic time-series prediction[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(6): 2320-2329.
- [16] Beck A. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes[J]. SIAM Journal on Optimization, 2015, 25(1): 185-209.
- [17] Rousseeuw P J, Leroy A M. Robust regression and outlier detection[M]. New York: Wiley, 2005: 43-44.

### 作者简介

郭威(1983—),男,副教授,博士,从事数据挖掘、机器学习等研究, E-mail: weiguo031@163.com;

徐涛(1962—),男,教授,博士生导师,从事数据挖掘、智能信息处理等研究, E-mail: xutao@nuaa.edu.cn.

(责任编辑:魏冰)