

加权光滑投影孪生支持向量回归算法

徐奔业, 顾斌杰, 潘 丰, 熊伟丽

(江南大学 物联网工程学院, 江苏 无锡 214122)

摘 要: 现有双边移位投影孪生支持向量回归(PPTSVR)算法在训练阶段没有考虑不同位置样本对超平面构造的影响, 当样本中存在异常点时会降低算法拟合性能。针对该问题, 提出一种加权光滑投影孪生支持向量回归算法。采用孤立森林法赋予每个样本不同的权值, 并且赋予样本中异常点很小的权值, 通过将权值引入算法目标函数, 削弱异常点对超平面构造的影响。为直接在原空间中寻求最优超平面, 引入正号函数, 将有约束优化问题转化为无约束优化问题, 并采用 Sigmoid 光滑函数对目标函数进行光滑处理, 证明其任意阶可微且严格凸的特性, 进而原空间中采用牛顿迭代法进行求解。在基准数据集和人工测试函数上的实验结果表明, 该算法相比于现有代表性回归算法具备更好的拟合性能和更快的训练速度, 尤其当训练样本中存在异常点时, 相比于 PPTSVR 算法拟合性能提升更明显。

关键词: 投影; 孪生支持向量回归; 孤立森林; Sigmoid 光滑函数; 牛顿迭代法

开放科学(资源服务)标志码(OSID):



中文引用格式: 徐奔业, 顾斌杰, 潘丰, 等. 加权光滑投影孪生支持向量回归算法[J]. 计算机工程, 2022, 48(12): 104-111, 118.

英文引用格式: XU B Y, GU B J, PAN F, et al. Weighted smooth projection twin support vector regression algorithm[J]. Computer Engineering, 2022, 48(12): 104-111, 118.

Weighted Smooth Projection Twin Support Vector Regression Algorithm

XU Benye, GU Binjie, PAN Feng, XIONG Weili

(School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China)

[Abstract] The existing Pair-shifted Projection Twin Support Vector Regression (PPTSVR) algorithm ignores the effects of samples at different locations on the hyperplane construction during the training process. If there are outliers in the samples, the fitting capacity of the algorithm will be weakened. Therefore, this study proposes a Weighted Smooth Projection Twin Support Vector Regression (WSPTSVR) algorithm. First, an isolation forest approach is utilized to assign different weights to each sample, and the effects of outliers on the hyperplane construction are weakened by assigning tiny weights to them. Second, to find the optimal hyperplane directly in the original space, the algorithm adopts a plus function to convert the constrained optimization problems into unconstrained ones, and utilizes a Sigmoid smooth function to smooth the objective function. It is proved that the objective function is differentiable and strictly convex at any order; then, a Newton iteration method is employed to solve the unconstrained optimization problems in the primal space. Finally, the effectiveness of the proposed algorithm is validated on benchmark datasets and an artificial test function. The experimental results show that the WSPTSVR algorithm outperforms several state-of-the-art algorithms in terms of fitting capacity and training speed. Especially, if there are outliers in the training samples, the fitting capacity of the proposed algorithm is greatly improved compared to that of the PPTSVR algorithm.

[Key words] projection; Twin Support Vector Regression (TSVR); isolated forest; Sigmoid smooth function; Newton iteration method

DOI: 10.19678/j.issn.1000-3428.0063542

0 概述

支持向量机(Support Vector Machine, SVM)是20世纪90年代由VAPNIK^[1-2]提出的一种机器学习算法。与传统的以降低经验风险为目标的神经网络相比, SVM

的主要思想是最小化结构风险, 这使得SVM具有良好的泛化性能^[3-4]。目前, SVM在入侵检测^[5]、图像处理^[6]、故障诊断^[7]、干扰识别^[8]等领域得到广泛应用。然而, SVM在训练大规模数据集时, 存在时间复杂度高导致训练速度慢的问题^[9]。KHEMCHANDANI等^[10]提出

基金项目: 国家自然科学基金(61773182)。

作者简介: 徐奔业(1996—), 男, 硕士研究生, 主研方向为机器学习、模式识别; 顾斌杰(通信作者), 副教授、博士; 潘 丰、熊伟丽, 教授、博士。

收稿日期: 2021-12-15 修回日期: 2022-01-24 E-mail: gubinjie1980@126.com

孪生支持向量机(Twin Support Vector Machine, TSVM)。与SVM相比, TSVM仅需求解两个较小规模的二次规划问题, 因此训练时间仅为SVM的1/4左右。CHEN等^[11]提出投影孪生支持向量机(Projection Twin Support Vector Machine, PTSVM)。PTSVM通过为每个类寻找一个最优投影轴使投影点类内方差最小化, 构建分类模型。

PENG^[12]把TSVM的思想用于回归领域, 提出孪生支持向量回归(Twin Support Vector Regression, TSVR)算法。TSVR通过求解两个较小规模的二次规划问题寻求回归函数, 与传统SVR相比, TSVR具有更好的泛化性能和更快的训练速度。CHEN等^[13]引入Sigmoid光滑函数, 对TSVR的目标函数进行光滑处理, 提出光滑孪生支持向量回归(Smooth Twin Support Vector Regression, STSVR)算法。STSVR通过求解一对无约束优化问题, 获得了比TSVR更快的训练速度。PENG等^[14]基于PTSVM的思想, 提出投影孪生支持向量回归(Projection Twin Support Vector Regression, PTSVR)算法, 包括双边移位投影孪生支持向量回归(Pair-shifted PTSVR, PPTSVR)算法和单边移位投影孪生支持向量回归(Single-shifted PTSVR, SPTSVR)算法。PTSVR将训练集中的每个训练点进行上下移位得到两个新的移位集, 从而利用PTSVM的思想求解两个最优超平面的法向量。PPTSVR和SPTSVR的主要区别在于, PPTSVR在两个移位集上构建回归函数, 而SPTSVR则是在原始集和一个移位集上构建回归函数。实验结果表明, PTSVR有着比TSVR更好的预测性能。

然而, PPTSVR在寻找最优超平面的过程中, 将所有训练样本对超平面的作用视为相同, 没有反映数据在空间中的分布情况, 当训练样本中存在异常点时, 会削弱算法的拟合性能。本文采用孤立森林法赋予每个训练样本不同的权值, 通过赋予潜在的异常点很小的权值, 削弱异常点对超平面构造的影响, 从而提高算法的拟合性能。同时, 为了避免在对偶空间中求解二次规划问题, 引入正号函数, 将有约束优化问题转化为不光滑的无约束优化问题, 并采用Sigmoid光滑函数进行光滑处理, 提出一种加权光滑投影孪生支持向量回归(Weighted Smooth Projection Twin Support Vector Regression, WSPTSVR)算法, 以证明其任意阶光滑且全局收敛的特性, 并采用牛顿迭代法进行求解。

1 WSPTSVR算法

本节首先采用孤立森林法判断样本中的潜在异常点, 并通过异常分数机制赋予每个样本不同的权值; 其次引入Sigmoid光滑函数, 提出WSPTSVR算

法, 并证明其具有全局唯一最优解; 最后在原空间中使用牛顿迭代法进行求解。

1.1 基于孤立森林法的加权系数确定

孤立森林法能够有效地检测出样本中的潜在异常点, 即使样本中没有异常点, 也能够根据异常分数赋予每个样本相应的权值, 从而提高算法的拟合性能^[15-17]。孤立森林法首先递归地分割给定样本集, 直到每个样本都被单独分离出来, 然后根据每个样本分离的路径长度计算其异常分数, 根据异常分数大小判断样本是否为异常点并赋予每个样本点相应的权值。通过孤立森林法确定加权系数共分为以下3个步骤:

1) 通过训练样本的子集构建 t 棵孤立树, 建立孤立森林。

2) 在训练后的孤立森林中, 根据每个样本点分离所需的路径长度赋予其相应的异常分数 S_i , S_i 的取值范围为0~1, 且取值越大, 该点为异常点的可能性越大, 异常分数 S_i 计算如下:

$$S_i = 2^{-\frac{E(h(i))}{c(n)}} \quad (1)$$

其中: $h(i)$ 为样本 i 的路径长度, 即从根节点到叶子节点的边数, 而异常点更容易被孤立, 因此 $h(i)$ 较小; $E(h(i))$ 为样本 i 在一组孤立森林中路径长度的均值; $c(n)$ 是样本数为 n 时路径长度的均值, 用来标准化样本 i 的路径长度 $h(i)$ 。 $c(n)$ 计算如下:

$$c(n) = 2H(n-1) - 2(n-1)/n \quad (2)$$

其中: $H(n-1) = \ln(n-1) + 0.577\ 215\ 664\ 9$ 为调和数。

3) 通过孤立森林法中的异常分数机制为每个样本点赋予相应的权值:

$$\rho_i = 1 - S_i, i = 1, 2, \dots, n \quad (3)$$

其中: ρ_i 表示第 i 个样本的权值。

文献[15]的研究表明, 当样本点的异常分数 $S_i > 0.60$ 时, 即视该样本点为潜在异常点, 应赋予很小的权值, 则样本点的权值矩阵可表示为对角矩阵:

$$W_{ij} = \begin{cases} \rho_i, & \text{当 } i=j \text{ 且 } S_i \leq 0.6 \\ 10^{-5}, & \text{当 } i=j \text{ 且 } S_i > 0.6 \\ 0, & \text{其他} \end{cases} \quad (4)$$

1.2 线性情况

假设训练集用 $D = \{(\mathbf{x}_i; \mathbf{y}_i), i \in I = \{1, 2, \dots, n\}\}$ 表示, 其中, $\mathbf{x}_i \in R^m$ 为输入, $\mathbf{y}_i \in R$ 为输出, I 为指标集, n 为样本个数, m 为样本特征数。为了后续表示简单起见, 分别用矩阵 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T \in R^{n \times m}$ 和向量 $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]^T \in R^n$ 表示输入和输出, 用矩阵 $\mathbf{J} = [\mathbf{X}, \mathbf{Y}] \in R^{n \times (m+1)}$ 表示训练样本, 分别用矩阵 $\mathbf{A}_e =$

$[X, Y + \varepsilon e]$ 和 $B_e = [X, Y - \varepsilon e] \in R^{n \times (m+1)}$ 表示上移和下移训练样本, 其中 $\varepsilon > 0$ 为不敏感因子, e 为全1的列向量。

$$\min_{\omega_1, \delta_1, \xi} \frac{C_3}{2} (\|\omega_1\|_2^2 + \delta_1^2) + \frac{1}{2} (E[\omega_1^T \delta_1]^T)^T W_{ij} (E[\omega_1^T \delta_1]^T) + C_1 e^T \xi, \text{ s.t. } F[\omega_1^T \delta_1]^T \leq -e + \xi, \xi \geq 0 \quad (5)$$

$$\min_{\omega_2, \delta_2, \eta} \frac{C_4}{2} (\|\omega_2\|_2^2 + \delta_2^2) + \frac{1}{2} (E[\omega_2^T \delta_2]^T)^T W_{ij} (E[\omega_2^T \delta_2]^T) + C_2 e^T \eta, \text{ s.t. } G[\omega_2^T \delta_2]^T \geq e - \eta, \eta \geq 0 \quad (6)$$

其中: $[\omega_1^T \delta_1]$ 和 $[\omega_2^T \delta_2] \in R^{m+1}$ 为两个超平面的法向量; $C_1, C_2 > 0$ 为惩罚参数; $C_3, C_4 > 0$ 为正则化参数; $\xi, \eta \geq 0$ 为不敏感损失参数; $E = J - ee^T J/n$; $F = B_e - ee^T A_e/n$; $G = A_e - ee^T B_e/n$; $\|\cdot\|_2$ 表示 L_2 范数。

然而, 式(5)和式(6)需要在对偶空间中求解二次规划问题, 训练速度较慢。为了加快训练速度, 采

$$\min_{\omega_1, \delta_1} \frac{C_3}{2} (\|\omega_1\|_2^2 + \delta_1^2) + \frac{1}{2} (E[\omega_1^T \delta_1]^T)^T W_{ij} (E[\omega_1^T \delta_1]^T) + C_1 e^T \max\{0, e + F[\omega_1^T \delta_1]^T\} \quad (9)$$

$$\min_{\omega_2, \delta_2} \frac{C_4}{2} (\|\omega_2\|_2^2 + \delta_2^2) + \frac{1}{2} (E[\omega_2^T \delta_2]^T)^T W_{ij} (E[\omega_2^T \delta_2]^T) + C_2 e^T \max\{0, e - G[\omega_2^T \delta_2]^T\} \quad (10)$$

定理1^[18] 无约束优化问题式(9)和式(10)连续但不光滑。

证明 分别令正号函数 $(x_1)_+ = \max\{0, e + F[\omega_1^T \delta_1]^T\}$ 和 $(x_2)_+ = \max\{0, e - G[\omega_2^T \delta_2]^T\}$ 。显然, 式(9)和式(10)的可微性和光滑性取决于正号函数 $(x_i)_+ (i=1, 2)$ 。由于 $(x_i)_+ = \begin{cases} x_i, & x_i \geq 0 \\ 0, & x_i < 0 \end{cases}$, 可得 $(x_i)_+$ 在 $x_i = 0$ 处的左右导数分别为0和1, 因此 $(x_i)_+$ 在 $x_i = 0$ 处不可微。又由于 $(x_i)_+$ 还可以表示为 $(x_i)_+ = \frac{|(x_i)_+| + (x_i)_+}{2}$, 因此 $(x_i)_+$ 是连续函数。所以, 正号函数 $(x_i)_+$ 连续但不光滑。

定理1表明式(9)和式(10)不光滑, 因此无法使

$$\min_{\omega_1, \delta_1} P_1 = \frac{C_3}{2} (\|\omega_1\|_2^2 + \delta_1^2) + \frac{1}{2} (E[\omega_1^T \delta_1]^T)^T W_{ij} (E[\omega_1^T \delta_1]^T) + C_1 e^T p(e + F[\omega_1^T \delta_1]^T, \alpha) \quad (14)$$

$$\min_{\omega_2, \delta_2} P_2 = \frac{C_4}{2} (\|\omega_2\|_2^2 + \delta_2^2) + \frac{1}{2} (E[\omega_2^T \delta_2]^T)^T W_{ij} (E[\omega_2^T \delta_2]^T) + C_2 e^T p(e - G[\omega_2^T \delta_2]^T, \alpha) \quad (15)$$

引理1^[19] 若矩阵 $A \in R^{n \times n}$ 是实对称矩阵, 则 A 是正定矩阵当且仅当存在矩阵 $B \in R^{m \times n}$ 使得 $A = B^T B$ 。

定理2 式(14)中的 P_1 和式(15)中的 P_2 是严格凸函数。

证明 对任意的正常数 α , P_1 显然是连续可微的, 以下证明其是严格凸函数:

令 $u_1 = [\omega_1^T \delta_1]^T$, 则由式(14)可得:

$$\nabla P_1(u_1) = (C_3 I + E^T W_{ij} E) u_1 + C_1 F^T \frac{1}{1 + \exp(-\alpha(Fu_1 + e))} \quad (16)$$

$$\nabla^2 P_1(u_1) = C_3 I + E^T W_{ij} E +$$

$$\alpha C_1 F^T \text{diag} \left(\frac{\exp(-\alpha(Fu_1 + e))}{(1 + \exp(-\alpha(Fu_1 + e)))^2} \right) F \quad (17)$$

在双边移位投影孪生支持向量回归算法的目标函数中引入权值矩阵 W_{ij} , 构造如下最优化问题:

用正号函数将其转化为无约束优化问题, 在原空间中直接进行求解。

由 Karush-Kuhn-Tucker(KKT) 条件可得:

$$\xi = \max\{0, e + F[\omega_1^T \delta_1]^T\} \quad (7)$$

$$\eta = \max\{0, e - G[\omega_2^T \delta_2]^T\} \quad (8)$$

式(5)和式(6)可改写为如下无约束优化问题:

用牛顿迭代法等梯度优化方法求解。为此, 采用 Sigmoid 光滑函数逼近正号函数 $(x_1)_+$ 和 $(x_2)_+$ 。

Sigmoid 光滑函数 $p(x, \alpha)$ 定义如下:

$$p(x, \alpha) = x + \frac{1}{\alpha} \log_{\alpha}(1 + e^{-\alpha x}) \quad (11)$$

其中: α 为正常数。

据此可得正号函数 $(x_1)_+$ 和 $(x_2)_+$ 的 Sigmoid 光滑函数分别如下:

$$(x_1)_+ = p(e + F[\omega_1^T \delta_1]^T, \alpha) \quad (12)$$

$$(x_2)_+ = p(e - G[\omega_2^T \delta_2]^T, \alpha) \quad (13)$$

将式(12)和式(13)分别代入式(9)和式(10), 得到线性情况下加权光滑投影孪生支持向量回归算法的两个无约束优化问题, 如式(14)、式(15)所示:

考虑到权值矩阵和对角矩阵都是正定阵, 两者可以分别分解为 $P^T P$ 和 $Q^T Q$ 的形式, 其中 P 和 Q 分别为两者的平方根矩阵。因此, $\nabla^2 P_1(u_1)$ 可以分解如下:

$$\nabla^2 P_1(u_1) = C_3 I + E^T P^T P E + \alpha C_1 F^T Q^T Q F = C_3 I + (PE)^T PE + \alpha C_1 (QF)^T QF \quad (18)$$

其中: C_1, C_3 和 α 都是正标量; I 为正定矩阵。由引理1可得 $\nabla^2 P_1(u_1)$ 是正定的, 因此 P_1 是严格凸函数。同理可证 P_2 也是连续可微的严格凸函数。

式(14)和式(15)二阶可微, 因此可以使用具有快速收敛能力的牛顿迭代法进行求解^[20-21]。由定理2可知式(14)和式(15)是严格凸的, 因此使用牛顿迭代法求解可以全局收敛, 并得到唯一最优解。牛顿迭代法可表示如下:

$$\mathbf{u}_1^{k+1} = \mathbf{u}_1^k - (\nabla^2 P_1(\mathbf{u}_1))^{\top} (\nabla P_1(\mathbf{u}_1)) \quad (19)$$

$$\mathbf{u}_2^{k+1} = \mathbf{u}_2^k - (\nabla^2 P_2(\mathbf{u}_2))^{\top} (\nabla P_2(\mathbf{u}_2)) \quad (20)$$

其中: $\mathbf{u}_2 = [\omega_2^{\top} \ \delta_2^{\top}]^{\top}$ 。 $\nabla P_2(\mathbf{u}_2)$ 和 $\nabla^2 P_2(\mathbf{u}_2)$ 分别如下:

$$\nabla P_2(\mathbf{u}_2) = (C_4 \mathbf{I} + \mathbf{E}^{\top} \mathbf{W}_{ij} \mathbf{E}) \mathbf{u}_2 - C_2 \mathbf{G}^{\top} \frac{1}{1 + \exp(-\alpha(-\mathbf{G} \mathbf{u}_2 + \mathbf{e}))} \quad (21)$$

$$\nabla^2 P_2(\mathbf{u}_2) = C_4 \mathbf{I} + \mathbf{E}^{\top} \mathbf{W}_{ij} \mathbf{E} + \alpha C_2 \mathbf{G}^{\top} \text{diag} \left(\frac{\exp(-\alpha(-\mathbf{G} \mathbf{u}_2 + \mathbf{e}))}{(1 + \exp(-\alpha(-\mathbf{G} \mathbf{u}_2 + \mathbf{e})))^2} \right) \mathbf{G} \quad (22)$$

首先, 通过上述牛顿迭代法求解式(14)和式(15)可得两个最优超平面的法向量 $\mathbf{u}_1 = [\omega_1^{\top} \ \delta_1^{\top}]^{\top}$ 和 $\mathbf{u}_2 = [\omega_2^{\top} \ \delta_2^{\top}]^{\top}$ 。

然后, 通过求解式(23)和式(24)两个无约束最小化问题, 可得两个最优超平面的偏置分别如式(25)、式(26)所示:

$$\min \left\{ \frac{1}{2} \sum_{i=1}^n (\omega_1^{\top} \mathbf{x}_i + \delta_1 (y_i + \varepsilon) + b_1)^2 \right\} \quad (23)$$

$$\min \left\{ \frac{1}{2} \sum_{i=1}^n (\omega_2^{\top} \mathbf{x}_i + \delta_2 (y_i - \varepsilon) + b_2)^2 \right\} \quad (24)$$

$$b_1 = -\omega_1^{\top} \bar{\mathbf{x}} - \delta_1 (\bar{y} + \varepsilon) \quad (25)$$

$$b_2 = -\omega_2^{\top} \bar{\mathbf{x}} - \delta_2 (\bar{y} - \varepsilon) \quad (26)$$

在通常情况下, $\delta_1, \delta_2 \neq 0$ ^[14,21], 可得移位函数 $f_1(\mathbf{x}) = -(\omega_1^{\top} \mathbf{x} + b_1)/\delta_1$ 和 $f_2(\mathbf{x}) = -(\omega_2^{\top} \mathbf{x} + b_2)/\delta_2$ 。

最终, 可得回归函数如下:

$$f(\mathbf{x}) = \frac{1}{2} (f_1(\mathbf{x}) + f_2(\mathbf{x})) = -\frac{1}{2} \left(\frac{\omega_1}{\delta_1} + \frac{\omega_2}{\delta_2} \right)^{\top} \mathbf{x} - \frac{1}{2} \left(\frac{b_1}{\delta_1} + \frac{b_2}{\delta_2} \right) \quad (27)$$

1.3 非线性情况

在非线性情况下, 利用核技巧, 构造加权光滑投影孪生支持向量回归算法的两个无约束优化问题如下:

$$\min P_1 = \frac{1}{2} (\varphi(\mathbf{E})[\omega_1^{\top} \ \delta_1^{\top}])^{\top} \mathbf{W}_{ij} (\varphi(\mathbf{E})[\omega_1^{\top} \ \delta_1^{\top}]) + \frac{C_3}{2} (\|\omega_1\|^2 + \delta_1^2) + C_1 \mathbf{e}^{\top} p(\mathbf{e} + \varphi(\mathbf{F})[\omega_1^{\top} \ \delta_1^{\top}], \alpha) \quad (28)$$

$$\min P_2 = \frac{1}{2} (\varphi(\mathbf{E})[\omega_2^{\top} \ \delta_2^{\top}])^{\top} \mathbf{W}_{ij} (\varphi(\mathbf{E})[\omega_2^{\top} \ \delta_2^{\top}]) + \frac{C_4}{2} (\|\omega_2\|^2 + \delta_2^2) + C_2 \mathbf{e}^{\top} p(\mathbf{e} - \varphi(\mathbf{G})[\omega_2^{\top} \ \delta_2^{\top}], \alpha) \quad (29)$$

其中: $\varphi(\cdot)$ 表示从实空间 R 到核特征空间 H 的映射。

定理2同样适用于非线性加权光滑投影孪生支

持向量回归算法, 即式(28)和式(29)是连续可微的严格凸函数, 亦可采用具有快速收敛能力的牛顿迭代法进行求解。

通过求解式(28)和式(29)可得两个最优超平面的法

向量 $\mathbf{u}_1 = [\omega_1^{\top} \ \delta_1^{\top}]^{\top}$ 和 $\mathbf{u}_2 = [\omega_2^{\top} \ \delta_2^{\top}]^{\top}$ 。令 $\bar{\mathbf{x}} = \sum_{i=1}^n \varphi(\mathbf{x}_i)/n$, 则非线性加权光滑投影孪生支持向量回归算法的偏置 b_1 和 b_2 也可以分别用式(25)和式(26)求得, 相应的回归函数转换如下:

$$f(\mathbf{x}) = \frac{1}{2} (f_1(\mathbf{x}) + f_2(\mathbf{x})) = -\frac{1}{2} \left(\frac{\omega_1}{\delta_1} + \frac{\omega_2}{\delta_2} \right)^{\top} \varphi(\mathbf{x}) - \frac{1}{2} \left(\frac{b_1}{\delta_1} + \frac{b_2}{\delta_2} \right) \quad (30)$$

1.4 算法步骤

在线性情况下, 采用牛顿迭代法训练加权光滑投影孪生支持向量回归算法的过程如下:

输入 惩罚参数 C_1 和 C_2 , 正则化参数 C_3 和 C_4 , 不敏感损失参数 ε , 精度 t_{tol} , 最大迭代次数 i_{max} 和训练数据集矩阵 \mathbf{J}

输出 回归函数 $f(\mathbf{x})$

步骤1 通过式(4)计算权值矩阵 \mathbf{W}_{ij} 。

步骤2 给定任意初始点 \mathbf{u}_1^0 和 \mathbf{u}_2^0 , 并令迭代步骤 $i=0$ 。

步骤3 基于 \mathbf{u}_1^i 和 \mathbf{u}_2^i , 通过式(19)和式(20)分别计算 \mathbf{u}_1^{i+1} 和 \mathbf{u}_2^{i+1} 直到 $\|\mathbf{u}_1^{i+1} - \mathbf{u}_1^i\|, \|\mathbf{u}_2^{i+1} - \mathbf{u}_2^i\| < t_{\text{tol}}$ 或 $i > i_{\text{max}}$, 并通过式(25)和式(26)分别计算偏置 b_1 和 b_2 。

步骤4 通过式(27)计算 $f(\mathbf{x})$ 。

非线性情况跟线性情况类似, 此处省略。

1.5 时间复杂度分析

本节将 WSPTSVR 算法的时间复杂度与 TSVR^[12]、STSVR^[13]、PTSVR^[14] 以及快速聚类加权孪生支持向量回归 (Fast Clustering-based Weighted Twin Support Vector Regression, FCWTSVR) 算法^[9] 进行对比。由于加法的时间复杂度远小于乘法的时间复杂度, 因此本文只考虑矩阵乘法运算的次数。另外, 核矩阵的求解是所有算法都不可避免的, 因此也不作考虑。

PPTSVR、SPTSVR 与 TSVR 一样, 在对偶空间中求解二次规划问题, 其时间复杂度为 $O(n^3)$ 。而 WSPTSVR 与 STSVR 一样, 在原空间中采用牛顿迭代法进行求解, 由于迭代过程中均为对角矩阵相乘和对角矩阵求逆, 因此一次迭代的时间复杂度为 $O(n)$, 而牛顿迭代法具有快速收敛能力, 因此 STSVR 与 WSPTSVR 算法的时间复杂度要小于 TSVR 及

PPTSVR。另外,由于WSPTSVR算法在目标函数中加入了加权项,因此时间复杂度比STSVR稍大。对于FCWTSVR,其时间复杂度虽然也为 $O(n^3)$,但是还采用了快速聚类的方法剔除异常点,因此时间复杂度要大于PPTSVR、SPTSVR和TSVR。

2 实验结果与分析

2.1 实验设计与参数选择

为了验证WSPTSVR算法的有效性,将其与TSVR、STSVR、PPTSVR、SPTSVR以及FCWTSVR分别在基准测试数据集和人工测试函数上进行比较。首先对数据集进行归一化处理,然后采用标准的五折交叉验证法将数据集划分为训练集和测试集。采用RMSE、MAE、ET这3个性能指标来综合评价回归算法的性能:

$$R_{\text{RMSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (31)$$

$$M_{\text{MAE}} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (32)$$

$$E_{\text{ET}} = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (33)$$

其中: n 为样本个数; y_i 为实际输出; \hat{y}_i 为 y_i 的预测值; $\bar{y} = \sum_{i=1}^n y_i / n$ 为 y_i 的平均值。

通常地,RMSE的值越小,算法的预测性能越好;MSE的值越小,算法的预测误差越小;ET的值越小,表明预测值与真实值的一致性越好。一个好的回归算法应该综合考量RMSE、MAE和ET的值。

此外,还分别统计了各算法的CPU运行时间。所有实验都是在MATLAB 2019a平台上用Intel i5-9400F@2.90 GHz处理器、16 GB内存的计算机完成的,并且所有的实验结果均取20次独立运行结果的平均值。

参数选择与算法性能密切相关,实验采用网格搜索法选取最优参数。由于 ε_1 和 ε_2 的设置对预测性能不会产生很大影响^[13,22-23],因此将 ε_1 和 ε_2 的值都设置为0.01。在TSVR中,令 $C_1 = C_2$ 并且在集合 $\{2^i | i = -8, \dots, 0, \dots, 8\}$ 中进行寻优。为了保证算法对比的公平性^[9,24],令STSVR、PPTSVR、SPTSVR和WSPTSVR算法中 $C_1 = C_2$ 、 $C_3 = C_4$,且在集合 $\{2^i | i = -8, \dots, 0, \dots, 8\}$ 中

进行寻优。此外,在STSVR和WSPTSVR算法中,设置精度 $t_{\text{tol}} = 10^{-6}$,最大迭代次数 $i_{\text{max}} = 50$ 。在非线性实验中,核函数统一采用高斯径向基函数 $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)$,参数 σ 在集合 $\{2^i | i = -5, \dots, 0, \dots, 5\}$ 中进行寻优。

2.2 基准测试数据集

为了对比各个算法的综合性能,将其在7个基准测试数据集上进行测试,使用的基准测试数据集如表1所示,其中样本数最小为103,最大为9 568,它们可以从UCI机器学习库下载。

表1 实验中使用的7个基准测试数据集

Table 1 Seven benchmark datasets used in the experiments

数据集	样本数	特征数
Concrete slump test	103	10
Auto MPG	392	8
Boston housing	506	14
Energy efficiency	768	9
Concrete compressive	1 030	9
Airfoil self-noise	1 503	6
CCPP	9 568	5

表2统计了各个算法在7个基准测试数据集上的实验结果,为了清楚起见,最优结果加粗表示。

由表2可以看出:1)与PPTSVR相比,WSPTSVR算法在大多数数据集上有着相似或者更小的RMSE、MAE和ET,这是由于WSPTSVR算法采用孤立森林法赋予样本中的异常点很小的权值,并通过异常分数赋予每个样本点不同的权值,能够有效地降低潜在噪声或异常点对回归超平面的影响,从而提升算法的预测性能;2)与采用聚类方法去除异常点的FCWTSVR相比,WSPTSVR算法在4个数据集上的RMSE值最优,在3个数据集上的MAE值最优,而FCWTSVR的RMSE、MAE和ET值均在3个数据集上最优,表明WSPTSVR算法的预测性能与当前提出的回归算法相比具有可比性。

在算法的训练时间方面,WSPTSVR算法的时间复杂度要小于TSVR、PPTSVR以及SPTSVR,实验结果也表明,WSPTSVR算法的训练速度快于TSVR、PPTSVR以及SPTSVR。但是由于WSPTSVR算法在目标函数中引入了权值矩阵,因此与STSVR相比,训练速度稍慢。而FCWTSVR由于采用了快速聚类算法剔除异常点,其时间复杂度在6种算法中是最大的,实验结果也表明,FCWTSVR的训练速度最慢。

表 2 6 种回归算法在基准测试数据集上的实验结果

Table 2 Experimental results of six regression algorithms on the benchmark datasets

数据集	算法	RMSE	MAE	ET	CPU 运行时间/ms
Concrete slump test	TSVR	13.167 0	10.945 4	0.612 9	14.0
	STSVR	12.970 6	10.938 9	0.592 2	0.2
	PPTSVR	12.990 4	10.931 2	0.594 3	13.9
	SPTSVR	13.011 4	10.946 4	0.596 1	4.1
	FCWTSVR	12.922 4	10.926 9	0.588 3	39.8
	WSPTSVR	12.917 7	10.916 5	0.593 4	0.5
Auto MPG	TSVR	3.355 2	2.554 9	0.185 9	22.3
	STSVR	3.340 7	2.549 6	0.186 7	1.1
	PPTSVR	3.340 2	2.554 0	0.186 8	18.9
	SPTSVR	3.339 3	2.551 5	0.186 6	22.4
	FCWTSVR	3.322 9	2.533 7	0.185 4	46.3
	WSPTSVR	3.335 9	2.550 2	0.184 8	2.8
Boston housing	TSVR	4.838 2	3.371 8	0.298 9	45.2
	STSVR	4.839 2	3.357 8	0.299 1	3.1
	PPTSVR	4.846 7	3.406 5	0.300 5	27.1
	SPTSVR	4.826 4	3.374 1	0.297 6	38.6
	FCWTSVR	4.828 2	3.355 9	0.284 6	96.7
	WSPTSVR	4.825 5	3.350 3	0.295 6	4.3
Energy efficiency	TSVR	3.208 2	2.266 3	0.114 4	81.9
	STSVR	3.207 3	2.265 4	0.113 4	8.2
	PPTSVR	3.207 7	2.266 1	0.115 2	107.9
	SPTSVR	3.207 2	2.264 7	0.114 2	102.6
	FCWTSVR	3.185 1	2.243 4	0.113 9	140.5
	WSPTSVR	3.206 7	2.263 6	0.115 6	14.9
Concrete compressive	TSVR	10.452 7	8.373 6	0.395 7	142.3
	STSVR	10.432 5	8.277 4	0.396 5	7.9
	PPTSVR	10.431 6	8.276 7	0.394 7	128.5
	SPTSVR	10.431 7	8.276 6	0.394 6	139.7
	FCWTSVR	10.427 3	8.276 4	0.397 4	182.1
	WSPTSVR	10.428 6	8.271 8	0.395 2	17.2
Airfoil self-noise	TSVR	5.013 0	3.940 9	0.508 6	386.4
	STSVR	4.903 5	3.935 9	0.501 4	8.1
	PPTSVR	4.896 8	3.844 6	0.498 5	192.3
	SPTSVR	4.912 7	3.794 2	0.491 4	327.8
	FCWTSVR	4.853 5	3.737 4	0.485 7	548.0
	WSPTSVR	4.812 9	3.745 5	0.488 6	28.4
CCPP	TSVR	4.588 1	3.616 8	0.076 9	3 2981.4
	STSVR	4.601 9	3.738 8	0.081 5	197.5
	PPTSVR	4.597 2	3.718 7	0.076 8	20 133.4
	SPTSVR	4.584 6	3.676 6	0.074 6	24 762.4
	FCWTSVR	4.573 5	3.634 1	0.073 7	46 222.9
	WSPTSVR	4.559 9	3.627 2	0.071 5	548.3

2.3 人工测试函数

为了进一步验证 WSPTSVR 算法在非线性情况下的拟合性能,在 $\text{sinc}(\mathbf{x})$ 函数上进行实验,并且人为添加两种不同类型的噪声。 $\text{sinc}(\mathbf{x})$ 函数的具体形式如下:

$$y_i = \text{sinc}(\mathbf{x}_i) + n_i = \frac{\sin(\mathbf{x}_i)}{\mathbf{x}_i} + n_i, \mathbf{x}_i \in [-3\pi, 3\pi] \quad (34)$$

其中: \mathbf{x}_i 表示输入; y_i 表示对应于 \mathbf{x}_i 的输出; n_i 表示噪声。两种不同类型噪声的具体形式如下:

$$n_{i_1} = \left(-\frac{|\mathbf{x}_i|}{8\pi} + 0.5 \right) \times \xi_i, \xi_i \sim U[-0.1, 0.1] \quad (35)$$

$$n_{i_2} = \left(-\frac{|\mathbf{x}_i|}{8\pi} + 0.5 \right) \times \xi_i, \xi_i \sim N[0, 0.1^2] \quad (36)$$

其中: $U[-0.1, 0.1]$ 表示在闭区间 $[-0.1, 0.1]$ 内服从均匀分布; $N[0, 0.1^2]$ 表示服从均值为0、方差为 0.1^2 的高斯分布。

随机生成47个混入噪声的独立样本作为训练样本和150个混入噪声的独立样本作为测试样本。

另外,在训练样本中人为加入3个异常点。

表3给出了2种不同类型噪声下6种回归算法的平均RMSE、MAE、ET以及CPU运行时间,最优结果加粗表示。

表3 6种回归算法在人工测试函数上的实验结果

Table 3 Experimental results of six regression algorithms on artificial test function

噪声类型	异常点个数	算法	RMSE	MAE	ET	CPU运行时间/ms
均匀分布噪声	0	TSVR	0.017 5	0.012 3	0.002 5	6.0
		STSVR	0.016 2	0.011 8	0.002 1	1.1
		PPTSVR	0.016 7	0.011 6	0.002 1	6.4
		SPTSVR	0.016 5	0.012 2	0.002 2	6.1
		FCWTSVR	0.014 9	0.001 1	0.002 3	9.1
		WSPTSVR	0.015 3	0.011 3	0.001 8	1.7
均匀分布噪声	3	TSVR	0.055 4	0.043 1	0.031 2	7.9
		STSVR	0.055 2	0.040 8	0.022 4	1.2
		PPTSVR	0.051 8	0.041 9	0.019 7	7.2
		SPTSVR	0.039 1	0.029 1	0.012 8	6.7
		FCWTSVR	0.031 6	0.022 3	0.007 4	11.9
		WSPTSVR	0.023 9	0.017 9	0.004 3	2.4
高斯分布噪声	0	TSVR	0.028 1	0.019 3	0.005 8	6.4
		STSVR	0.024 9	0.017 5	0.004 6	1.3
		PPTSVR	0.024 6	0.017 0	0.004 5	6.8
		SPTSVR	0.024 8	0.017 6	0.004 8	6.2
		FCWTSVR	0.022 1	0.017 1	0.004 0	9.6
		WSPTSVR	0.023 4	0.016 7	0.003 8	1.8
高斯分布噪声	3	TSVR	0.061 1	0.044 6	0.025 9	9.4
		STSVR	0.050 5	0.039 1	0.017 8	1.4
		PPTSVR	0.052 2	0.040 5	0.017 4	7.6
		SPTSVR	0.048 5	0.038 2	0.016 4	7.7
		FCWTSVR	0.033 8	0.023 4	0.011 9	11.4
		WSPTSVR	0.034 1	0.026 1	0.008 2	2.3

从表3可以看出:当样本中没有异常点时,FCWTSVR和WSPTSVR的RMSE、MAE以及ET值要小于其他4种回归算法,表明FCWTSVR和WSPTSVR算法具备更好的预测性能和抗噪声能力;当样本中存在异常点时,WSPTSVR算法的RMSE、MAE和ET值均明显小于PPTSVR,以RMSE为例,WSPTSVR算法比PPTSVR平均约小44.2%,表明本文算法在样本中存在异常点时,有着更好的预测性能。这是由于本文算法采用孤立森林法赋予异常点很小的权值,因此受异常点的影响较小。另外,FCWTSVR采用聚类的方法剔除样本中的异常点,同样获得了较好的预测性能。在训练时间方面,由于WSPTSVR算法直接在原空间中采用牛顿迭代法进行求解,训练速度比TSVR、PPTSVR和SPTSVR更快,而与STSVR相比,由于目标函数中添

加了权值矩阵,因此训练速度稍慢。

图1给出了高斯分布噪声下不同回归算法在 $\text{sinc}(x)$ 函数上的拟合曲线。从图1可以看出,与PPTSVR相比,WSPTSVR算法在噪声和异常点的干扰下更接近真实曲线,拟合效果更好。这是由于WSPTSVR算法采用孤立森林法判断样本中的异常点,并利用样本的异常分数赋予噪声和异常点很小的权值,从而获得较好的拟合效果。反观TSVR、STSVR、PPTSVR和SPTSVR在异常点处的拟合曲线扭曲较为明显,受异常点的影响较大,拟合效果较差。原因是这4种回归算法赋予所有样本点相同的权值,而异常点的存在迫使拟合曲线偏向异常点,从而使得拟合效果变差。另外,FCWTSVR采用聚类的方法确定并剔除异常点,因此在异常点处也获得了较好的拟合效果。

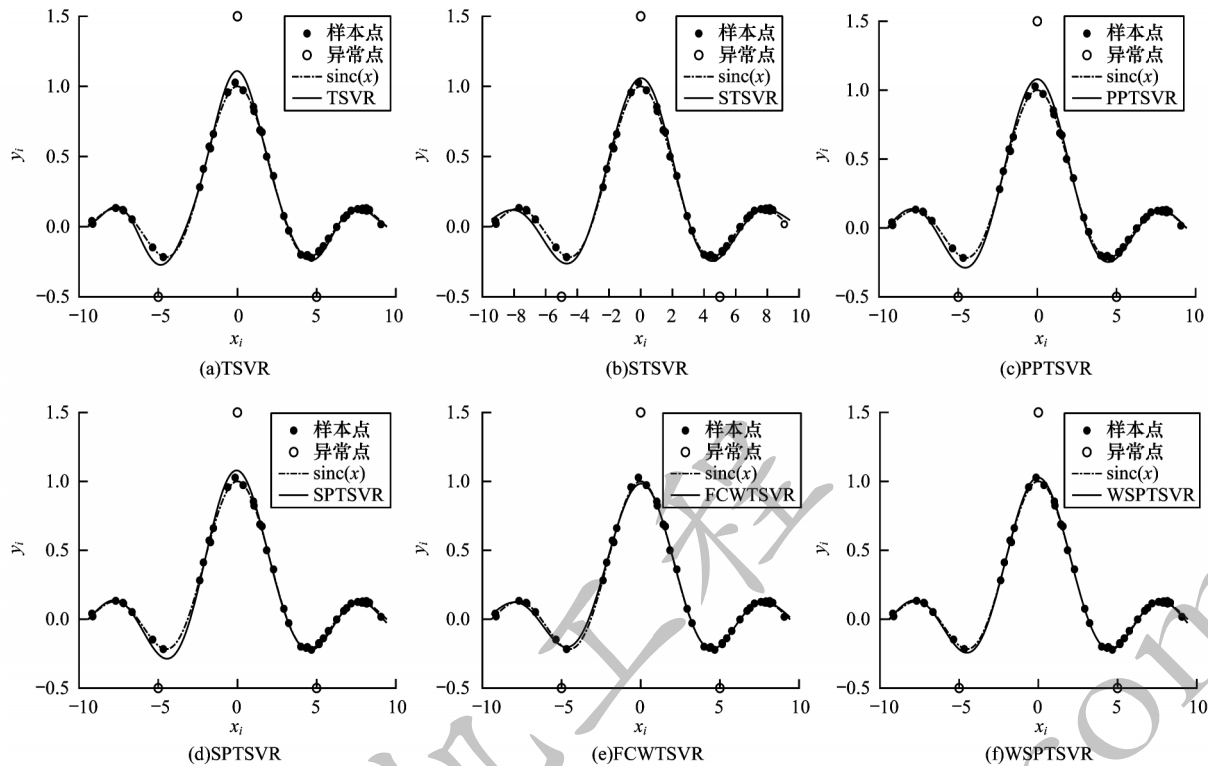


图1 高斯分布噪声下6种回归算法的拟合曲线

Fig.1 Fitting curves of six regression algorithms under noise with Gaussian distribution

4 结束语

本文提出一种加权光滑投影孪生支持向量回归算法。采用孤立森林法判断样本中潜在的异常点,并通过赋予潜在的异常点很小的权值,有效地削弱了其对超平面构造的影响。引入Sigmoid光滑函数,通过在原空间中采用牛顿迭代法求解无约束优化问题,获得了比双边移位投影孪生支持向量回归算法更快的训练速度。实验结果表明,与其他代表性回归算法相比,该算法受异常点的影响更小,拟合效果更佳。下一步将从寻找逼近能力更优的光滑函数入手,使WSPTSVR算法获得更好的拟合性能。

参考文献

- [1] VAPNIK V N. The nature of statistical learning theory[M]. New York, USA: Springer, 1995.
- [2] VAPNIK V N. Statistical learning theory[M]. New York, USA: Wiley, 1998.
- [3] 王海,翁晨傲,李克,等.一种面向基站扇区方向角估计的改进SVM算法[J]. 计算机工程,2021,47(4):120-126. WANG H, WENG C N, LI K, et al. An improved SVM algorithm for azimuth estimation of base station sector[J]. Computer Engineering, 2021, 47(4): 120-126. (in Chinese)
- [4] 鲁淑霞,蔡莲香,张罗幻.基于动量加速零阶减小方差的鲁棒支持向量机[J]. 计算机工程,2020,46(12):88-95,104. LU S X, CAI L X, ZHANG L H. Robust support vector machine based on momentum acceleration zero-order variance reduction[J]. Computer Engineering, 2020, 46(12): 88-95, 104. (in Chinese)
- [5] KIM S T, HAN I G, LEE C Y, et al. A development of unknown intrusion detection system with SVM[J]. Convergence Security Journal, 2007, 7(4): 23-28.
- [6] CHEN Y, DING S H, HU G L, et al. Facial beautification method based on age evolution[J]. Computer Aided Drafting, Design and Manufacturing, 2013, 23(4): 7-12.
- [7] BEN ABID F, ZGARNI S, BRAHAM A. Distinct bearing faults detection in induction motor by a hybrid optimized SWPT and aiNet-DAG SVM[J]. IEEE Transactions on Energy Conversion, 2018, 33(4): 1692-1699.
- [8] 罗彬坤,刘利民,董健,等.基于SAE-GA-SVM模型的雷达新型干扰识别[J]. 计算机工程,2020,46(6):281-287. LUO B S, LIU L M, DONG J, et al. Radar new jamming identification based on SAE-GA-SVM model[J]. Computer Engineering, 2020, 46(6): 281-287. (in Chinese)
- [9] GU B J, FANG J W, PAN F, et al. Fast clustering-based weighted twin support vector regression[J]. Soft Computing, 2020, 24(8): 6101-6117.
- [10] KHEMCHANDANI J R, CHANDRA S. Twin support vector machines for pattern classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(5): 905-910.
- [11] CHEN X, YANG J, YE Q. Recursive projection twin support vector machine via within-class variance minimization[J]. Pattern Recognition, 2011, 44(10/11): 2643-2655.
- [12] PENG X J. TSVR: an efficient twin support vector machine for regression[J]. Neural Networks, 2010, 23(3): 365-372.
- [13] CHEN X B, YANG J, LIANG J, et al. Smooth twin support vector regression[J]. Neural Computing and Applications, 2012, 21(3): 505-513.
- [14] PENG X J, CHEN D. PTSVR: regression models via projection twin support vector machine[J]. Information Sciences, 2018, 435: 1-14.

(下转第118页)

(上接第111页)

- [15] LIU F T, TING K M, ZHOU Z H. Isolation forest[C]// Proceedings of the 8th IEEE International Conference on Data Mining. Washington D. C. , USA: IEEE Press, 2009: 413-422.
- [16] LIU F T, TING K M, ZHOU Z H. Isolation-based anomaly detection[J]. ACM Transactions on Knowledge Discovery from Data, 2012, 6(1): 1-39.
- [17] FENG W, SHEN G L, XU B Y, et al. Isolation forest-based least squares twin margin distribution support vector regression[J]. International Journal of Innovative Computing, Information and Control, 2021, 17(2): 565-579.
- [18] DING S F, HUANG H J, XU X Z, et al. Polynomial smooth twin support vector machines[J]. Applied Mathematics & Information Sciences, 2014, 8(4): 2063-2071.
- [19] BARLOW J L. Matrix analysis[J]. Computing Reviews, 2013, 54(8): 462-463.
- [20] WANG L D, GAO C, ZHAO N N, et al. A projection wavelet weighted twin support vector regression and its primal solution[J]. Applied Intelligence, 2019, 49(8): 3061-3081.
- [21] BI J B, BENNETT K P. A geometric approach to support vector regression[J]. Neurocomputing, 2003, 55(1/2): 79-108.
- [22] TANVEER M, SHUBHAM K. A regularization on Lagrangian twin support vector regression[J]. International Journal of Machine Learning and Cybernetics, 2017, 8(3): 807-821.
- [23] LÓPEZ J, MALDONADO S. Robust twin support vector regression via second-order cone programming[J]. Knowledge-Based Systems, 2018, 152: 83-93.
- [24] GU B J, SHEN G L, PAN F, et al. Least squares twin projection support vector regression [J]. International Journal of Innovative Computing Information and Control, 2019, 15(6): 2275-2288.

编辑 陆燕菲