孪生支持向量回归机研究进展

丁世飞[1,2],张子晨[1],郭丽丽[1,2],张 健[1,2],徐 晓[1,2]
(1. 中国矿业大学计算机科学与技术学院,江苏徐州 221116;
2. 矿山数字化教育部工程研究中心(中国矿业大学),江苏徐州 221116)

摘 要： 孪生支持向量回归机(Twin Support Vector Regression,TSVR or TWSVR)是一种基于统计学习理论的回归算法,它以结构风险最小化原理为理论基础,通过适当地选择函数子集及该子集中的判别函数,使学习机的实际风险达到最小,保证了在有限训练样本上得到的小误差分类器对独立测试集的测试误差仍然较小.孪生支持向量回归机通过将线性不可分样本映射到高维特征空间,使得映射后的样本在该高维特征空间内线性可分,保证了其具有较好的泛化性能.孪生支持向量回归机的算法思想基于孪生支持向量机(Twin Support Vector Machine,TWSVM),几何意义是使所有样本点尽可能地处于两条回归超平面的上(下)不敏感边界之间,最终的回归结果由两个超平面的回归值取平均得到.孪生支持向量回归机需求解两个规模较小的二次规划问题(Quadratic Programming Problems,QPPs)便可得到两条具有较小拟合误差的回归超平面,训练时间和拟合精度都高于传统的支持向量回归机(Support Vector Regression,SVR),且其QPPs的对偶问题存在全局最优解,避免了容易陷入局部最优的问题,故孪生支持向量回归机已成为机器学习的热门领域之一.但孪生支持向量回归机作为机器学习领域的一个较新的理论,其数学模型与算法思想都尚不成熟,在泛化性能、求解速度、矩阵稀疏性、参数选取、对偶问题等方面仍存在进一步改进的空间.本文首先给出了两种孪生支持向量回归机的数学模型与几何意义,然后将孪生支持向量回归机的几个常见的改进策略归纳如下.

(1)加权孪生支持向量回归机

由于孪生支持向量回归机中每个训练样本受到的惩罚是相同的,但每个样本对超平面的影响不同,尤其是噪声和离群值会使算法性能降低,并且在不同位置的训练样本应给予不同的处罚更为合理,因此考虑在孪生支持向量回归机的每个QPP中引入一个加权系数,给予不同位置的训练样本不同程度的惩罚.

(2)拉格朗日孪生支持向量回归机

由于孪生支持向量回归机的对偶问题中半正定矩阵的逆矩阵可能不存在,若存在,则对偶问题不是严格凸函数,可能存在多个解,因此考虑使用松弛变量的2范数代替原有的1范数,使对偶问题更简单,易于求解.

(3)最小二乘孪生支持向量回归机

由于孪生支持向量回归机的求解需要在对偶空间进行,得到的解为近似解,考虑通过最小二乘法将原问题的不等式约束转化为等式约束,使得原问题可以在原空间内求解,在很大程度上降低计算时间,提高泛化性能,且不损失精度.

(4)ν-孪生支持向量回归机

通过引入一组参数ν_1与ν_2自动调节ε_1与ε_2的值以控制训练样本的特定部分对两条回归超平面所能造成的最大误差,从而自适应给定数据的结构,提高孪生支持向量回归机的拟合精度.

(5)ε-孪生支持向量回归机

在孪生支持向量回归机的原问题中引入正则化项以达到结构风险最小化的目的,使对偶问题转化为稳定的正定二次规划问题,并通过SOR求解对偶问题,加快训练速度.

(6)孪生参数不敏感支持向量回归机

克服参数的选取对孪生支持向量回归机超平面构造的影响,使算法非常适合于存在异方差噪声数据的数据集,训练速度和泛化性能也有提升.

本文同时对以上算法的数学模型、改进算法及应用进行了系统地分析与总结,给出了以上算法在9个UCI基准数据集上的回归性能与计算时间,并在模型结构层面逐一分析每个算法的表现与耗时的根本原因.对于其他不便于归类的孪生支持向量回归机改进算法及应用,本文也对其作逐一总结.整体来看,最小二乘孪生支持向量回归机在性能和计算时间方面表现最佳,拉格朗日孪生支持向量回归机、ν-孪生支持向量回归机的性能并列次优且计算时间接近,加权孪生支持向量回归机、ε-孪生支持向量回归机和孪生参数不敏感支持向量回归机的性能不理想,但计算时间接近.本文旨在使读者对孪生支持向量回归机的不同改进算法之间的异同点与优缺点产生更深刻的理解与认识,从而将更多优秀的改进策略应用于孪生支持向量回归机,最终为进一步提高孪生支持向量回归机的性能以及扩展孪生支持向量回归机的应用范围提供较为清晰的思路.

收稿日期:2022-06-20;修回日期:2022-09-23;责任编辑:朱梅玉

关键词： 孪生支持向量回归机；拟合精度；泛化能力；计算时间
基金项目： 国家自然科学基金（No.61976216，No.62276265）
中图分类号： TP311 **文献标识码：** A **文章编号：** 0372-2112(2023)04-1117-18
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220703

Survey on Twin Support Vector Regression

DING Shi-fei[1,2], ZHANG Zi-chen[1], GUO Li-li[1,2], ZHANG Jian[1,2], XU Xiao[1,2]

（1. *School of Computer Science and Technology*, *China University of Mining and Technology*, *Xuzhou*, *Jiangsu* 221116, *China*;
2. *Mine Digitization Engineering Research Center of Ministry of Education*（*China University of Mining and Technology*）,
Xuzhou, *Jiangsu* 221116, *China*）

Abstract: Twin support vector regression is a regression algorithm based on statistical learning theory. It employs the theoretical principle of structural risk minimization; by appropriately selecting a subset of functions and obtaining the discriminant functions in that subset, it minimizes the actual risk of the learning machine, ensuring that the small test error of a classifier obtained on a limited training sample remains small on an independent test set. Twin support vector regression ensures better generalization performance because it maps linearly inseparable samples to a high-dimensional feature space, making the mapped samples linearly separable within that high-dimensional feature space. The algorithm of twin support vector regression is based on the twin support vector machine, and the geometric meaning of twin support vector regression is to make all sample points as far as possible between the upper (lower) insensitive boundaries of the two regression hyperplanes. The final regression result is then obtained by averaging the regression values of the two hyperplanes. In twin support vector regression, it is only necessary to solve two small-scale quadratic programming problems (QPPs) to obtain two regression hyperplanes with small fitting errors. Hence, its training time and fitting accuracy are higher than those of traditional support vector regression. Moreover, the dual problems of two QPPs have a globally optimal solution, and this makes it harder for the algorithm to become trapped in local optima. Hence, twin support vector regression has become a popular field of machine learning. However, there is still room for further improvement in the generalization performance, solution speed, matrix sparsity, parameter selection, and dual problem of twin support vector regression. In this paper, the mathematical models and geometric meanings of two types of twin support vector regression are presented; then, the following common improvement strategies for twin support vector regression are summarized:

(1) Weighted twin support vector regression

Because each training sample in twin support vector regression receives the same penalty but has different effects on the hyperplane, noise and outliers can degrade the performance of twin support vector regression. It is more reasonable for training samples at different locations to be given different penalties; thus, the introduction of a weighted factor in each QPP of twin support vector regression has been proposed to give training samples at different locations different degrees of penalties.

(2) Lagrange twin support vector regression

The inverse matrix of the semi-positive definite matrix in the dual problem of twin support vector regression may not exist. Even if it does, the dual problem is not a strictly convex function and may have multiple solutions. Thus the use of the 2-norm of the slack variables to replace the original 1-norm has been considered to make the dual problem simpler and easier to solve.

(3) Least squares twin support vector regression

Twin support vector regression needs to be solved in dual space, and the obtained solution is approximate. Hence, a method was proposed that transforms the inequality constraint of the original problem into an equation constraint using the least squares method so that the original problem can be solved in the original space. This substantially reduces the computation time and improves the generalization performance without loss of accuracy.

(4) ν-twin support vector regression

This approach introduces a set of parameters ν_1 and ν_2 to control the maximum error that a particular part of the training sample can cause to the two regression hyperplanes. This adapts the hyperplanes to the structure of the given data and improves the fitting accuracy of the twin support vector regression.

(5) ε-twin support vector regression

A regularization term is introduced into the original formulation of twin support vector regression to minimize the structural risk so that the dual problem is transformed into a stable positive definite quadratic programming problem. The dual problem is solved by successive over relaxation (SOR) method to speed up the training.

(6) Twin parameter insensitive support vector regression

This method overcomes the effect of parameter selection on twin support vector regression, making the algorithm well suited for data sets that contain heteroscedastic noisy data as well as enhancing training speed and generalization performance.

This paper also systematically analyzes and summarizes the mathematical models, improved algorithms, and applications of the above algorithms; analyzes the regression performance and calculation time of the above algorithms on nine UCI benchmark datasets through experiments; and analyzes the root cause of each algorithm's performance and time consumption at the level of model structure. This paper also summarizes other improved twin support vector regression algorithms and applications of twin support vector regression that are not easy to categorize. Overall, the least squares twin support vector regression performs the best in terms of performance and computation time. Lagrangian twin support vector regression and v-twin support vector regression both obtain the next best performance and have similar computation times. Weighted twin support vector regression, ε-twin support vector regression, and twin parameter insensitive support vector regression do not perform well, but have similar computation times. The purpose of this paper is to provide readers with a deeper understanding and perception of the similarities, differences, advantages, and disadvantages of different improved algorithms based on twin support vector regression, so that they may apply better improvement strategies to twin support vector regression. Moreover, this paper aims to eventually provide a clear idea for further improving the performance of twin support vector regression and expanding its range of applications.

Key words: twin support vector regression; fitting accuracy; generalization ability; computational time

Foundation Item(s): National Natural Science Foundation of China (No.61976216, No.62276265)

1 引言

支持向量机(Support Vector Machine,SVM)是一种用于分类和回归问题的出色工具,由Cortes等人于1995年提出[1]. 作为SVM的回归形式,支持向量回归机(Support Vector Regression,SVR)自然也成为解决连续数值拟合等问题的首要工具[2],但其在求解时需要求解一对二次规划问题,因此引入了更多变量和约束,扩大了问题的范围且增加了问题的计算复杂性. 因此,SVR在处理高维复杂数据集时面临着很大的困难[3]. 尽管目前存在分块算法[4]、分解算法[5]和顺序最小优化(Sequential Minimal Optimization,SMO)算法[6]等降低SVM或SVR计算复杂度的方法,但这些算法的实现较为烦琐,不易推广使用.

孪生支持向量机(Twin Support Vector Machine,TWSVM)[7,8]是在SVM的基础上发展而来的一种二分类机器学习算法. 它的算法思想是为每个类中的点构建一个超平面,使每个类中的点都环绕其本类超平面,且远离非本类超平面超过1距离. 每个超平面通过求解一对二次规划问题(Quadratic Programming Problems,QPPs)得到,因此其规模较SVM中的QPP小很多,计算速度是SVM的四倍,且性能比SVM有明显提高[7].

鉴于SVR存在的诸多缺点和TWSVM在弥补SVM缺点上取得的成功,学者们转而把目光集中在孪生支持向量回归机(Twin Support Vector Regression,TSVR or TWSVR)的构建上. Peng于2010年构建了一种孪生支持向量回归机的形式,称为TSVR[3]. TSVR旨在生成两个非平行超平面以分别确定未知回归变量的不敏感函数的上下界,最终的回归结果在两个超平面的回归结果中取平均值,其计算速度也是SVR的4倍,且其泛化性能比SVR有进一步的提升[3]. TWSVM的原作者Khemchandani等人指出TSVR存在一些理论上的问题,如在TSVR中ε_1和ε_2仅对超参数b起作用,对超参数w不起作用,且回归变量仅取决于$\varepsilon_1-\varepsilon_2$,这意味着当$\varepsilon_1$和$\varepsilon_2$相等时,其值对回归结果没有影响,进而在Bi等人提出的数学理论推导[9]的支持下提出了另一种形式的孪生支持向量回归机,即TWSVR[10,11]. 目前,TSVR和TWSVR已被应用于风速预测、股价预测、正交频分复用系统信道估计等领域中[12~14]. TSVR和TWSVR在机器学习领域是一个较新的理论,在泛化性能、求解速度和矩阵稀疏性等方面仍存在许多缺点以待改进与完善.

本文的主要贡献概括为:(1)介绍了TSVR和TWSVR的数学模型;(2)从TSVR和TWSVR的几个频次较高的改进策略的角度分析TSVR和TWSVR的最新改进算法与应用,并分析每个改进策略的优缺点;(3)通过实验验证TSVR和TWSVR及其各个改进策略的性能与计算时间;(4)通过对以上改进策略的总结给出未来的研究方向.

2 两种孪生支持向量回归机的数学模型

本节分别介绍TSVR和TWSVR的数学模型. 为了不失一般性,在本文中,假设在高维空间$\mathbb{R}^n$中存在l个训练样本,且可表示为$\boldsymbol{A}=(\boldsymbol{A}_1;\ \boldsymbol{A}_2;\ \cdots;\ \boldsymbol{A}_l)$,每个训练样本$\boldsymbol{A}_i$又可表示为$\boldsymbol{A}_i=(A_{i1},A_{i2},\cdots,A_{in}),i=1,2,\cdots,l$. 显然,矩阵$\boldsymbol{A}$的每行代表一个样本点. 令$\boldsymbol{Y}=(y_1,y_2,\cdots,y_l)\in\mathbb{R}$代表响应变量集合,在回归问题中$y_i(i=1,2,\cdots,l)$的取值为实值. 令$\boldsymbol{e}$为长度为$l$,元素全为1的列向量.

2.1 TSVR

考虑所要寻找的两个非平行超平面分别为$f_1(\boldsymbol{x})=\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}+b_1$和$f_2(\boldsymbol{x})=\boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x}+b_2$,它们由求解以下两个QPPs得到:

$$\min_{\boldsymbol{w}_1,b_1\in\mathbb{R}}\left[\frac{1}{2}\left(\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\right)^{\mathrm{T}}\left(\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\right)+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}\right] \tag{1}$$
$$\text{s.t.}\ \ \boldsymbol{Y}-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\geqslant\boldsymbol{e}\varepsilon_1-\boldsymbol{\xi},\quad\boldsymbol{\xi}\geqslant 0$$

$$\min_{\boldsymbol{w}_2,b_2\in\mathbb{R}}\left[\frac{1}{2}\left(\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\right)^{\mathrm{T}}\left(\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\right)+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}\right] \tag{2}$$
$$\text{s.t.}\ \ \boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2-\boldsymbol{Y}\geqslant\boldsymbol{e}\varepsilon_2-\boldsymbol{\eta},\quad\boldsymbol{\eta}\geqslant 0$$

其中,惩罚参数$C_1,C_2>0,\varepsilon_1,\varepsilon_2\geqslant 0,\boldsymbol{\xi}$和$\boldsymbol{\eta}$为松弛变量. TSVR的几何意义是使所有样本点尽可能地处于$f_1(\boldsymbol{x})$超平面的ε_1不敏感下边界与$f_2(\boldsymbol{x})$超平面的ε_2不敏感上边界之间. 最终的回归结果由两个超平面的回归值取平均得到.

在求解上述目标函数时,因其原始形式较难直接求解,常转而求其对偶问题. 其对偶问题为

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\sigma\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{f}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\sigma\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\alpha}\right] \tag{3}$$
$$\text{s.t.}\ \ 0\leqslant\boldsymbol{\alpha}\leqslant C_1\boldsymbol{e}$$

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\sigma\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\sigma\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\gamma}-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\gamma}\right]$$
$$\text{s.t.}\ \ 0\leqslant\boldsymbol{\gamma}\leqslant C_2\boldsymbol{e} \tag{4}$$

其中$\boldsymbol{\alpha}=(\alpha_1;\ \alpha_2;\ \cdots;\ \alpha_l),\boldsymbol{\gamma}=(\gamma_1;\ \gamma_2;\ \cdots;\ \gamma_l)$为引入的拉格朗日乘子,$\boldsymbol{G}=[\boldsymbol{A}\ \boldsymbol{e}],\boldsymbol{f}=\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1,\boldsymbol{h}=\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2,\sigma\boldsymbol{I}$为正则化项,$\sigma$是一个很小的数(如10的−7次方,视样本点的数据值而定). 求得$\boldsymbol{\alpha}$和$\boldsymbol{\gamma}$后,即可求得两个超平面. 最终的回归器通过如下函数得到回归结果:

$$f(\boldsymbol{x})=\frac{1}{2}\left(f_1(\boldsymbol{x})+f_2(\boldsymbol{x})\right)=\frac{1}{2}\left(\boldsymbol{w}_1+\boldsymbol{w}_2\right)^{\mathrm{T}}\boldsymbol{x}+\frac{1}{2}\left(b_1+b_2\right) \tag{5}$$

对于非线性情况,两个非平行超平面分别为$f_1(\boldsymbol{x})=\boldsymbol{w}_1^{\mathrm{T}}K\left(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}}\right)+b_1$和$f_2(\boldsymbol{x})=\boldsymbol{w}_2^{\mathrm{T}}K\left(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}}\right)+b_2$,在构建和求解超平面中,只要令$\boldsymbol{A}=K\left(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}}\right)$,即求得两个超平面,其中$K$为核函数. 最终的回归器为

$$f(\boldsymbol{x})=\frac{1}{2}\left(f_1(\boldsymbol{x})+f_2(\boldsymbol{x})\right)=\frac{1}{2}\left(\boldsymbol{w}_1+\boldsymbol{w}_2\right)^{\mathrm{T}}K\left(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}}\right)+\frac{1}{2}\left(b_1+b_2\right) \tag{6}$$

2.2 TWSVR

与TSVR相同,TWSVR也要寻找$f_1(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_1+b_1$和$f_2(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_2+b_2$两个非平行超平面. 两个超平面对应的一组QPPs分别为

$$\min_{\boldsymbol{w}_1,b_1\in\mathbb{R}}\left[\frac{1}{2}\left\|\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\right\|^2+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}_1\right]$$
$$\text{s.t.}\ \ \left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)-\left(\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1\right)\leqslant\boldsymbol{\xi}_1,\quad\boldsymbol{\xi}_1\geqslant 0 \tag{7}$$

$$\min_{\boldsymbol{w}_2,b_2\in\mathbb{R}}\left[\frac{1}{2}\left\|\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\right\|^2+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}_2\right]$$
$$\text{s.t.}\ \ \boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\leqslant\boldsymbol{\xi}_2,\quad\boldsymbol{\xi}_2\geqslant 0 \tag{8}$$

通过分别在式(7)和式(8)中引入正则化项以控制结构风险,两个超平面的QPPs分别为

$$\min_{\boldsymbol{w}_1,b_1\in\mathbb{R}}\left[\frac{C_3}{2}\left(\left\|\boldsymbol{w}_1\right\|^2+b_1^2\right)+\frac{1}{2}\left\|\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\right\|^2+C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}_1\right] \tag{9}$$
$$\text{s.t.}\ \ \left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)-\left(\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1\right)\leqslant\boldsymbol{\xi}_1,\quad\boldsymbol{\xi}_1\geqslant 0$$

$$\min_{\boldsymbol{w}_2,b_2\in\mathbb{R}}\left[\frac{C_4}{2}\left(\left\|\boldsymbol{w}_2\right\|^2+b_2^2\right)+\frac{1}{2}\left\|\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\right\|^2+C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}_2\right] \tag{10}$$
$$\text{s.t.}\ \ \boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\leqslant\boldsymbol{\xi}_2,\quad\boldsymbol{\xi}_2\geqslant 0$$

其对偶问题为

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+C_3\boldsymbol{I}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{f}^{\mathrm{T}}\boldsymbol{H}\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+C_3\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\alpha}-\left(\varepsilon_1+\varepsilon_2\right)\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha}\right] \tag{11}$$
$$\text{s.t.}\ \ 0\leqslant\boldsymbol{\alpha}\leqslant C_1\boldsymbol{e}$$

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{H}\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+C_4\boldsymbol{I}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\gamma}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{H}\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+C_4\boldsymbol{I}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\gamma}-\boldsymbol{g}^{\mathrm{T}}\boldsymbol{\alpha}-\left(\varepsilon_1+\varepsilon_2\right)\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\gamma}\right] \tag{12}$$
$$\text{s.t.}\ \ 0\leqslant\boldsymbol{\gamma}\leqslant C_2\boldsymbol{e}$$

其中,$\boldsymbol{\alpha}=(\alpha_1;\ \alpha_2;\ \cdots;\ \alpha_l),\boldsymbol{\gamma}=(\gamma_1;\ \gamma_2;\ \cdots;\ \gamma_l)$为引入的拉格朗日乘子,$\boldsymbol{H}=[\boldsymbol{A}\ \boldsymbol{e}],\boldsymbol{f}=\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2,\boldsymbol{g}=\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1,C_3$和$C_4$为正则化项参数. 求得$\boldsymbol{\alpha}$和$\boldsymbol{\gamma}$后,即完成对TWSVR的求解. 最终的回归器依然由式(5)得到,对于非线性情况,两个非平行超平面依然分别为$f_1(\boldsymbol{x})=K\left(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}}\right)\boldsymbol{w}_1+b_1$和$f_2(\boldsymbol{x})=K\left(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{T}}\right)\boldsymbol{w}_2+b_2$,在构建和求解超平面中,依然在构建和求解超平面中令$\boldsymbol{A}=K\left(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}}\right)$,即求得两个超平面,其中$K$为核函数,最终的回归器仍然由式(6)得到. 注意到TWSVR的对偶问题式(11)和式(12)与

TSVR的对偶问题式(3)和式(4)最大的差别在于TWSVR的对偶问题式多了$(\varepsilon_1+\varepsilon_2)\boldsymbol{e}^{\mathrm{T}}$项,这也是TWSVR可以克服TSVR中$\varepsilon_1$和$\varepsilon_2$仅对超参数$b$起作用,对超参数$\boldsymbol{w}$不起作用,且$\varepsilon_1$和$\varepsilon_2$相等时,其值对回归结果没有影响这两大问题的原因[3,10].

3 孪生支持向量回归机研究进展

作为机器学习领域的一种新理论,TSVR和TWSVR因其明显的优势成为一个研究热点. 然而原始TSVR和TWSVR在算法鲁棒性、泛化性能、求解速度方面仍然存在一些不足与缺陷. 目前,学者们对TSVR和TWSVR的改进研究主要集中在引入加权法、拉格朗日法、最小二乘法、正则化法、引入不同的损失函数、小波以及参数不敏感法等方法,提高TSVR和TWSVR训练速度、泛化能力,并降低回归误差. 下面将通过回顾TSVR和TWSVR的几个最常被使用的改进策略来探究其最新理论研究进展.

3.1 加权孪生支持向量回归机

3.1.1 加权孪生支持向量回归机的数学模型

加权孪生支持向量回归机(Weighted Twin Support Vector Regression, WTSVR)由Xu等人于2012年提出[15]. 考虑到在TSVR中每个训练样本受到的惩罚是相同的,但是每个样本点对超平面的影响不同,尤其是噪声和离群值会使被确定的平面的性能降低,并且在不同位置的训练样本应给予不同的处罚更为合理,Xu等人引入了两个加权系数σ_1和σ_2来根据训练样本的不同位置进行不同的惩罚. 在线性可分情况下,WTSVR所要求解的QPPs为

$$\min_{\boldsymbol{w}_1, b_1\in\mathbb{R}}\left[\frac{1}{2}\left\|\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\right\|^2+C_1\left(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}+\sigma_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}^*\right)\right]$$
$$\text{s.t. } \boldsymbol{Y}-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\geqslant\boldsymbol{e}\varepsilon_1-\boldsymbol{\xi}-\boldsymbol{\xi}^* \tag{13}$$

$$\min_{\boldsymbol{w}_2, b_2\in\mathbb{R}}\left[\frac{1}{2}\left\|\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\right\|^2+C_1\left(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}+\sigma_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}^*\right)\right]$$
$$\text{s.t. } \left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)-\boldsymbol{Y}\geqslant\boldsymbol{e}\varepsilon_2-\boldsymbol{\eta}-\boldsymbol{\eta}^* \tag{14}$$

其中,$\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\eta},\boldsymbol{\eta}^*$为松弛变量;$\sigma_1$和$\sigma_2$为引入的惩罚参数(即权重),需要预先设置. 令$\boldsymbol{G}=[\boldsymbol{A}\ \boldsymbol{e}]$,$\boldsymbol{f}=\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1$,$\boldsymbol{h}=\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2$,可得式(13)和式(14)的对偶问题分别为

$$\min_{\alpha,\gamma\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\delta\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{f}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\delta\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\alpha}+\varepsilon_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\gamma}\right] \tag{15}$$
$$\text{s.t. } 0\leqslant\boldsymbol{\alpha}\leqslant C_1\sigma_1\boldsymbol{e},\quad \boldsymbol{\gamma}\geqslant 0$$

$$\min_{\beta,\rho\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\delta\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\beta}+\boldsymbol{h}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\delta\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\beta}-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{\beta}+\varepsilon_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\rho}\right] \tag{16}$$
$$\text{s.t. } 0\leqslant\boldsymbol{\beta}\leqslant C_2\sigma_2\boldsymbol{e},\quad \boldsymbol{\rho}\geqslant 0$$

求得$\boldsymbol{\alpha}$和$\boldsymbol{\beta}$后,通过$\boldsymbol{u}_1=\left[\boldsymbol{w}_1;\ b_1\right]=\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\delta\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{f}-\boldsymbol{\alpha}\right)$与$\boldsymbol{u}_2=\left[\boldsymbol{w}_2;\ b_2\right]=\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\delta\boldsymbol{I}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{h}-\boldsymbol{\beta}\right)$即可得到WTSVR超平面的参数. 对于非线性情况,只要在式(13)~(16)的基础上令$\boldsymbol{A}=K\left(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}}\right)$,即求得两个超平面,其中$K$为核函数. 最终的回归器依然由式(6)获得. 实验结果表明WTSVR比SVR和TSVR取得了更小的拟合误差.

3.1.2 加权孪生支持向量回归机的改进算法

Xu等人[16]在WTSVR的基础上进一步利用训练样本的局部位置信息,提出基于K近邻加权TSVR(K-Nearest Neighbor-based WTSVR, KNNWTSVR),同时引入了逐次超松弛(Successive Over Relaxation, SOR)来提高对偶问题的求解速度. Tanveer等人[17]将正则化项添加到KNNWTSVR的目标函数中,并用2范数代替松弛变量的1范数,提出正则化KNNWTSVR(Regularized version of the KNN-based WTSVR, RKNNWTSVR),不仅克服了奇异点对超平面的影响,而且提高了泛化性能. Gupta等人[18]提出的具有原空间无约束最小化问题的KNNWTSVR(KNNWTSVR in primal as a pair of Unconstrained minimization Problems, KNNUPWTSVR),具有更好的泛化性能和更快的学习速度. Wang等人[19]提出一种投影小波WTSVR(Projection Wavelet WTSVR, PWWTSVR),在WTSVR的基础上采用小波变换理论计算权重矩阵以预处理训练样本,进一步减少训练样本中噪声与离群值对超平面确定的影响. Gu等人[20]提出基于快速聚类的WTSVR(Fast Clustering-based WTSVR, FC-WTSVR),采用密度峰值聚类(Density Peak Clustering, DPC)作为快速聚类算法来确定聚类中心和异常值,并引入协方差矩阵以进一步反映样本的先验结构信息. 程昊翔等人[21]在原始空间中计算密度加权值,同时在目标函数中引入正则化项,提出了一种基于密度加权的TSVR(DM-TSVR). DM-TSVR可以很好地反映训练数据集的内在分布,使数据点准确影响训练模型. 徐奔业等人[22]提出加权光滑投影TSVR(Weighted Smooth Projection TSVR, WSPTSVR). 该算法先用孤立森林赋权重,再用正号函数将目标函数转化为无约束优化问题,并用sigmoid光滑函数进行光滑处理,进而在原空间使用Newton法求解.

3.1.3 加权孪生支持向量回归机的应用

Wang等人[14]还将PWWTSVR应用于正交频分复用系统(Orthogonal Frequency Division Multiplexing system, OFDM)的信道估计中. 李朔等人[23]提出一种基于K近邻的WTSVR(K-Nearest Neighbor TSVR, KNN-TSVR),用于信道参数的插值估计. 马艺梅等人[24]建立了基于小波加权的TSVR(Wavelet Transform based

WTSVR，WTWTSVR）并用于估计基于导频的信道.

3.1.4 加权孪生支持向量回归机的优缺点

WTSVR通过给予样本以不同程度的惩罚（即权重），使得最终的回归器可以在一定程度上避免过拟合问题，产生很强的泛化能力. 同时，该方法原理简单，容易实现. 但WTSVR只考虑了经验风险，没有考虑结构化风险，且难以自适应地处理对偶问题中半正定矩阵$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$的求逆问题. 同时，每个样本点权重的赋予方式较为粗糙，在构建两个超平面时对一部分样本点赋予相同的权重，增加了数据中噪声与奇异点对超平面的影响，且其目标函数失去了稀疏性.

3.2 拉格朗日孪生支持向量回归机

3.2.1 拉格朗日孪生支持向量回归机的数学模型

拉格朗日孪生支持向量回归机（Lagrangian Twin Support Vector Regression，LTSVR）由Balasundaram等人于2013年提出[25]. 考虑到在式（3）和式（4）中的$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$是半正定矩阵，它的逆矩阵有可能不存在，若存在，则式（3）和式（4）并不是严格凸函数，可能存在多个解，因此Balasundaram等人使用松弛变量的2范数的平方来代替原始TSVR的1范数，并提出了LTSVR. 在线性可分情况下，LTSVR所要求解的QPPs为

$$\min_{\boldsymbol{w}_1,b_1\in\mathbb{R}}\left[\frac{1}{2}\left\|\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\right\|^2+\frac{C_1}{2}\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{\xi}_1\right]\quad(17)$$
$$\text{s.t. }\boldsymbol{Y}-\left(\boldsymbol{A}\boldsymbol{w}_1+\boldsymbol{e}b_1\right)\geqslant\boldsymbol{e}\varepsilon_1-\boldsymbol{\xi}_1,\quad\boldsymbol{\xi}_1\geqslant 0$$

$$\min_{\boldsymbol{w}_2,b_2\in\mathbb{R}}\left[\frac{1}{2}\left\|\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2-\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)\right\|^2+\frac{C_2}{2}\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{\xi}_2\right]\quad(18)$$
$$\text{s.t. }\left(\boldsymbol{A}\boldsymbol{w}_2+\boldsymbol{e}b_2\right)-\boldsymbol{Y}\geqslant\boldsymbol{e}\varepsilon_2-\boldsymbol{\xi}_2,\quad\boldsymbol{\xi}_2\geqslant 0$$

令$\boldsymbol{G}=[\boldsymbol{A}\ \boldsymbol{e}]$，可得式（17）和（18）的简单对偶形式为

$$\min_{\boldsymbol{u}_1\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{Q}_1\boldsymbol{u}_1-\boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{u}_1\right]\quad(19)$$
$$\text{s.t. }\boldsymbol{u}_1\geqslant 0$$

$$\min_{\boldsymbol{u}_2\in\mathbb{R}}\left[\frac{1}{2}\boldsymbol{u}_2^{\mathrm{T}}\boldsymbol{Q}_2\boldsymbol{u}_2-\boldsymbol{r}_2^{\mathrm{T}}\boldsymbol{u}_2\right]\quad(20)$$
$$\text{s.t. }\boldsymbol{u}_2\geqslant 0$$

其中，$\boldsymbol{Q}_1=I/C_1+\boldsymbol{H}$，$\boldsymbol{Q}_2=I/C_2+\boldsymbol{H}$，$\boldsymbol{H}=\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}$，$\boldsymbol{r}_1=\left(\boldsymbol{H}-\boldsymbol{I}\right)\left(\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1\right)$，$\boldsymbol{r}_2=\left(\boldsymbol{I}-\boldsymbol{H}\right)\left(\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2\right)$. 解得$\boldsymbol{u}_1$和$\boldsymbol{u}_2$后，可通过$\left[\boldsymbol{w}_1;\ b_1\right]=\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{Y}-\boldsymbol{e}\varepsilon_1-\boldsymbol{u}_1\right)$，$\left[\boldsymbol{w}_2;\ b_2\right]=\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{Y}+\boldsymbol{e}\varepsilon_2+\boldsymbol{u}_2\right)$得到LTSVR超平面的参数. 对于非线性情况，只要在式（17）~（20）的基础上令$\boldsymbol{A}=K\left(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}}\right)$，即求得两个超平面，其中，$K$为核函数. 最终的回归器依然由式（6）获得.

3.2.2 拉格朗日孪生支持向量回归机的改进算法

Tanveer等人[26]在LTSVR的基础上加入正则化项并提出了正则化LTSVR（Regularization for LTSVR，RLTSVR）. 在此基础上，他们又将隐式正则化项加入目标函数以使$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$保持为正定矩阵，增加了算法的泛化性和鲁棒性，并提出了隐式RLTSVR（Implicit RLTSVR，IRLTSVR）[27]. IRLTSVR减少了计算复杂度，但目标函数包含一个不光滑项，Tanveer等人采用了广义Hessian法与引入光滑函数来处理. Gupta等人[28]提出具有pinball损失的鲁棒不对称LTSVR（Robust Asymmetric Lagrangian ν-TSVR Using Pinball Loss Function，URALTSVR）. 其对偶形式无任何约束，且pinball损失函数可以灵活处理训练样本中存在的非对称噪声以及重采样的不稳定性，还可使$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$保持为正定矩阵，从而提高泛化性能. Balasundaram等人[29]将LTSVR的对偶问题处理为无约束优化的形式并用基于梯度的迭代法求解，提出了一种无约束凸优化TSVR（Unconstrained LTSVR，ULTSVR），该算法的对偶问题是分段二次的且可微的，采用广义Hessian法、光滑函数和Newton-Armijo法或通过函数迭代算法可获得其对偶问题解.

3.2.3 拉格朗日孪生支持向量回归机的应用

Ganaie等人[30]提出一种改进的TSVR（Improved TSVR，ITSVR），实现了结构风险最小化原则，避免了矩阵求逆的计算，使用SOR求解，应用于脑年龄估计.

3.2.4 拉格朗日孪生支持向量回归机的优缺点

LTSVR求解的迭代过程极其简单，在计算速度上有很大优势. 其最终的对偶形式很简单，因为需要进行迭代求解，所以相比较TSVR的对偶问题更易求解，求解时间也更短. 同时，LTSVR不影响泛化性能并能在较少的训练时间内提供较小的误差. 但LTSVR中只考虑了经验风险而未考虑结构风险最小化，放弃了SVR标志性的平衡结构风险最小化与经验风险的优点，因此在某些情况下可能会导致过拟合或只能寻找到局部最优值. 同时，LTSVR并没有彻底解决WTSVR中半正定矩阵$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$的求逆较为困难的问题.

3.3 最小二乘孪生支持向量回归机

3.3.1 最小二乘孪生支持向量回归机的数学模型

学者们在将最小二乘法与孪生支持向量回归机结合的过程中产生了两种形式，即孪生最小二乘支持向量回归机（Twin Least Squares Support Vector Regression，TLSSVR）[31]和最小二乘孪生支持向量回归机（Least Squares Twin Support Vector Regression，LSTSVR）[32]. 在本文中，统一使用"最小二乘孪生支持向量回归机"来代指这两种算法. 下面介绍这两种算法的数学模型.

3.3.1.1 TLSSVR的数学模型

TLSSVR由Zhao等人于2013年提出[30]. 它的设计思想来源于最小二乘支持向量回归机（Least Squares Support Vector Regression，LSSVR）. 在非线性情况下，TLSSVR要求解的QPPs为

$$\min_{w_1, b_1 \in \mathbb{R}} \left[\frac{1}{2} \| \boldsymbol{w}_1 \|^2 + \frac{1}{2} C_1 \boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{\xi}_1^2 \right]$$
$$\text{s.t. } \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1 + \varepsilon_1 + \boldsymbol{\xi}_1, \quad \boldsymbol{\xi}_2 \geqslant 0 \tag{21}$$
$$\min_{w_2, b_2 \in \mathbb{R}} \left[\frac{1}{2} \| \boldsymbol{w}_2 \|^2 + \frac{1}{2} C_2 \boldsymbol{v}_2^{\mathrm{T}} \boldsymbol{\xi}_2^2 \right]$$
$$\text{s.t. } \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2 - \varepsilon_2 + \boldsymbol{\xi}_2, \quad \boldsymbol{\xi}_2 \geqslant 0 \tag{22}$$

其中,$\boldsymbol{v}_1$和$\boldsymbol{v}_2$为调节松弛变量$\boldsymbol{\xi}_1$和$\boldsymbol{\xi}_2$的权重.通过在式(21)和式(22)中引入拉格朗日乘子,可得如下解:

$$\begin{bmatrix} 0 & \boldsymbol{e}^{\mathrm{T}} \\ \boldsymbol{e} & \tilde{\boldsymbol{K}}_1 \end{bmatrix} \begin{bmatrix} b_1 \\ \boldsymbol{\alpha}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y} - \boldsymbol{e}\varepsilon_1 \end{bmatrix} \tag{23}$$
$$\begin{bmatrix} 0 & \boldsymbol{e}^{\mathrm{T}} \\ \boldsymbol{e} & \tilde{\boldsymbol{K}}_2 \end{bmatrix} \begin{bmatrix} b_2 \\ \boldsymbol{\alpha}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y} + \boldsymbol{e}\varepsilon_2 \end{bmatrix} \tag{24}$$

其中,$\boldsymbol{\alpha}_1$和$\boldsymbol{\alpha}_2$为引入的拉格朗日乘子,$\boldsymbol{v}_1$和$\boldsymbol{v}_2$的定义如下:

$$\boldsymbol{v}_1 = \begin{cases} \rho_1, \boldsymbol{\alpha}_1 < 0 \\ 1, \boldsymbol{\alpha}_1 \geqslant 0 \end{cases}, \quad \boldsymbol{v}_2 = \begin{cases} \rho_1, \boldsymbol{\alpha}_2 < 0 \\ 1, \boldsymbol{\alpha}_2 \geqslant 0 \end{cases} \tag{25}$$

其中,ρ_1和ρ_2为定值,$\tilde{\boldsymbol{K}}_k = K\left(\boldsymbol{A}_i, \boldsymbol{A}_j\right) + \delta_{ij}/\boldsymbol{v}_k C_k, k=1,2$,$\delta_{ij} = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases}$.两个所求的超平面为$f_1(\boldsymbol{x}) = \boldsymbol{\alpha}_1^{\mathrm{T}} K\left(\boldsymbol{A}, \boldsymbol{x}\right) + b_1$和$f_2(\boldsymbol{x}) = \boldsymbol{\alpha}_2^{\mathrm{T}} K\left(\boldsymbol{A}, \boldsymbol{x}\right) + b_2$,其中,$K$为核函数,最终的回归器依然由式(6)获得.在小规模与大规模数据集上的实验结果证明了TLSSVR的有效性与效率.

3.3.1.2　LSTSVR的数学模型

LSTSVR由Huang等人于2013年提出[32].在线性情况下,最小二乘孪生支持向量回归机所要求解的QPPs为

$$\begin{aligned} &\min_{w_1, b_1 \in \mathbb{R}} \left[\frac{1}{2} \left\| \boldsymbol{Y} - \boldsymbol{e}\varepsilon_1 - \left(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1 \right) \right\|^2 + \frac{1}{2} C_1 \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{\xi} \right] \\ &\text{s.t. } \boldsymbol{Y} - \left(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1 \right) = \boldsymbol{e}\varepsilon_1 - \boldsymbol{\xi}, \quad \boldsymbol{\xi} \geqslant 0 \end{aligned} \tag{26}$$
$$\begin{aligned} &\min_{w_2, b_2 \in \mathbb{R}} \left[\frac{1}{2} \left\| \boldsymbol{Y} + \boldsymbol{e}\varepsilon_2 - \left(\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2 \right) \right\|^2 + \frac{1}{2} C_2 \boldsymbol{\eta}^{\mathrm{T}} \boldsymbol{\eta} \right] \\ &\text{s.t. } \left(\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2 \right) - \boldsymbol{Y} = \boldsymbol{e}\varepsilon_2 - \boldsymbol{\eta}, \quad \boldsymbol{\eta} \geqslant 0 \end{aligned} \tag{27}$$

令$\boldsymbol{G} = \left[\boldsymbol{A}\ \boldsymbol{e} \right]$,$\boldsymbol{f} = \boldsymbol{Y} - \boldsymbol{e}\varepsilon_1$,$\boldsymbol{h} = \boldsymbol{Y} + \boldsymbol{e}\varepsilon_2$,从式(26)和式(27)可得到$\boldsymbol{u}_1 = \left[\boldsymbol{w}_1;\ b_1 \right] = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \omega \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{f}$和$\boldsymbol{u}_2 = \left[\boldsymbol{w}_2;\ b_2 \right] = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \omega \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{h}$,从而直接求得两个超平面.对于非线性情况,只要在式(26)和式(27)的基础上令$\boldsymbol{A} = K\left(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}} \right)$,即求得两个超平面,其中,$K$为核函数.最终的回归器依然由式(6)获得.实验结果说明LSTSVR比SVR,LSSVR和TSVR的误差更小,且计算时间明显缩短.

3.3.2　最小二乘孪生支持向量回归机的改进算法

Zhang等人[33]将p范数引入LSTSVR,提出一种L_p范数LSTSVR(L_p-norm LSTSVR,PLSTSVR),旨在最小化结构风险的同时降低目标函数的求解难度.Huang等人[34]提出一种特征选择LSTSVR(Feature Selection-LSTSVR,FS-LSTSVR),将LSTSVR中的目标函数转化为标准线性规划(Linear Programming,LP),并构造了对偶空间中的L_p问题,然后采用广义Newton法解决.Gu等人[35]提出最小二乘投影TSVR(Least Squares Twin Projection Support Vector Regression,LSTPSVR).该方法将投影数据的方差整合到目标函数中,并在求解线性方程的过程中将输入和输出数据的经验协方差和相关系数嵌入最终的线性矩阵中.王怡芮等人[36]将大间隔思想引入LSTSVR以增强模型的泛化性能,同时引入模糊隶属度函数衡量训练样本的隶属度值,提出了模糊最小二乘大间隔TSVR(Least Squares large interval Fuzzy TSVR,LSFTSVR).该算法能够很好地反映训练数据集的内在分布,使数据点准确影响训练模型.曹杰等人[37]提出了一种增量式约简LSTSVR(Incremental Reduced LSTSVR,IRLSTSVR)算法.该算法利用约简方法选取一定样本作为支持向量并构建行列不等的核矩阵,再通过分块矩阵求逆方法高效更新逆矩阵,在保证解的稀疏性的同时增加泛化性能.Feng等人[38]提出了一种基于隔离森林的最小二乘孪生边缘分布SVR(Isolation Forest-based Least Squares Twin Margin Distribution SVR,IFLSTMDSVR).该算法利用隔离森林来隔离潜在的异常值,并给它们分配合适的异常值分数,以避免LSTSVR对异常值敏感的问题.同时,该算法考虑了边界分布信息.

3.3.3　最小二乘孪生支持向量回归机的应用

Zhang等人提出具有异方差高斯噪声的TLSSVR(TLSSVR of Heteroscedastic Gaussian Noise Model,TLSSVR-HGN)[39]和具有同方差高斯噪声的TLSSVR(TLSSVR of Homoscedastic Gaussian Noise Model,TLSSVR-GN)[39],以弥补TLSSVR中存在的不足.Yan等人提出p-LSTSVR(PLSTSVR)[40],并设计了一种迭代算法来解决PLSTSVR的优化问题,同时使用两种元启发式优化算法优化PLSTSVR的参数,用于预测交通流量数据.Wang使用LSTSVR预测关键性能指标(Key Performance Indicators,KPIs)[41].Zhang等人[42]提出一种具有异方差高斯噪声的孪生近端LSSVR(Twin Proximal LSSVR Model Based on Heteroscedastic Gaussian Noise,TPLSSVR-HGN).该算法较TLSSVR提高了训练速度、泛化能力和预测精度.Hazarika等人[43]提出了基于小波核的LSTSVR,即Morlet小波核LSTSVR和Mexican hat小波核LSTSVR,用于风速预测.

3.3.4　最小二乘孪生支持向量回归机的优缺点

最小二乘孪生支持向量回归机将TSVR的目标函数中的不等式约束转化为等式约束,因此最终的解只需要在原空间内求解两个线性方程组即可,不必转化到对偶空间求解,可大大节省计算时间.同时,最小二乘孪生支持向量回归机的泛化性更好,计算复杂度更

小且训练速度更快,且不损失精度,在处理高维大规模数据集时此优势尤为明显. 但算法对异常点的敏感程度较高,噪声与孤立点等异常点会导致超平面的性能不佳.

3.4 ν-孪生支持向量回归机

3.4.1 ν-孪生支持向量回归机的数学模型

在 Shao 等人[44]提出的 ε-孪生支持向量回归机(ε-TSVR,ε-TSVR)和Peng等人[45]提出的ν-孪生支持向量机(ν-TWSVM,ν-TWSVM)以及在Bi等人[9]提出的数学理论推导的支持下,Rastogi 等人[46]于 2017 年提出了 ν-TWSVM 真正意义上的回归形式,即 ν-孪生支持向量回归机(ν-Twin Support Vector Regression,ν-TWSVR). 在线性可分情况下,ν-TWSVR所要求解的QPPs为

$$\min_{\boldsymbol{w}_1, b_1 \in \mathbb{R}} \left[\frac{1}{2} C_1\left(\left\| \boldsymbol{w}_1 \right\|^2 + b_1^2 \right) + \frac{1}{2} \left\| \boldsymbol{Y} - \left(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1 \right) \right\|^2 + C_2\left(\nu_1 \varepsilon_1 + \frac{1}{l} \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\xi} \right) \right] \tag{28}$$

$$\text{s.t. } \boldsymbol{Y} - \left(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1 \right) \geqslant -\boldsymbol{e}\varepsilon_1 - \boldsymbol{\xi}$$

$$\min_{\boldsymbol{w}_2, b_2 \in \mathbb{R}} \left[\frac{1}{2} C_3\left(\left\| \boldsymbol{w}_2 \right\|^2 + b_2^2 \right) + \frac{1}{2} \left\| \boldsymbol{Y} - \left(\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2 \right) \right\|^2 + C_4\left(\nu_2 \varepsilon_2 + \frac{1}{l} \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\eta} \right) \right] \tag{29}$$

$$\text{s.t. } \left(\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2 \right) - \boldsymbol{Y} \geqslant -\boldsymbol{e}\varepsilon_2 - \boldsymbol{\eta}$$

其中,$\nu_1, \nu_2, C_1, C_2, C_3, C_4 \geqslant 0$ 为预先指定的参数,$\boldsymbol{\xi}$ 和 $\boldsymbol{\eta}$ 为松弛变量,l 为训练集样本数. 令 $\boldsymbol{G} = [\boldsymbol{A}\ \boldsymbol{e}]$,引入拉格朗日乘子 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$,可得式(28)和式(29)的对偶问题分别为

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}} \left[\frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + C_1 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\alpha} - \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + C_1 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\alpha} + \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{\alpha} \right] \tag{30}$$

$$\text{s.t. } 0 \leqslant \boldsymbol{\alpha} \leqslant \frac{C_2}{l} \boldsymbol{e}, \quad \boldsymbol{e}^{\mathrm{T}} \alpha \leqslant C_2 \nu_1$$

$$\min_{\boldsymbol{\beta} \in \mathbb{R}} \left[\frac{1}{2} \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\beta} + \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\beta} - \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{\beta} \frac{1}{2} \boldsymbol{u}_2^{\mathrm{T}} \boldsymbol{Q}_2 \boldsymbol{u}_2 - \boldsymbol{r}_2^{\mathrm{T}} \boldsymbol{u}_2 \right] \tag{31}$$

$$\text{s.t. } 0 \leqslant \boldsymbol{\beta} \leqslant \frac{C_4}{l} \boldsymbol{e}, \quad \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\beta} \leqslant C_4 \nu_2$$

求得 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 后,通过 $\boldsymbol{u}_1 = \left[\boldsymbol{w}_1;\ b_1 \right] = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + C_1 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\alpha})$ 与 $\boldsymbol{u}_2 = \left[\boldsymbol{w}_2;\ b_2 \right] = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}}\left(\boldsymbol{Y} + \boldsymbol{\beta} \right)$ 即可得到 ν-TWSVR 超平面的参数. 对于非线性情况,只要在式(28)~(31)的基础上令 $\boldsymbol{A} = K\left(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}} \right)$,即求得两个超平面,其中 K 为核函数. 最终的回归器依然由式(6)获得. 实验表明 ν-TWSVR 在人工数据集及 UCI 基准数据集上的表现优于 ε-TSVR.

3.4.2 ν-孪生支持向量回归机的改进算法

Xu 等人[47]提出一种不对称 ν-TSVR(Asymmetric ν-TSVR,Asy-ν-TSVR). Asy-ν-TSVR 的提出基于 TSVR 而不是 TWSVR,但也通过引入 pinball 损失函数、学习训练样本的分布给予处于不同位置的训练样本不同程度的惩罚. Asy-ν-TSVR 可以通过选择 ν 与 p 两个参数有效控制拟合误差,提高泛化能力. Gupta 等人[48]在 Asy-ν-TSVR 的基础上又引入一种基于改进正则化的拉格朗日法,构建了使用pinball损失的改进正则化拉格朗日 Asy-ν-TSVR(Improved Regularization based Lagrangian Asy-ν-TSVR using Pinball Loss Function, LAsy-ν-TSVR). 根据3.2节所述,拉格朗日法主要用于获得获得原问题的对偶问题的全局最优解. 向目标函数中添加改进正则化项的目的是实现模型的结构风险最小化原则. 类似于LTSVR,该方法的对偶问题也可以转化为线性,从而可以通过简单的线性收敛方法获得,降低其计算复杂度. Xue 等人[49]将 ν-SVR 与 ν-TWSVR 结合并提出了粗糙 ν-TWSVR(Rough ν-TWSVR). Rough ν-TWSVR 取消了 ν-TWSVR 中利用极端样本点的机制,改而利用更多的训练样本的信息,还根据训练样本的位置使不同的样本对回归器产生不同的影响. Rough ν-TWSVR 考虑到 ε 不敏感函数可能会产生的过拟合问题以及 ν-TWSVR 中对所有训练样本都进行同样力度的惩罚的问题,引入了粗糙集理论以实现结构风险最小化. Wang 等人[50]在 ν-TWSVR 的目标函数中引入 WT 以对训练样本前处理、引入正则化项以控制结构化风险并利用3.1节所述的加权方法对不同位置的样本进行不同的惩罚,最终提出基于 WT 的加权 ν-TWSVR(Wavelet Transform-based Weighted ν-TWSVR, WTWTSVR).

3.4.3 ν-孪生支持向量回归机的应用

Li 等人[51]设计了 OFDM 系统中基于 ν-TWSVR 的信道估计器. ν-TWSVR 计算出的支持向量较少,使得信道估计器的测试速度更快、内存消耗更少.

3.4.4 ν-孪生支持向量回归机的优缺点

ν-TWSVR 可以通过给定的参数 ν_1 与 ν_2 自动调节 ε_1 与 ε_2 的值以控制训练样本的某些特定部分对上(下)边界回归器所能造成的最大误差,因此可以根据给定数据的结构获得更高的拟合精度. 但当构建对 ε 不敏感的上(下)界回归器时,只考虑上(下)界超平面附近的那些点容易导致过拟合问题. 此外,在 ν-TWSVR 中,所有的训练样本都被给予相同的惩罚,事实上,不同的样本对回归器函数有不同的影响. 这可能会由于噪声的影响而降低其性能.

3.5 ε-孪生支持向量回归机

3.5.1 ε-孪生支持向量回归机的数学模型

ε-孪生支持向量回归机(ε-twin support vector regression, ε-TSVR)由 Shao 等人于 2013 年提出[44]. ε-TSVR 通过求解两个类似于 SVM 的问题来得到一对 ε-不敏感约近函数,即两个超平面. 在线性可分情况下,ε-孪生支持向量回归机所要求解的 QPPs 为

$$\min_{\boldsymbol{w}_1, b_1 \in \mathbb{R}} \left[\frac{1}{2} C_3 \left(\| \boldsymbol{w}_1 \|^2 + b_1^2 \right) + \frac{1}{2} \boldsymbol{\xi}^{*\mathrm{T}} \boldsymbol{\xi}^* + C_1 \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\xi} \right] \quad \text{s.t. } \boldsymbol{Y} - \left(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e} b_1 \right) = \boldsymbol{\xi}^* \tag{32}$$

$$\min_{\boldsymbol{w}_2, b_2 \in \mathbb{R}} \left[\frac{1}{2} C_4 \left(\| \boldsymbol{w}_2 \|^2 + b_2^2 \right) + \frac{1}{2} \boldsymbol{\eta}^{*\mathrm{T}} \boldsymbol{\eta}^* + C_2 \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\eta} \right] \quad \text{s.t. } \left(\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e} b_2 \right) - \boldsymbol{Y} = \boldsymbol{\eta}^* \tag{33}$$

其中,$C_1, C_2, C_3, C_4, \varepsilon_1, \varepsilon_2$ 均为非负的参数. 令 $\boldsymbol{G} = [\boldsymbol{A}\ \boldsymbol{e}]$, 引入拉格朗日乘子 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\gamma}$, 可得式(32)和式(33)的对偶问题分别为

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}} \left[\frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{G} \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\alpha} - \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{G} \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\alpha} + \left(\boldsymbol{e}^{\mathrm{T}} \varepsilon_1 + \boldsymbol{Y}^{\mathrm{T}} \right) \boldsymbol{\alpha} \right] \quad \text{s.t. } 0 \leqslant \boldsymbol{\alpha} \leqslant C_1 \boldsymbol{e} \tag{34}$$

$$\min_{\boldsymbol{\gamma} \in \mathbb{R}} \left[\frac{1}{2} \boldsymbol{\gamma}^{\mathrm{T}} \boldsymbol{G} \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + C_4 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\gamma} + \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{G} \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{\gamma} + \left(\boldsymbol{Y}^{\mathrm{T}} - \boldsymbol{e}^{\mathrm{T}} \varepsilon_2 \right) \boldsymbol{\gamma} \right] \quad \text{s.t. } 0 \leqslant \boldsymbol{\gamma} \leqslant C_2 \boldsymbol{e} \tag{35}$$

求得 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\gamma}$ 后,通过 $\boldsymbol{u}_1 = [\boldsymbol{w}_1;\ b_1] = \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + C_3 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} (\boldsymbol{Y} - \boldsymbol{\alpha})$ 与 $\boldsymbol{u}_2 = [\boldsymbol{w}_2;\ b_2] = \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + C_4 \boldsymbol{I} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} (\boldsymbol{Y} + \boldsymbol{\gamma})$ 即可得到 ε-TSVR 超平面的参数. 对于非线性情况,只要在式(32)~(35)的基础上令 $\boldsymbol{A} = K\left(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}} \right)$, 即求得两个超平面,其中 K 为核函数,最终的回归器依然由式(6)获得. 实验证明相较于 TSVR, LSSVR 以及 ε-SVR, ε-TSVR 可以在更短的训练时间内明显提高正则化性能.

3.5.2 ε-孪生支持向量回归机的改进算法

Ye 等人[52]将距离计算方法由 L_2 范数改为 L_1 范数以提高 ε-TSVR 的泛化能力,进而提出 L_1 范数 ε-TSVR(L_1-norm ε-TSVR, L_1-ε-TSVR). Ye 等人[53]还进一步将 3.1 节和 3.2 节所述的加权法与拉格朗日法结合到 ε-TSVR 中并提出了加权拉格朗日 ε-TSVR(Weighted Lagrangian ε-TSVR with Quadratic Loss Functions, WL-ε-TSVR). 加权法用于给予不同样本不同的处罚权值以限制离群点对超平面的影响,拉格朗日法以松弛变量的 2 范数的平方来代替 ε-TSVR 中的 1 范数,同时二次损失函数的引入使其对偶问题只需要求解无约束优化问题即可,提高了训练速度.

3.5.3 ε-孪生支持向量回归机的应用

张冰等人[13]提出依赖最近邻 ε-TSVR(Dependency Nearest Neighbor ε-TSVR, DNN-ε-TSVR)并用于预测股市中高频振荡的股价数据.

3.5.4 ε-孪生支持向量回归机的优缺点

ε-TSVR 在其原始问题中引入了正则化性以达到结构风险最小化的目的,使它的对偶问题为稳定的正定二次规划问题,从而提高其性能. 另外,ε-TSVR 通过 SOR 方法求解对偶问题,从而加快了训练速度. 但 ε-TSVR 中以 L_2 范数计算距离的方法对可能存在大误差的数据集比较敏感且不鲁棒,同时,与 v-TWSVR 类似,由于异常值的影响,所有样本都被给予相同的惩罚,这会降低回归器的回归性能.

3.6 孪生参数不敏感支持向量回归机

3.6.1 孪生参数不敏感支持向量回归机的数学模型

孪生参数不敏感支持向量回归机(Twin Parametric Insensitive Support Vector Regression, TPISVR)由 Peng 等人于 2012 年提出[54]. TPISVR 以 v/c 的值来控制不敏感上下界的误差与支持向量. 在非线性情况下,孪生参数不敏感支持向量回归机所要求解的 QPPs 为

$$\min_{\boldsymbol{w}_1, b_1 \in \mathbb{R}} \left[\frac{1}{2} \| \boldsymbol{w}_1 \|^2 - \frac{v_1}{l} \boldsymbol{e}^{\mathrm{T}} \left(\varphi(\boldsymbol{A}) \boldsymbol{w}_1 + \boldsymbol{e} b_1 \right) + \frac{C_1}{l} \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\xi} \right] \quad \text{s.t.} \boldsymbol{Y} \geqslant \boldsymbol{e}^{\mathrm{T}} \left(\varphi(\boldsymbol{A}) \boldsymbol{w}_1 + \boldsymbol{e} b_1 \right) - \boldsymbol{\xi},\ \ \boldsymbol{\xi} \geqslant 0 \tag{36}$$

$$\min_{\boldsymbol{w}_2, b_2 \in \mathbb{R}} \left[\frac{1}{2} \| \boldsymbol{w}_2 \|^2 + \frac{v_2}{l} \boldsymbol{e}^{\mathrm{T}} \left(\varphi(\boldsymbol{A}) \boldsymbol{w}_2 + \boldsymbol{e} b_2 \right) + \frac{C_2}{l} \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\eta} \right] \quad \text{s.t.} \boldsymbol{Y} \leqslant \boldsymbol{e}^{\mathrm{T}} \left(\varphi(\boldsymbol{A}) \boldsymbol{w}_2 + \boldsymbol{e} b_2 \right) + \boldsymbol{\eta},\ \ \boldsymbol{\eta} \geqslant 0 \tag{37}$$

其中,$v_1, v_2, C_1, C_2 > 0$ 为预先指定的参数. 引入拉格朗日乘子 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$, 从式(36)和式(37)可以得到

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}} \left[\frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \varphi(\boldsymbol{A}) \varphi(\boldsymbol{A})^{\mathrm{T}} \boldsymbol{\alpha} + \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{\alpha} - \frac{v_1}{l} \boldsymbol{e}^{\mathrm{T}} \varphi(\boldsymbol{A}) \varphi(\boldsymbol{A})^{\mathrm{T}} \boldsymbol{\alpha} \right] \quad \text{s.t. } 0 \leqslant \boldsymbol{\alpha} \leqslant \frac{C_1}{l} \boldsymbol{e},\ \ \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\alpha} = v_1 \tag{38}$$

$$\min_{\boldsymbol{\beta} \in \mathbb{R}} \left[\frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \varphi(\boldsymbol{A}) \varphi(\boldsymbol{A})^{\mathrm{T}} \boldsymbol{\alpha} + \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{\alpha} - \frac{v_1}{l} \boldsymbol{e}^{\mathrm{T}} \varphi(\boldsymbol{A}) \varphi(\boldsymbol{A})^{\mathrm{T}} \boldsymbol{\alpha} \right] \quad \text{s.t. } 0 \leqslant \boldsymbol{\beta} \leqslant \frac{C_2}{l} \boldsymbol{e},\ \ \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\beta} = v_2 \tag{39}$$

在求得 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 后,两个超平面通过 $\boldsymbol{w}_1 = \varphi(\boldsymbol{A})^{\mathrm{T}} \left(v_1 / l \times \boldsymbol{e} - \boldsymbol{\alpha} \right)$, $\boldsymbol{w}_2 = \varphi(\boldsymbol{A})^{\mathrm{T}} \left(\boldsymbol{\beta} - v_2 / l \times \boldsymbol{e} \right)$, $\quad b_1 = \mathrm{mean}\left(\sum_{i \in \mathrm{SV}_1} \left(Y_i - \varphi(\boldsymbol{A}_i) \boldsymbol{w}_1 \right) \right)$, $b_2 = \mathrm{mean}\left(\sum_{i \in \mathrm{SV}_2} \left(Y_i - \varphi(\boldsymbol{A}_i) \boldsymbol{w}_2 \right) \right)$ 得到,其中 mean 代表求平均函数, SV_1 是满足 $\alpha_i \in \left(0, C_1/l \right), i = 1, 2, \cdots, l$ 的样本点的集合, SV_2 是满足 $\beta_i \in \left(0, C_2/l \right), i = 1, 2, \cdots, l$ 的样本点的集合, $|\cdot|$ 代表集合的基数, $\boldsymbol{A}_i$ 和 Y_i 分别为训练集中第 i 个数据点及其响应变量, K 为核函数. 最终的回归器由下式获得:

$$f(\boldsymbol{x})=\frac{1}{2}\left(f_1(\boldsymbol{x})+f_2(\boldsymbol{x})\right)=\frac{1}{2}\left(\boldsymbol{\beta}-\boldsymbol{\alpha}+\frac{v_1-v_2}{l}\boldsymbol{e}\right)^{\mathrm{T}}\boldsymbol{x}+\frac{1}{2}\left(b_1+b_2\right) \tag{40}$$

3.6.2 孪生参数不敏感支持向量回归机的改进算法

Gupta 等人[55]将 3.2 节所述的拉格朗日法引入 TPISVR 并提出拉格朗日 TPISVR（Lagrangian TPISVR，LTPISVR）. 在此基础上，他们又将 LTPISVR 中的拉格朗日项进一步改为隐式拉格朗日项，提出无约束 LTPISVR（Unconstrained LTPISVR，ULTPISVR）[56]. ULTPISVR 中采用广义 Hessian 法与光滑函数处理 LTPISVR 中目标函数存在的不光滑部分，因此将目标函数的对偶问题转化为无约束问题. 丁世飞等人[57]将 LSTSVR 中的最小二乘理论与 TPISVR 结合，提出了最小二乘 TPISVR（Least Squares TPISVR，LSTPISVR），同时改进一种元启发式优化算法来优化 LSTPISVR 的参数. 黄华娟等人[58]提出光滑 TPISVR（Smooth TPISVR，STPISVR）. 首先引入正号函数，将 TPISVR 目标函数转化为2个不可微的无约束优化问题；其次引入 CHKS 光滑函数和正则化项对 TPISVR 进行正则化以及光滑逼近，从而将不可微的模型转化为可微的无约束优化问题，并用牛顿-阿米霍方法求解.

3.6.3 孪生参数不敏感支持向量回归机的优缺点

TPISVR 的两个目标函数确定训练样本的上下界对参数不敏感，使其非常适合于存在异方差噪声数据的数据集. 实验表明 TPISVR 比 SVR，TSVR 等算法拥有更快的训练速度和更好的泛化性能. 但 TPISVR 的目标函数不是强凸的，这会导致算法可能无法找到全局最优解，只能找到局部最优解，降低了拟合性能.

3.7 孪生支持向量回归机的其他改进算法及应用

3.7.1 改进算法

Peng 等人[59]提出一种处理训练样本中噪声点的 TSVR，称之为孪生投影 SVR（Twin Projection Support Vector Regression，TPSVR）. TPSVR 在目标函数中引入了一对新项，以找到训练样本的两个投影轴，可以更好地适应训练点. Chen 等人[60]提出一种鲁棒稀疏的 TSVR（Robust and Sparse TSVR，RSTSVR），引入正则化项和L_1范数增强鲁棒性和稀疏性. Peng 等人[61]提出基于L_1范数损失的 TSVR（L_1-norm Loss based TSVR，L_1-TSVR），其对偶问题中无须求逆矩阵. Peng 等人[62]又提出一种区间 TSVR（Interval TSVR，ITSVR），用于处理现实世界的区间数据. 该算法将 Hausdorff 距离作为间隔数据的间隔内核合并到高斯内核中. Parastalooi 等人[63]提出一种改进的 TSVR，首先对训练样本进行聚类，然后将每个样本在聚类中的密度引入原始问题的目标函数中. Gupta 等人[64]提出一种基于 Huber 损失的 TSVR（TSVR with Huber Loss，HN-TSVR）及其正则化版本（Regularization based TSVR，RHN-TSVR），较好地抑制了噪声与离群样本带来的负面影响，避免了奇异矩阵的出现，且使其对偶问题是凸函数，易于求解. Peng 等人[65]提出一种在原空间迭代求解的 TSVR（Primal TSVR，PTSVR）和一种使用简单的反向拟合策略来迭代选择其基本函数并更新增强矢量的稀疏 TSVR（Sparse TSVR，STSVR）. Singh 等人[66]提出一种简化的 TSVR（Reduced TSVR，RTSVR），利用矩形核以显著减少训练时间. Balasundaram 等人[67]提出ε不敏感 Huber 损失 TSVR. Singla 等人[68]将缩放的 Hinge 损失函数引入 TSVR 以提高模型的鲁棒性. López 等人[69]提出一种二阶锥规划（Second-order Cone Programming，SOCP）的鲁棒框架并将其扩展到 TSVR. Wang 等人[70]提出一种基于变分不等式（Variational Inequality，VI）的安全筛选规则以加快 TSVR 的训练速度，该方法被称为基于安全筛选规则 TSVR（Safe Screening Rule based TSVR，SSR-TSVR）. 该规则通过在求解 QPPs 之前无损地丢弃一些特征达到加速训练的效果. Chen 等人[71]针对 TSVR 提出光滑 TSVR（Smooth TSVR，STSVR）. STSVR 引入光滑函数以将 TSVR 的对偶问题转化为无约束优化问题，从而通过 Newton 法求解. Balasundaram 等人[72]在 TSVR 中引入正则化项、将不平滑的加函数替换为光滑近似函数或其广义导数来最小化结构化风险，并提出原空间的无约束 TSVR（Unconstrained TSVR Model in the Primal Space，UPTSVR）. Zhong 等人[73]将 TSVR 的目标函数改为线性规划问题，同时使用L_1范数距离，提高泛化性能，降低计算时间，提出线性规划 TSVR（Linear Programming TSVR，LPTSVR）. Niu 等人[74]使用 Huber 损失函数表达数据噪声和离群样本的多样性. 王岩等人[75]提出一种具有稀疏性的改进的稀疏 TSVR（Improved Sparse TSVR，ISTSVR），通过引入正则化项以及将训练样本的子集代替全部训练样本来使核函数稀疏，从而使算法具备稀疏性与鲁棒性. 黄华娟等人[76]提出一种光滑 CHKS TSVR（Smooth CHKS TSVR，SCTSVR），在文献[71]的基础上采用 CHKS 光滑函数代替 sigmoid 光滑函数以提高逼近精度. 沈葛亮等人[77]使用灰狼优化算法（Grey Wolf Optimization，GWO）优化 TSVR 的超参数. 郝运河等人[78]利用增量学习优化 TSVR，降低逆矩阵的计算量，提高训练速度. 黄华娟等人[79]利用 K.K.T. 互补条件将 TSVR 的目标函数转化为无约束优化问题，并引入极大熵函数以逼近其中存在的不可微项. 彭新俊等人[80]提出一种快速原空间 TSVR（Fast Primal TSVR，FPTSVR），在原始空间中直接采用 Newton 法优化 TSVR 的目标函数而不将其转化为对偶问题，避免性能损失. Wang 等人[81]提出一种提出了一种L_1-范数 TSVR 的特

征选择方法(A Feature Selection Method for L_1-norm TSVR, L_1-FTSVR)来自动选择输入数据中的重要特征．许颖春等人[82]用L_1范数将TSVR的两个原始二次规划问题转化为线性规划问题,提出了基于L_1范数的TSVR(L_1-TSVR). 陈素根等人[83]在TSVR的基础上构造了混合Hε损失函数,以自适应不同分布的噪声．

3.7.2 应用

Mehta等人[84]利用LTSVR提出了一种数字图像中的水印插入法．Meng等人[85]使用PSO优化TSVR的参数以估计蔗糖母液的纯度和过饱和度．Gupta等人[86]使用TSVR预测金融领域的时间序列数据．Hazarika等人[87]结合数据预处理和TSVR以及其他机器学习算法来预测河流中的沉积物负荷．Khemchandani等人[88]提出一种估计函数及其导数的TSVR(TSVR of a Function and its Derivatives),用于估计包括计算机图形学和图像处理等在内的应用问题的函数及其导数或偏导数．高闯等人使用文献[16]中提出的KNNWTSVR预测炼钢转炉终点碳质量分数和温度[89]. 麻芳兰等人[90]将正则化项引入TSVR并用其预测甘蔗收割机切割器入土切割时的负载压力．Wang等人[91]在孪生扩展SVR(Twin Extended SVR, T-X-SVR)的辅助下提出了一种虚拟模型架构框架．Cai等人[92]提出了一种改进的非线性STSVR(Improved Nonlinear STSVR, NSTSVR),并在两个功率放大器的发射机上对NSTSVR的建模效果和线性化能力进行了实验验证和分析．Chen等人[93]使用调整余弦相似度(Adjusted Cosine Similarity, ACS)方法筛选原始数据子序列并在该子序列上训练TSVR,用于预测现实世界收集的电力负载数据．Ding等人[94]将数据预处理、参数优化、深度学习等多种方法与TWSVR结合,进一步提高了改进后的模型预测国际各城市电力负载数据的能力．Feng等人[95]使用数据预处理方法和参数优化算法进一步提高TSVR的性能,并用于预测长江径流量．Gupta等人[96]调查了TSVR与其他几个替代方法在预测风速数据方面的能力．Li[97]使用TSVR预测天然气中的水合物形成温度(Hydrate Formation Temperature, HFT). 邱梦婵等人[98]使用核独立成分分析(Kernel Independent Component Analysis, KICA)提取样本特征,再用TSVR预测,加强了TSVR的除噪能力．尹湘锋等人[99]分别用线性核函数TSVR和多项式核函数TSVR预测股价走势,用遗传算法(Genetic Algorithm, GA)和PSO优化参数．汪淼等人[100]用TSVR预测80 t转炉炼钢实际生产数据中的终点碳质量分数和终点温度．

3.7.3 小结

从本节可得出结论,TSVR和TWSVR的其他改进算法主要围绕控制模型复杂度、避免过拟合、简化对偶问题的求解难度并找到其全局最优解、降低相关参数对模型的敏感程度、降低离群样本与噪声点对模型的干扰以及增强解的稀疏性等方面展开,从而增强模型的拟合能力、泛化性能以及鲁棒性,降低计算时间．在TSVR和TWSVR的其他应用方面,现有文献主要利用元启发式优化算法来优化模型的参数或对应用进行前处理,并采用数据预处理的方式分离数据中存在的噪声,增强模型的预测能力．同时,也可将TSVR和TWSVR与其他机器学习或深度学习模型相结合以进一步提升模型的整体性能．本文所涉及的各种改进的TSVR和TWSVR的优缺点整理见表1.

表1 各种改进的孪生支持向量回归机的优缺点

算法	优点	缺点
WTSVR	泛化能力强	无法自适应处理$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$的逆,目标函数不稀疏
LTSVR	求解时间短的同时未损失泛化能力	未考虑结构风险最小化、$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$的逆求解难、可能过拟合或陷入局部最优
LSTSVR	计算复杂度更小且训练速度更快,且不损失精度	受异常点影响较大
ν-TWSVR	自动调整参数以使精度最大化	受异常点影响较大
ε-TSVR	对偶问题稳定,训练速度较快	对存在大误差的数据集不鲁棒,受异常点影响较大
TPISVR	适合存在异方差噪声的数据	难以找到全局最优解

4 主要改进算法实验比较

本节对TSVR和TWSVR及上述的几个主要改进算法进行实验比较．为了保证实验的有效性与可靠性,实验中均采用十折交叉验证,运行环境为16 GB内存, i7-6700HQ处理器, 2.7 GHz主频, Windows 8操作系统．所有算法均在Matlab 2019a上实现和运行．在实验中,所有算法的核函数均采用高斯核函数(Radial Basis Function, RBF). 参数设置方面,所有参数的选择均采用网格搜索法,其中TPISVR中的参数ν的范围为$\{i\times 0.05|i=1,2,\cdots,10\}$,所有算法中的$\varepsilon$与$\nu$-TWSVR中的参数$\nu$的范围为$\{i\times 0.1|i=1,2,\cdots,10\}$,其他所有参数的范围均为$\{2^i|i=-5,-8,\cdots,19\}$. 为了节省模型选择参数的时间,每种算法中均令$C_1=C_2, \varepsilon_1=\varepsilon_2$. 所有数据在划分训练集与测试集之前均已进行归一化．本文所采用的UCI数据集的具体信息见表2.

所有算法在各数据集上的性能表现见表3. MAE表示平均绝对误差(Mean Absolute Error, MAE), RMSE表示均方根误差(Root Mean Square Error, RMSE), STD表

表2 数据集详细信息

数据集	样本数	样本维数
AutoMpg (AM)	392	6
Boston Housing (BH)	506	13
Machine CPU (MCPU)	209	6
Plastic	1 650	2
Qsar Fish Toxicity (QFT)	908	6
Real Estate Valuation (REV)	414	6
Servo	167	4
Wine Quality Red (WQR)	1 599	11
Yacht Hydrodynamics (YH)	308	6

示RMSE误差的标准差. 从表3中可以看出,在9个UCI数据集上,LSTSVR总体回归误差是最小的,在3个数据集上取得了最佳结果. TPISVR的回归误差总体而言较大. LTSVR和v-TWSVR分别在Plastic,WQR和MCPU,QFT数据集上获得了最小的回归误差,性能较为接近. WTSVR尽管未在任一数据集上取得最佳结果,但总体结果优于TPISVR. 表4展示了各算法的CPU计算时间,单位为s. 请注意,TSVR的目标函数使用Matlab内置的二次规划函数求解,其他算法的目标函数均使用其他求解器进行求解,因此其他算法的CPU时间较TSVR有显著增加,这是因为求解器的求解速度远没有Matlab内置函数的求解速度快.

从表4中可以看出,LSTSVR的计算时间明显较短,计算成本优势明显,在QFT,REV,Servo,YH等4个数据集上的计算时间已几乎趋近于0,主要原因是其二次规划问题可直接计算得到,无须求解. TSVR由于使用Maltab内置函数求解,耗时仅次于LSTSVR. 考虑到LSTSVR在3个数据集上取得了最佳回归结果,且计算成本几乎可忽略不计,可认为其回归表现是这7个算法中最优秀的. TPISVR和WTSVR耗时较长,且通过表3可得出其回归性能较差,可认为这两种算法在7个数据集上表现不佳.

根据表3和表4的实验结果以及算法的具体情况,可以得出更多分析. WTSVR为每个训练样本分配一个权重,虽然从直觉上提高了计算精度,但精度没有达到理想程度,反而增加了计算时间. LTSVR将TSVR中的2范数改为1范数,极大地提高了计算效率,且泛化性能几乎不受影响,但由于全局寻优能力较差,拟合结果较LSTSVR还有一定差距. LSTSVR采用了速度最快的最小二乘策略,取消了参数限制,因此计算复杂度是最低的,耗时也最少. 结合其回归结果与计算时间来看,其在3个数据集上性能已为最优,而且计算成本显著低于其他算法,因此其效率最高,且可适用于数据集规模非常大的场景. v-TWSVR作为和LTSVR并列次优的算法,一是得益于TWSVR拥有完备的数学基础,二是得益于自动调节参数变化. 同时,v-TWSVR的时间成本和LTSVR差不多,且都在可接受范围内,因此其也是一

表3 各算法回归结果比较

数据集	TSVR			WTSVR			LTSVR			LSTSVR			v-TWSVR			ε-TSVR			TPISVR		
	MAE	RMSE	STD	MAE	RMSE	STD	MAE	RMSE	STD	MAE	RMSE	STD	MAE	RMSE	STD	MAE	RMSE	STD	MAE	RMSE	STD
AM	0.062	0.087	0.028	0.067	0.098	0.021	0.057	0.080	0.020	**0.056**	**0.077**	**0.015**	0.078	0.105	0.020	0.074	0.102	0.021	0.184	0.218	0.028
BH	**0.055**	**0.074**	**0.010**	0.104	0.141	0.024	0.130	0.179	0.022	0.130	0.179	0.022	0.083	0.110	0.014	0.076	0.106	0.017	0.140	0.189	0.022
MCPU	0.077	0.121	0.096	0.050	0.097	0.112	0.064	0.118	0.113	0.064	0.118	0.113	**0.044**	**0.085**	**0.096**	0.045	0.086	0.101	0.067	0.111	0.097
Plastic	0.126	0.154	0.010	0.124	0.150	0.009	**0.119**	**0.149**	**0.010**	0.119	0.149	0.010	0.120	0.149	0.010	0.131	0.162	0.010	0.294	0.335	0.014
QFT	0.078	0.104	0.009	0.070	0.097	0.010	0.068	0.097	0.010	0.068	0.095	0.010	**0.067**	**0.094**	**0.009**	0.072	0.104	0.009	0.113	0.149	0.011
REV	0.078	0.100	0.012	0.055	0.075	0.016	0.048	0.067	0.009	**0.047**	**0.067**	**0.009**	0.049	0.068	0.010	0.052	0.073	0.012	0.112	0.134	0.012
Servo	0.046	0.070	0.021	0.112	0.134	0.025	0.047	0.069	0.022	**0.045**	**0.068**	**0.024**	0.104	0.146	0.032	0.094	0.118	0.032	0.127	0.178	0.029
WQR	0.102	0.128	0.008	0.113	0.139	0.012	**0.099**	**0.127**	**0.011**	0.099	0.128	0.011	0.112	0.142	0.011	0.112	0.139	0.010	0.134	0.158	0.010
YH	0.059	0.071	0.010	0.093	0.146	0.050	0.113	0.145	0.028	0.110	0.147	0.034	0.186	0.239	0.063	**0.075**	**0.119**	**0.035**	0.177	0.239	0.069

表4 各算法CPU计算时间比较

数据集	TSVR	WTSVR	LTSVR	LSTSVR	v-TWSVR	ε-TSVR	TPISVR
AM	3.093 8	114.046 9	52.656 3	0.078 1	50.687 5	106.531 3	78.828 1
BH	4.843 8	206.671 9	62.718 8	0.015 6	190.640 6	192.875 0	113.328 1
MCPU	1.171 9	30.077 0	27.315 8	0.046 8	27.939 8	27.440 6	28.922 6
Plastic	55.265 6	104.328 1	441.859 4	0.062 5	189.562 5	254.968 8	194.937 5
QFT	14.687 5	81.479 3	70.637 3	0.000 0	184.720 8	180.836 4	28.095 8
REV	3.046 9	46.769 1	24.944 6	0.000 0	40.544 7	40.092 3	28.111 4
Servo	1.015 6	27.393 8	23.041 3	0.000 0	20.685 7	25.287 8	24.211 4
WQR	55.703 1	290.437 5	415.484 4	0.062 5	358.937 5	359.375 0	185.828 1
YH	1.656 3	24.866 6	20.872 9	0.000 0	17.643 7	25.833 8	20.826 1

个较优秀的算法. ε-TSVR和TSVR各自仅在1个数据集上取得最优结果,但由于ε-TSVR计算时间较长,而TSVR可用Matlab内置函数求解,计算时间短,因此TSVR略优于ε-TSVR. TPISVR对其参数不敏感,因此在处理普通数据集时效果较差,较适合处理异方差噪声数据.

5 总结与展望

TSVR和TWSVR作为机器学习领域的一个较新的理论,其数学模型与算法思想都尚不成熟,存在许多待改进的空间,另外,在现实生活中,需要精确拟合曲线、预测时序数据的场景普遍存在,TSVR和TWSVR作为计算速度4倍于传统SVR的算法,其潜力更大,可应用场景更为广泛. 因此,TSVR和TWSVR的改进是机器学习领域研究中的一个重要内容. 本文总结了近年来围绕TSVR和TWSVR所产生的改进算法. 已有的改进策略主要可分为加权法、拉格朗日法、最小二乘法、基于ν的方法、基于ε的方法以及参数不敏感法,涵盖训练样本加权、对偶问题中存在不可逆矩阵的处理、训练样本稀疏化、引入正则化项、不依赖参数、对偶问题非严格凸或不可微的处理等多种改进策略. 同时注意到基于TSVR和TWSVR的应用较少,仅有的应用仅停留在单纯使用或参数优化上,因此TSVR和TWSVR的应用还需要进一步完善. 此外,TSVR和TWSVR的性能还有进一步提升与优化的空间. 本文认为,还需要在以下几个研究方向上对TSVR和TWSVR加以完善和改进.

(1)训练样本权值的精细化设计

离群样本与噪声对超平面的影响非常大,是影响TSVR和TWSVR性能的一个重要因素. 现有的权重设计往往存在过度惩罚问题. 如何为样本分配合适的权重,如对离群样本与噪声分配较小权重以抵消其影响,对超出上/下边界函数的样本点根据其所在的位置对两个超平面产生的影响作精确的权重配置,是接下来要研究的一个重要内容.

(2)原问题的精确求解

目前,大多数TSVR和TWSVR的改进形式都通过其对偶问题求解. 但对偶问题在其非严格凸、不可微、半正定矩阵无逆矩阵等三种情况下都会直接导致回归器其无法直接求解或无法取得较为精确的解. 实际上,对偶问题取得的解本身就是近似解. 因此如何得到原问题的精确解是另一个亟待解决的重要内容.

(3)引入多分类孪生支持向量回归机

多分类孪生支持向量机(Multi class Twin Support Vector Machines, MTWSVMs)是近年来的一个研究热点. 对应地,MTWSVMs也有其回归形式. 另外,多类别形式是现实世界数据集最常见的分布形式,而孪生支持向量回归机只假定训练集被分为两类,这不符合现实世界数据集的分布. 因此,研究多超平面孪生支持向量回归机,并结合聚类算法为样本点实现预选取合适的标签,再通过多超平面孪生支持向量回归机进行拟合与预测,也是一个重要内容.

(4)拓宽孪生支持向量回归机的应用领域

目前,对TSVR和TWSVR的研究主要集中在理论研究上,应用研究尚未得到深入而广泛的探索. TSVR和TWSVR可以应用于风速预测、股价预测、河流中悬浮泥沙负荷模拟、岩石抗压强度、国际汇率预测、通货膨胀以及电力负荷预测等多个领域中.

参考文献

[1] CORTES C, VAPNIK V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[2] DRUCKER H, BURGES C J C, KAUFMAN L, SMOLA A, VAPNIK V. Support vector regression machines[C]// Proceedings of the 9th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 1996: 155-161.

[3] PENG X J. TSVR: An efficient twin support vector machine for regression[J]. Neural Networks, 2010, 23(3): 365-372.

[4] LIEPERT M. Topological fields chunking for German with SVM's: Optimizing SVM-parameters with GA's[C]// Proceedings of the International Conference on Recent Advances in Natural Language Processing. INCOMA Ltd.: Borovets, 2003: 1-5.

[5] DONG J X, KRZYZAK A, SUEN C Y. Fast SVM training algorithm with decomposition on very large data sets[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(4): 603-618.

[6] SHEVADE S K, KEERTHI S S, BHATTACHARYYA C, et al. Improvements to the SMO algorithm for SVM regression[J]. IEEE Transactions on Neural Networks, 2000, 11(5): 1188-1193.

[7] JAYADEVA, KHEMCHANDANI R, CHANDRA S. Twin support vector machines for pattern classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(5): 905-910.

[8] 丁世飞, 张健, 张谢锴, 等. 多分类孪生支持向量机研究进展[J]. 软件学报, 2018, 29(1): 89-108.
DING S F, ZHANG J, ZHANG X K, et al. Survey on multi class twin support vector machines[J]. Journal of Software, 2018, 29(1): 89-108. (in Chinese)

[9] BI J B, BENNETT K P. A geometric approach to support vector regression[J]. Neurocomputing, 2003, 55(1-2):

79-108.

[10] KHEMCHANDANI R, GOYAL K, CHANDRA S. TWSVR: Regression via twin support vector machine[J]. Neural Networks, 2016, 74: 14-21.

[11] JAYADEVA, KHEMCHANDANI R, CHANDRA S. TWSVR: Twin support vector machine based regression [M]//Twin Support Vector Machines. Berlin: Springer Cham, 2017: 63-101.

[12] HOUSSEIN E H. Particle swarm optimization-enhanced twin support vector regression for wind speed forecasting [J]. Journal of Intelligent Systems, 2019, 28(5): 905-914.

[13] 张冰, 王传美, 贺素香. 改进的TSVR模型在股市高频数据上的预测[J]. 计算机工程与设计, 2019, 40(11): 3241-3246.

ZHANG B, WANG C M, HE S X. Prediction of improved TSVR model on high frequency stock market[J]. Computer Engineering and Design, 2019, 40(11): 3241-3246. (in Chinese)

[14] WANG L D, MA Y M, CHANG X D, et al. Projection wavelet weighted twin support vector regression for OFDM system channel estimation[J]. Artificial Intelligence Review, 2021, 54(1): 469-489.

[15] XU Y T, WANG L S. A weighted twin support vector regression[J]. Knowledge-Based Systems, 2012, 33: 92-101.

[16] XU Y T, WANG L S. K-nearest neighbor-based weighted twin support vector regression[J]. Applied Intelligence, 2014, 41(1): 299-309.

[17] TANVEER M, SHUBHAM K, ALDHAIFALLAH M, et al. An efficient regularized K-nearest neighbor based weighted twin support vector regression[J]. Knowledge-Based Systems, 2016, 94: 70-87.

[18] GUPTA D. Training primal K-nearest neighbor based weighted twin support vector regression via unconstrained convex minimization[J]. Applied Intelligence, 2017, 47(3): 962-991.

[19] 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.

[20] 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.

[21] 程昊翔, 王坚. 密度加权孪生支持向量回归机[J]. 控制与决策, 2016, 31(4): 755-758.

CHENG H X, WANG J. Density-weighted twin support vector regression[J]. Control and Decision, 2016, 31(4): 755-758. (in Chinese)

[22] 徐奔业, 顾斌杰, 潘丰, 等. 加权光滑投影孪生支持向量回归机[J/OL]. 计算机工程, 2022: 1-10. DOI: 10.19678/j.issn.1000-3428.0063542.

XU B Y, GU B J, PAN F, et al. Weighted smooth projection twin support vector regression[J]. Computer Engineering, 2022: 1-10. DOI:10.19678/j.issn.1000-3428.0063542. (in Chinese)

[23] 李朔, 雷为民, 张伟. 基于KNN-TSVR算法的MIMO-OFDM系统信道估计[J]. 东北大学学报(自然科学版), 2022, 43(2): 176-181, 242.

LI S, LEI W M, ZHANG W. Channel estimation for MIMO-OFDM system based on KNN-TSVR algorithm[J]. Journal of Northeastern University (Natural Science), 2022, 43(2): 176-181, 242. (in Chinese)

[24] 马艺梅, 王立东, 陈雪波, 等. 基于小波加权TSVR算法的OFDM系统信道估计[J]. 小型微型计算机系统, 2020, 41(7): 1433-1437.

MA Y M, WANG L D, CHEN X B, et al. OFDM system channel estimation based on wavelet weighted TSVR algorithm[J]. Journal of Chinese Computer Systems, 2020, 41(7): 1433-1437. (in Chinese)

[25] BALASUNDARAM S, TANVEER M. On Lagrangian twin support vector regression[J]. Neural Computing and Applications, 2013, 22(1): 257-267.

[26] 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.

[27] TANVEER M, SHUBHAM K, ALDHAIFALLAH M, et al. An efficient implicit regularized Lagrangian twin support vector regression[J]. Applied Intelligence, 2016, 44(4): 831-848.

[28] GUPTA D, GUPTA U. On robust asymmetric Lagrangian ν-twin support vector regression using pinball loss function[J]. Applied Soft Computing, 2021, 102: 107099.

[29] BALASUNDARAM S, GUPTA D. Training Lagrangian twin support vector regression via unconstrained convex minimization[J]. Knowledge-Based Systems, 2014, 59: 85-96.

[30] GANAIE MA, TANVEER M, BEHESHTI I. Brain age prediction using improved twin SVR[J]. Neural Computing and Applications, 2022: 1-11.

[31] ZHAO Y P, ZHAO J, ZHAO M. Twin least squares support vector regression[J]. Neurocomputing, 2013, 118: 225-236.

[32] HUANG H J, DING S F, SHI Z Z. Primal least squares twin support vector regression[J]. Journal of Zhejiang University SCIENCE C, 2013, 14(9): 722-732.
[33] ZHANG Z Q, LV T L, WANG H, et al. A novel least square twin support vector regression[J]. Neural Processing Letters, 2018, 48(2): 1187-1200.
[34] HUANG H J, WEI X X, ZHOU Y Q. A sparse method for least squares twin support vector regression[J]. Neurocomputing, 2016, 211: 150-158.
[35] 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.
[36] 王怡芮, 朱志祥. 模糊最小二乘大间隔孪生支持向量回归机[J]. 计算机与数字工程, 2020, 48(6): 1275-1280.
WANG Y R, ZHU Z X. Fuzzy twin least squares large margin distribution support vector regression[J]. Computer & Digital Engineering, 2020, 48(6): 1275-1280. (in Chinese)
[37] 曹杰, 顾斌杰, 熊伟丽, 等. 增量式约简最小二乘孪生支持向量回归机[J]. 计算机科学与探索, 2021, 15(3): 553-563.
CAO J, GU B J, XIONG W L, et al. Incremental reduced least squares twin support vector regression[J]. Journal of Frontiers of Computer Science and Technology, 2021, 15(3): 553-563. (in Chinese)
[38] 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.
[39] ZHANG S G, LIU C, ZHOU T, et al. Twin least squares support vector regression of heteroscedastic Gaussian noise model[J]. IEEE Access, 8: 94076-94088.
[40] YAN H, QI Y, YE Q L, et al. Robust least squares twin support vector regression with adaptive FOA and PSO for short-term traffic flow prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(9): 14542-14556.
[41] WANG S Y. The prediction method of KPIs by using LS-TSVR[C]//2022 7th International Conference on Big Data Analytics (ICBDA). Guangzhou: IEEE, 2022: 177-180.
[42] ZHANG S G, YUAN Q Y, YUAN F, et al. Twin proximal least squares support vector regression machine based on heteroscedastic Gaussian noise[J/OL]. Journal of Intelligent & Fuzzy Systems, 2022: 1-15. (2022-07-13) [2022-08-17]. https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs211631.
[43] HAZARIKA B B, GUPTA D, NATARAJAN N. Wavelet kernel least square twin support vector regression for wind speed prediction[J/OL]. Environmental Science and Pollution Research International, 2022: 1-17. (2022-01-24) [2022-08-17]. https://pubmed.ncbi.nlm.nih.gov/35067890/.
[44] SHAO Y H, ZHANG C H, YANG Z M, et al. An ε-twin support vector machine for regression[J]. Neural Computing and Applications, 2013, 23(1): 175-185.
[45] PENG X J. A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms[J]. Information Sciences, 2010, 180(20): 3863-3875.
[46] RASTOGI R, ANAND P, CHANDRA S. A ν-twin support vector machine based regression with automatic accuracy control[J]. Applied Intelligence, 2017, 46(3): 670-683.
[47] XU Y T, LI X Y, PAN X L, et al. Asymmetric ν-twin support vector regression[J]. Neural Computing and Applications, 2018, 30(12): 3799-3814.
[48] GUPTA U, GUPTA D. An improved regularization based Lagrangian asymmetric ν-twin support vector regression using pinball loss function[J]. Applied Intelligence, 2019, 49(10): 3606-3627.
[49] XUE Z X, ZHANG R, QIN C D, et al. A rough ν-twin support vector regression machine[J]. Applied Intelligence, 2018, 48(11): 4023-4046.
[50] WANG L D, GAO C, ZHAO N N, et al. Wavelet transform-based weighted ν-twin support vector regression[J]. International Journal of Machine Learning and Cybernetics, 2020, 11(1): 95-110.
[51] LI S, LEI W M, ZHANG W. ν-Twin SVR based OFDM system wireless channel estimation[C]//International Conference on Intelligent Automation and Soft Computing. Chicago: Springer, 2022: 973-978.
[52] YE Y F, CAO H, BAI L, et al. Exploring determinants of inflation in China based on L1-ϵ-twin support vector regression[J]. Procedia Computer Science, 2013, 17: 514-522.
[53] YE Y F, BAI L, HUA X Y, et al. Weighted Lagrange ε-twin support vector regression[J]. Neurocomputing, 2016, 197: 53-68.
[54] PENG X J. Efficient twin parametric insensitive support vector regression model[J]. Neurocomputing, 2012, 79: 26-38.
[55] GUPTA D, ACHARJEE K, RICHHARIYA B. Lagrangian twin parametric insensitive support vector regression (LTPISVR) [J]. Neural Computing and Applications,

2020, 32(10): 5989-6007.

[56] GUPTA D, RICHHARIYA B. Efficient implicit Lagrangian twin parametric insensitive support vector regression via unconstrained minimization problems[J]. Annals of Mathematics and Artificial Intelligence, 2021, 89: 301-332.

[57] 丁世飞, 黄华娟. 最小二乘孪生参数化不敏感支持向量回归机[J]. 软件学报, 2017, 28(12): 3146-3155.
DING S F, HUANG H J. Least squares twin parametric insensitive support vector regression[J]. Journal of Software, 2017, 28(12): 3146-3155. (in Chinese)

[58] 黄华娟, 韦修喜, 周永权. 光滑孪生参数化不敏感支持向量回归机[J]. 郑州大学学报(工学版), 2022, 43(2): 28-34.
HUANG H J, WEI X X, ZHOU Y Q. Smooth twin parametric insensitive support vector regression[J]. Journal of Zhengzhou University (Engineering Science), 2022, 43(2): 28-34. (in Chinese)

[59] PENG X J, XU D, SHEN J D. A twin projection support vector machine for data regression[J]. Neurocomputing, 2014, 138: 131-141.

[60] CHEN X B, YANG J, CHEN L. An improved robust and sparse twin support vector regression via linear programming[J]. Soft Computing, 2014, 18(12): 2335-2348.

[61] PENG X J, CHEN D. An l1-norm loss based twin support vector regression and its geometric extension[J]. International Journal of Machine Learning and Cybernetics, 2019, 10(9): 2573-2588.

[62] PENG X J, CHEN D J, KONG L Y, et al. Interval twin support vector regression algorithm for interval input-output data[J]. International Journal of Machine Learning and Cybernetics, 2015, 6(5): 719-732.

[63] PARASTALOOI N, AMIRI A, ALIHEIDARI P. Modified twin support vector regression[J]. Neurocomputing, 2016, 211: 84-97.

[64] GUPTA U, GUPTA D. On regularization based twin support vector regression with Huber loss[J]. Neural Processing Letters, 2021, 53(1): 459-515.

[65] PENG X J. Primal twin support vector regression and its sparse approximation[J]. Neurocomputing, 2010, 73(16-18): 2846-2858.

[66] SINGH M, CHADHA J, AHUJA P, et al. Reduced twin support vector regression[J]. Neurocomputing, 2011, 74(9): 1474-1477.

[67] BALASUNDARAM S, PRASAD S C. Robust twin support vector regression based on Huber loss function[J]. Neural Computing and Applications, 2020, 32: 11285-11309.

[68] SINGLA M, GHOSH D, SHUKLA K K, et al. Robust twin support vector regression based on rescaled Hinge loss[J]. Pattern Recognition, 2020, 105: 107395.

[69] LÓPEZ J, MALDONADO S. Robust twin support vector regression via second-order cone programming[J]. Knowledge-Based Systems, 2018, 152: 83-93.

[70] WANG H M, XU Y T. Scaling up twin support vector regression with safe screening rule[J]. Information Sciences, 2018, 465: 174-190.

[71] CHEN X B, YANG J, LIANG J, et al. Smooth twin support vector regression[J]. Neural Computing and Applications, 2012, 21(3): 505-513.

[72] BALASUNDARAM S, MEENA Y. Training primal twin support vector regression via unconstrained convex minimization[J]. Applied Intelligence, 2016, 44(4): 931-955.

[73] ZHONG P, XU Y T, ZHAO Y H. Training twin support vector regression via linear programming[J]. Neural Computing and Applications, 2012, 21(2): 399-407.

[74] NIU J Y, CHEN J, XU Y T. Twin support vector regression with Huber loss[J]. Journal of Intelligent & Fuzzy Systems, 2017, 32(6): 4247-4258.

[75] 王岩, 朱齐丹, 刘志林, 等. 改进的稀疏孪生支持向量回归算法[J]. 系统工程与电子技术, 2012, 34(9): 1940-1945.
WANG Y, ZHU Q D, LIU Z L, et al. Improved sparse twin support vector regression algorithm[J]. Systems Engineering and Electronics, 2012, 34(9): 1940-1945. (in Chinese)

[76] 黄华娟, 丁世飞, 史忠植. 光滑CHKS孪生支持向量回归机[J]. 计算机研究与发展, 2015, 52(3): 561-568.
HUANG H J, DING S F, SHI Z Z. Smooth CHKS twin support vector regression[J]. Journal of Computer Research and Development, 2015, 52(3): 561-568. (in Chinese)

[77] 沈葛亮, 顾斌杰, 潘丰. 基于灰狼优化算法的孪生支持向量回归机[J]. 南京理工大学学报, 2020, 44(2): 202-208.
SHEN G L, GU B J, PAN F. Twin support vector regression based on grey wolf optimization algorithm[J]. Journal of Nanjing University of Science and Technology, 2020, 44(2): 202-208. (in Chinese)

[78] 郝运河, 张浩峰. 基于双支持向量回归机的增量学习算法[J]. 计算机科学, 2016, 43(2): 230-234, 249.
HAO Y H, ZHANG H F. Incremental learning algorithm based on twin support vector regression[J]. Computer Science, 2016, 43(2): 230-234, 249. (in Chinese)

[79] 黄华娟, 韦修喜. 基于自适应调节极大熵的孪生支持向量回归机[J]. 南京大学学报(自然科学), 2019, 55(6):

1030-1039.
HUANG H J, WEI X X. Twin support vector regression based on adaptive adjustment maximum entropy[J]. Journal of Nanjing University (Natural Science), 2019, 55(6): 1030-1039. (in Chinese)
[80] 彭新俊, 王翼飞. 快速原空间孪生支持向量回归算法[J]. 模式识别与人工智能, 2011, 24(1): 22-29.
PENG X J, WANG Y F. Fast twin support vector regression algorithm in primal space[J]. Pattern Recognition and Artificial Intelligence, 2011, 24(1): 22-29. (in Chinese)
[81] WANG F, WU Q, FU Y L. Feature selection method for L1-norm twin support vector regression[C]//2021 International Conference on Control, Automation and Information Sciences. Xi'an: IEEE, 2021: 668-673.
[82] 许颖春, 范丽亚. 基于L_1-范数的非线性TSVR[J]. 聊城大学学报(自然科学版), 2017, 30(3): 6-11.
XU Y C, FAN L Y. Nonlinear TSVR based on L_1-norm [J]. Journal of Liaocheng University (Natural Science Edition), 2017, 30(3): 6-11. (in Chinese)
[83] 陈素根, 石婷. 新型鲁棒孪生支持向量回归机[J/OL]. 计算机科学与探索, 2021: 1-13. (2021-10-18)[2022-08-17]. https://kns. cnki. net/kcms/detail/11.5602. tp. 20211014.1714.004. html.
CHEN S G, SHI T. A novel robust twin support vector regression[J/OL]. Journal of Frontiers of Computer Science and Technology, 2021: 1-13. (2021-10-18) [2022-08-17]. https://kns.cnki.net/kcms/detail/11.5602.tp.20211014.1714.004. html. (in Chinese)
[84] MEHTA R, GUPTA K, YADAV A K. An adaptive framework to image watermarking based on the twin support vector regression and genetic algorithm in lifting wavelet transform domain[J]. Multimedia Tools and Applications, 2020, 79: 18657-18678.
[85] MENG Y M, LAN Q L, QIN J, et al. Data-driven soft sensor modeling based on twin support vector regression for cane sugar crystallization[J]. Journal of Food Engineering, 2019, 241: 159-165.
[86] GUPTA D, PRATAMA M, MA Z Y, et al. Financial time series forecasting using twin support vector regression[J]. PLoS One, 2019, 14(3): e0211402.
[87] HAZARIKA B B, GUPTA D, BERLIN M. Modeling suspended sediment load in a river using extreme learning machine and twin support vector regression with wavelet conjunction[J]. Environmental Earth Sciences, 2020, 79: 1-15.
[88] KHEMCHANDANI R, KARPATNE A, CHANDRA S. Twin support vector regression for the simultaneous learning of a function and its derivatives[J]. International Journal of Machine Learning and Cybernetics, 2013, 4(1): 51-63.
[89] 高闯, 沈明钢, 王焕清. 基于孪生支持向量回归机的转炉炼钢终点预测[J]. 中国冶金, 2019, 29(4): 12-16.
GAO C, SHEN M G, WANG H Q. End-point prediction of BOF steelmaking based on twin support vector regression [J]. China Metallurgy, 2019, 29(4): 12-16. (in Chinese)
[90] 麻芳兰, 罗晓虎, 李科, 等. 基于正则化TSVR的甘蔗收割机切割器入土切割负载压力预测研究[J]. 中国农机化学报, 2021, 42(2): 8-14, 24.
MA F L, LUO X H, LI K, et al. Prediction of load pressure of cutter of sugarcane harvester based on regularized TSVR[J]. Journal of Chinese Agricultural Mechanization, 2021, 42(2): 8-14, 24. (in Chinese)
[91] WANG Q H, WU D, LI G Y, et al. A virtual model architecture for engineering structures with twin extended support vector regression (T-X-SVR) method[J]. Computer Methods in Applied Mechanics and Engineering, 2021, 386: 114121.
[92] CAI T F, LI M Y, YAO Y, et al. An improved nonlinear smooth twin support vector regression based-behavioral model for joint compensation of frequency-dependent transmitter nonlinearities[J]. International Journal of RF and Microwave Computer-Aided Engineering, 2021, 31 (6): e22636.
[93] CHEN X, WU Y F, LIAO Q Y, et al. An improved twin support vector regression machine based on adjusted cosine similarity for load prediction[C]//2021 China International Conference on Electricity Distribution. Shanghai: IEEE, 2021: 611-614.
[94] DING S F, ZHANG Z C, GUO L L, et al. An optimized twin support vector regression algorithm enhanced by ensemble empirical mode decomposition and gated recurrent unit[J]. Information Sciences, 2022, 598: 101-125.
[95] FENG Z K, NIU W J, WAN X Y, et al. Hydrological time series forecasting via signal decomposition and twin support vector machine using cooperation search algorithm for parameter identification[J]. Journal of Hydrology, 2022, 612: 128213.
[96] GUPTA D, NATARAJAN N, BERLIN M. Short-term wind speed prediction using hybrid machine learning techniques[J]. Environmental Science and Pollution Research International, 2022, 29(34): 50909-50927.
[97] LI C Z. Twin support vector regression for prediction of natural gas hydrate formation conditions[J]. Industrial &

Engineering Chemistry Research, 2021, 60(50): 18519-18529.

[98] 邱梦婵, 苏成利, 钟国财, 等. 基于KICA-TSVR方法乙烯裂解产物分布软测量[J]. 控制工程, 2015, 22(3): 458-464.
QIU M C, SU C L, ZHONG G C, et al. The soft measuring yield of pyrolysis product twin support vector regression machine based on KICA[J]. Control Engineering of China, 2015, 22(3): 458-464. (in Chinese)

[99] 尹湘锋, 崔浩锋, 文雪婷. 基于两类核函数的TSVR在股价预测中的比较[J]. 统计与决策, 2021, 37(12): 43-46.
YIN X F, CUI H F, WEN X T. Comparison of TSVR based on two kinds of kernel functions in stock price forecasting[J]. Statistics & Decision, 2021, 37(12): 43-46. (in Chinese)

[100] 汪淼, 李胜利, 高闯, 等. 80t转炉终点预报TSVR模型精度[J]. 钢铁, 2020, 55(7): 53-57.
WANG M, LI S L, GAO C, et al. End-point prediction TSVR model accuracy of 80t BOF steelmaking[J]. Iron & Steel, 2020, 55(7): 53-57. (in Chinese)

作者简介

丁世飞 男，1963年1月生，山东青岛人．博士、教授、博士生导师．主要研究方向为人工智能、机器学习、深度学习、模式识别等．

E-mail: dingsf@cumt.edu.cn

张子晨（通讯作者） 男，1992年8月生，江苏徐州人．现为中国矿业大学计算机应用技术专业博士研究生．主要研究方向为支持向量机、机器学习．

E-mail: 452871681@qq.com