软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(12):3146−3155 [doi: 10.13328/j.cnki.jos.005240]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

最小二乘孪生参数化不敏感支持向量回归机*

丁世飞[1,2], 黄华娟[1,2,3]

[1](中国矿业大学 计算机科学与技术学院,江苏 徐州 221116)

[2](中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100190)

[3](广西民族大学 信息科学与工程学院,广西 南宁 530006)

通讯作者: 黄华娟, E-mail:hhj-025@163.com

摘 要: 孪生参数化不敏感支持向量回归机(twin parametric insensitive support vector regression,简称 TPISVR)是一种新型机器学习方法.与其他回归方法相比,TPISVR 在处理异方差噪声方面具有独特的优势.标准 TPISVR 的训练算法可以归结为在对偶空间求解一对具有不等式约束的二次规划问题.然而,这种求解方法的时间消耗比较大.引入最小二乘思想,将 TPISVR 的两个二次规划问题转化为两个线性方程组,并在原始空间上直接求解,提出了最小二乘孪生参数化不敏感支持向量回归机(least squares TPISVR,简称 LSTPISVR).为了解决 LSTPISVR 的参数选择问题,提出了混沌布谷鸟优化算法,并用其对 LSTPISVR 的参数进行优化选择.在人工数据集和 UCI 数据集上的实验结果表明:LSTPISVR 在保持精度不下降的情况下,具有更高的运行效率.

关键词: 孪生参数化不敏感支持向量回归机;异方差性;最小二乘;混沌布谷鸟优化算法

中图法分类号: TP181

中文引用格式: 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.软件学报,2017,28(12):3146−3155. http://www.jos.org.cn/1000-9825/5240.htm

英文引用格式: Ding SF, Huang HJ. Least squares twin parametric insensitive support vector regression. Ruan Jian Xue Bao/ Journal of Software, 2017,28(12):3146−3155 (in Chinese). http://www.jos.org.cn/1000-9825/5240.htm

Least Squares Twin Parametric Insensitive Support Vector Regression

DING Shi-Fei[1,2], HUANG Hua-Juan[1,2,3]

[1](School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

[2](Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, The Chinese Academy of Science, Beijing 100190, China)

[3](College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China)

Abstract: Twin parametric insensitive support vector regression (TPISVR) is a novel machine learning method proposed. Compared to other regression methods, TPISVR has unique advantages in dealing with heteroscedastic noise. Standard TPISVR can be attributed to solve a pair of quadratic programming problem (QPP) with inequality constraints in the dual space. However, this method is subject to the constraints of time and memory when number of samples are large. This paper introduces the least squares ideas, and proposes the least squares twin parametric insensitive support vector regression (LSTPISVR) which transforms the two QPPs of TPISVR into linear equations and solves them directly on the original space. Further, a chaotic cuckoo optimization algorithm is introduced for parameter selection of LSTPISVR. Experiments on artificial datasets and UCI datasets show that LSTPISVR not only has fast learning speed, but also shows good generalization performance.

* 基金项目: 国家自然科学基金(61379101, 61662005, 61672522); 国家重点基础研究发展计划(973)(2013CB329502)

Foundation item: National Natural Science Foundation of China (61379101, 61662005, 61672522); National Basic Research Program of China (973) (2013CB329502)

收稿时间: 2016-01-10; 修改时间: 2016-05-26, 2016-10-08; 采用时间: 2016-11-04; jos 在线出版时间: 2017-03-24

CNKI 网络优先出版: 2017-03-24 12:34:09, http://kns.cnki.net/kcms/detail/11.2560.TP.20170324.1234.001.html

Key words: twin parametric insensitive support vector regression; heteroscedastic; least squares; chaotic cuckoo optimization algorithm

支持向量机(support vector machine,简称 SVM)是由 Vapnik 等人提出的基于统计学习理论的机器学习方法[1],它不仅是一种小样本学习方法,也是一种基于结构风险最小化原则的方法.与人工神经网络相比[2,3],SVM 成功解决了高维问题和局部极小值问题,因此具有更好的泛化能力.目前,SVM 已成功应用到模式识别[4]、时间序列预测[5]、文本分类[6]和图像处理[7]等多个领域.

虽然 SVM 取得了较好的学习性能,但是其训练时间非常高,达到 $O(l^3)$,其中,l 是整个训练集的样本数.为了提高 SVM 的训练速度,学者们已经提出了多种改进算法,比如选块算法(chunking algorithm)[8]、分解算法(decomposition algorithm)[9]、序列最小优化算法(sequential minimal optimization,简称 SMO)[10]等等.这些经典的改进方法虽然在一定程度上提高了 SVM 的学习性能,但算法比较复杂,实现上有一定难度.近年来,学者们开始研究基于标准 SVM 的变形算法[11,12].2007 年,Jayadeva 等人[13]在深入研究了标准 SVM 形式的基础上,提出了孪生支持向量机(twin support vector machines,简称 TWSVM).TWSVM 是要得到两个不平行的分类超平面,使得每一个超平面靠近其中的一类样本,而远离另一类样本.TWSVM 在形式上类似于标准 SVM,但其计算效率是 SVM 的 4 倍.鉴于其明显的分类优势,TWSVM 已被应用于说话人识别、医学检测等领域[14−17].在回归问题求解方面,2010 年,基于 TWSVM 的思想,Peng 等人[18]提出了孪生支持向量回归机(twin support vector regression,简称 TSVR).TSVR 产生一对超平面,分别确定目标回归函数的ε不敏感上、下界.TSVR 仅需要求解一对较小规模的二次规划问题(quadratic programming problem,简称 QPP),每个 QPP 所含约束条件的数目仅为传统支持向量回归机(support vector regression,简称 SVR)的一半,并且 TSVR 的对偶问题中没有等式约束,这使得 TSVR 的训练速度大为提高[19].然而,TSVR 丧失了稀疏性,其预测速度比 SVR 的慢.

目前,大部分有关 TSVR 和 SVR 的学习算法都是基于假设样本噪声在整个区域是一致的或者函数依赖事先已知的前提下提出的,而在实际应用中,这种假设不一定成立,比如,我们经常会碰到异方差噪声,这种噪声依赖于区域的位置.为了解决这种问题,2010 年,Hao 等人[20]引入一种参数化不敏感损失函数,提出了参数化不敏感支持向量回归机(par-v-SVR).与 SVR 相比,par-v-SVR 更适合于求解异方差噪声问题.而且,与 SVR 一样,par-v-SVR 的训练速度也不够理想.2012 年,Peng[21]为了提高 par-v-SVR 的训练速度,结合 TSVR 和 par-v-SVR 的思想,提出了孪生参数化不敏感支持向量回归机(twin parametric insensitive support vector regression,简称 TPISVR).TPISVR 要产生两个不平行的函数,分别确定目标回归函数的参数化不敏感上、下界.理论分析和实验结果表明:与 par-v-SVR 相比,TPISVR 在保证精度不下降的情况下获得了更快的训练速度.TPISVR 的标准算法也是在对偶空间求解两个二次规划问题,然而对于样本数目较大的问题,这种求解方法将受到时间和内存的制约.

本文引入最小二乘思想,将 TPISVR 的两个二次规划问题转化为两个线性方程组,并在原空间上直接求解,提出了最小二乘孪生参数化不敏感支持向量回归机(least squares TPISVR,简称 LSTPISVR).与 TPISVR 一样,LSTPISVR 至少存在 4 个参数.为了解决 PLTPISVR 的参数选择问题,本文提出具有较强寻优能力的混沌布谷鸟优化算法,并用其对 LSTPISVR 的参数进行优化选择.在人工数据集和 UCI 数据集上的实验结果表明:LSTPISVR 在保持精度不下降的情况下,比 TPISVR 具有更高的运行效率.

1 TPISVR 基本理论

给定训练集$\{(x_1,y_1),\ldots,(x_l,y_l)\}\in R^n\times R,i=1,\ldots,l$.令 $A_{l\times n}$ 为训练样本输入数据集,即 $\{x_k\}_{k=1}^{l}$;令 $Y_{l\times 1}$ 为训练样本输出数据集,即 $A_{l\times n}$ 对应的回归为 $Y_{l\times 1}=[y_1,y_2,\ldots,y_l]^T$.

先简要地回顾一下 TPISVR 算法.TPISVR 的目标是,通过训练数据集以获得如下的一对不平行函数:

$$f_1(x)=w_1^Tx+b_1,f_2(x)=w_2^Tx+b_2 \tag{1}$$

分别确定回归函数的不敏感下、上界.而这对函数的确定,可以通过求解下面的一对二次规划问题:

$$\left.\begin{aligned}&\min\frac{1}{2}\|w_1\|^2-\frac{v_1}{l}e^T(Aw_1+b_1e)+\frac{c_1}{l}e^T\xi\\&\text{s.t. } Y\geqslant Aw_1+b_1e-\xi,\xi\geqslant 0\end{aligned}\right\} \tag{2}$$

$$\left.\begin{aligned}&\min\frac{1}{2}\|w_2\|^2+\frac{v_2}{l}e^T(Aw_2+b_2e)+\frac{c_2}{l}e^T\eta\\&\text{s.t. } Y\leqslant Aw_2+b_2e+\eta,\eta\geqslant 0\end{aligned}\right\} \tag{3}$$

其中,v_1,c_1,v_2 和 c_2 为惩罚参数,ξ和η为松弛变量.

引入拉格朗日乘子α,公式(2)的对偶形式为

$$\left.\begin{aligned}&\max-\frac{1}{2}\alpha^TAA^T\alpha-Y^T\alpha+\frac{v_1}{l}e^TAA^T\alpha\\&\text{s.t. } 0\leqslant\alpha\leqslant\frac{c_1}{l}e,e^T\alpha=v_1\end{aligned}\right\} \tag{4}$$

根据 KKT 条件,我们可以计算出:

$$w_1=A^T\left(\frac{v_1}{l}e-\alpha\right),b_1=\frac{1}{|SV_1|}\sum_{i\in SV_1}(y_i-Aw_1) \tag{5}$$

其中,SV_1 是满足 $\alpha_i\in\left(0,\frac{c_1}{l}\right),i=1,...,l$ 的样本指标的集合,$|\cdot|$是集合的基数.

同理,引入拉格朗日乘子β,公式(3)的对偶形式为

$$\left.\begin{aligned}&\max-\frac{1}{2}\beta^TAA^T\beta+Y^T\beta+\frac{v_2}{l}e^TAA^T\beta\\&\text{s.t. } 0\leqslant\beta\leqslant\frac{c_2}{l}e,e^T\beta=v_2\end{aligned}\right\} \tag{6}$$

优化后可计算出 w_2 和 b_2:

$$w_2=A^T\left(\beta-\frac{v_2}{l}e\right),b_2=\frac{1}{|SV_2|}\sum_{i\in SV_2}(y_i-Aw_2) \tag{7}$$

其中,SV_2 是满足 $\beta_i\in\left(0,\frac{c_2}{l}\right),i=1,...,l$ 的样本指标的集合,$|\cdot|$是集合的基数.

分别求出 w_1,b_1,w_2,b_2 后,可得到最终的回归函数:

$$f(x)=\frac{1}{2}(f_1(x)+f_2(x)) \tag{8}$$

2 最小二乘孪生参数化不敏感支持向量回归机(LSTPISVR)

由第 1 节的理论得知,标准 TPISVR 算法可归结为在对偶空间求解两个二次规划问题.然而,这种求解方法对于解决样本数目较大的问题将受到时间和内存的制约.针对这个问题,我们引入最小二乘方法,把 TPISVR 的不等式约束条件修改为等式约束条件,然后直接在原始空间对带有等式约束的 QPP 进行求解.这一策略可以简化 TPISVR 的计算复杂性.基于这个思想,我们将提出最小二乘孪生参数化不敏感支持向量回归机(least squares twin parametric insensitive support vector regression,简称 LSTPISVR).

2.1 线性LSTPISVR

对于线性情况,引入最小二乘方法,并分别加入正则项 b_1^2 和 b_2^2,这样可以避免矩阵奇异性,经验上能提高泛化能力,则公式(2)和公式(3)变为

$$\left.\begin{aligned}&\min\frac{1}{2}(\|w_1\|^2+b_1^2)-v_1e^T(Aw_1+b_1e)+\frac{c_1}{2}\xi^T\xi\\&\text{s.t. } Y=Aw_1+b_1e-\xi\end{aligned}\right\} \tag{9}$$

$$\left.\begin{aligned}&\min\frac{1}{2}(\|w_2\|^2+b_2^2)+v_2e^T(Aw_2+b_2e)+\frac{c_2}{2}\eta^T\eta\\&\text{s.t. } Y=Aw_2+b_2e+\eta\end{aligned}\right\}\tag{10}$$

其中,v_1,c_1,v_2 和 c_2 为惩罚参数,ξ和η为松弛变量.

为了求解公式(9),把其等式约束条件代入目标函数,则公式(9)变为

$$L=\frac{1}{2}(\|w_1\|^2+b_1^2)-v_1e^T(Aw_1+b_1e)+\frac{c_1}{2}\|Aw_1+b_1e-Y\|^2\tag{11}$$

对公式(11)关于 w_1 和 b_1 求导,并令其为 0,则有:

$$\frac{\partial L}{\partial w_1}=w_1-v_1A^Te+c_1A^T(Aw_1+b_1e-Y)=0\tag{12}$$

$$\frac{\partial L}{\partial b_1}=b_1-v_1e^Te+c_1e^T(Aw_1+b_1e-Y)=0\tag{13}$$

结合公式(12)和公式(13)两个式子,并用矩阵的形式来表示,可得:

$$c_1\begin{bmatrix}A^TA+\dfrac{1}{c_1}I & A^Te\\ e^TA & e^Te+\dfrac{1}{c_1}\end{bmatrix}\begin{bmatrix}w_1\\b_1\end{bmatrix}-v_1\begin{bmatrix}A^Te\\e^Te\end{bmatrix}-c_1\begin{bmatrix}A^TY\\e^TY\end{bmatrix}=0\tag{14}$$

令 H=[A e],则我们可以得到:

$$\begin{bmatrix}w_1\\b_1\end{bmatrix}=(c_1H^TH+I)^{-1}(c_1H^TY+v_1H^Te)\tag{15}$$

采用类似的方法,我们也可以得到公式(10)的解:

$$\begin{bmatrix}w_2\\b_2\end{bmatrix}=(c_2H^TH+I)^{-1}(c_2H^TY-v_2H^Te)\tag{16}$$

分别求出 w_1,b_1,w_2,b_2 后,可得到最终的回归函数:

$$f(x)=\frac{1}{2}(f_1(x)+f_2(x))\tag{17}$$

由文献[18]可知,TPISVR 的时间复杂度为 $O(l^3)$,其中,l 为训练集的样本数.注意到公式(15)和公式(16),本文提出的算法仅仅是计算 2 个线性方程组,最终计算 2 个维数是(n+1)×(n+1)的逆矩阵,其中,n 为维数,远远小于训练集的样本数 l.并且与 TPISVR 的对偶 QPPs(公式(4)和公式(6))相比,公式(15)和公式(16)没有任何约束条件,这意味着 LSTPISVR 的训练速度要比 TPISVR 的快,特别是处理大样本数据集时,这种优势更加明显.

2.2 非线性LSTPISVR

对于非线性情况,我们利用含有核函数的非线性回归估计函数代替线性回归估计函数,把线性 LSTPISVR 推广到非线性 LSTPISVR.

对于非线性 LSTPISVR,其目标是要找到以下两个函数:

$$f_1(x)=K(x^T,A^T)w_1+b_1,f_2(x)=K(x^T,A^T)w_2+b_2\tag{18}$$

其中,$K(x^T,A^T)$是任意的核函数.用松弛变量的 2 范式代替原来的 1 范式,则原来的不等式约束就变成了等式约束:

$$\left.\begin{aligned}&\min\frac{1}{2}(\|w_1\|^2+b_1^2)-v_1e^T(K(A,A^T)w_1+b_1e)+\frac{c_1}{2}\xi^T\xi\\&\text{s.t. } Y=K(A,A^T)w_1+b_1e-\xi\end{aligned}\right\}\tag{19}$$

$$\left.\begin{aligned}&\min\frac{1}{2}(\|w_2\|^2+b_2^2)+v_2e^T(K(A,A^T)w_2+b_2e)+\frac{c_2}{2}\eta^T\eta\\&\text{s.t. } Y=K(A,A^T)w_2+b_2e+\eta\end{aligned}\right\}\tag{20}$$

采用与线性情况相似的方法,我们可以得到公式(19)和公式(20)的解.令 R=[K(A,A^T) e],则:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = (c_1 R^T R + I)^{-1}(c_1 R^T Y + v_1 R^T e) \tag{21}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (c_2 R^T R + I)^{-1}(c_2 R^T Y - v_2 R^T e) \tag{22}$$

与线性 LSTPISVR 不同,非线性 LSTPISVR 计算的是 2 个维数是(l+1)×(l+1)的逆矩阵,其中,l 是训练样本集的数目.与非线性 TPISVR 的 QPPs 相比,公式(21)和公式(22)没有约束条件,所以非线性 LSTPISVR 的学习速度比非线性 TPISVR 要快.

2.3 混沌布谷鸟优化算法的LSTPISVR参数选择

在 LSTPISVR 算法中有 c_1,c_2,v_1,v_2 这 4 个惩罚参数,在非线性情况下,在此基础上再多一个核函数的核参数 σ.LSTPISVR 性能的好坏在一定程度上依赖于参数的选择.本文提出一种具有较强优化能力的混沌布谷鸟优化算法,并用其对 LSTPISVR 的参数进行优化选择.

2009 年,借鉴布谷鸟种寄生繁衍现象,Yang 和 Deb 提出了布谷鸟搜索(cuckoo search,简称 CS)算法[22].CS 的基本算法流程如下.

(1) 在 D 维解空间中随机生成 N 个鸟窝 $Nest_i(x_1,x_2,\ldots,x_D)$,$1\leqslant i\leqslant N$,计算每个鸟窝的适应度值,记为 $f(Nest_i)$,$1\leqslant i\leqslant N$,保留适应度最优的鸟窝位置进行迭代;

(2) 设 $x_i^{(t)}$ 为第 i 个鸟窝在第 t 代的鸟窝位置,$Levy(\lambda)$为随机搜索路径,那么布谷鸟寻窝的路径和位置更新公式为

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda)(i = 1,2,...,n) \tag{23}$$

公式(23)中,α为步长,⊕为点对点乘法.

(3) 设鸟窝主人发现外来鸟蛋的概率为 P,产生一个均匀分布随机数 $r\in[0,1]$,若 $r>P$,则对所发现的鸟窝位置进行扰动,否则不变.

与其他群智能优化算法一样,CS 算法在搜索过程中也容易陷入局部最优和搜索精度不够高等不足.为了解决这个问题,在本文中,我们把混沌变量引入到 CS 算法中,利用混沌变量随机性、遍历性和规律性等特性,让 CS 可以跳出局部最优,从而增强 CS 的全局搜索能力,进而提高 CS 算法解的精度.

一般地,一个典型的混沌映射系统 Logistic 方程可以表述为

$$y_{n+1}=4y_n(1-y_n) \tag{24}$$

其中,n=1,2,…,n,y_n 为混沌变量,$y_n\in[0,1]$.

在改进算法中,当算法连续迭代 k 次,解的值都不发生变化时,我们可以认为算法出现了早熟收敛现象.此时,我们利用混沌的遍历性,对鸟窝当前的最优位置 x_{best} 进行混沌优化,方法如下.

将 x_{best} 通过方程映射到 Logistic 方程的定义域[0,1]上:

$$y_1^k = \frac{x_{best} - x_{\min}^k}{x_{\max}^k - x_{\min}^k} \tag{25}$$

对 y_1^k 通过 $y_{n+1}^k = 4y_n^k(1-y_n^k)(n=1,2,...,n)$ 进行 T 次迭代,得到混沌序列 $y^k = (y_1^k, y_2^k, ..., y_T^k)$.

然后,将混沌序列通过下面的式子逆映射回原解空间:

$$x_{best,m}^{*k} = x_{\min}^k + (x_{\max}^k - x_{\min}^k)y_m^k, m = 1,2,...,T \tag{26}$$

计算可行解序列中每个可行解矢量的适应值,并保留适应度值最优时对应的可行解矢量,记为 x_{best}^{*k}.从当前鸟窝群随机选择一个鸟窝,并用 x_{best}^{*k} 的位置矢量代替选出鸟窝的位置矢量.

在本文中,依据回归问题的评价准则,我们用公式(27)来设计适应度函数:

$$fitness = \frac{1}{(RMSE)^2} \tag{27}$$

则基于混沌布谷鸟优化算法的 LSTPISVR 参数选择算法步骤可以简要概括如下:

Step 1(初始化):在四维空间中随机生成 N 个鸟窝 $Nest_i(c_1,c_2,v_1,v_2)$,i=1,…,N,或者在五维空间中随机生成 N 个鸟窝 $Nest_i(c_1,c_2,v_1,v_2,\sigma)$,$i$=1,…,$N$;

Step 2(计算适应度值):根据公式(27)计算每个鸟窝的适应度值.

Step 3(更新):每个鸟窝按公式(23)、公式(26)计算新的位置.

Step 4(终止算法):如果已经达到终止条件,输出最优值,否则转入 Step 2.

3 实验与分析

为了测试 LSTPISVR 算法的性能,我们将对 1 个人工数据集和 6 个 UCI 数据集进行测试,并与 TPISVR, TSVR 和 par-v-SVR 的测试结果进行比较.以上几种方法都是在 Intel(R) Core(TM)2 Duo CPU E4500、2G 内存和 MATLAB7.11.0 的环境中运行的.如果没有其他特别说明,在本文中,我们都是采用 10 折交叉验证方法来测量这些方法的回归精度,并且对于非线性情况,我们仅仅考虑高斯核函数 $K(x,x_i)=\exp\left(-\dfrac{\|x-x_i\|^2}{2\sigma^2}\right)$.与很多其他的机器学习方法一样,这 4 种方法的学习性能对参数的选择非常敏感.表 1 列出了在线性情况下各种算法需要调整的参数.从表 1 我们可以看出:TPISVR 和 LSTPISVR 分别有 4 个惩罚参数需要进行调整,参数的选择合适与否将影响到算法的性能,因此,选择合适的参数选择算法是非常重要的.在实验中,所有的参数将从{2^{-8},…,2^7}这个范围内进行选择.LSTPISVR 参数选择方法采用的是第 2.3 节提出的混沌布谷鸟算法,而其他的 3 个算法采用的是常用的网格搜索优化算法.混沌布谷鸟算法的参数设置如下:步长 α=1.5,概率 P=0.35,k=5.

Table 1 The parameters of four algorithms with linear kernel

表 1 线性情况下,4 种算法分别包含的参数

LSTPISVR	TPISVR	TSVR	par-v-SVR
(c_1,c_2,v_1,v_2)	(c_1,c_2,v_1,v_2)	(c_1,c_2)	(c,v)

3.1 人工数据集上的实验

sin$c$ 函数经常被用来测试各种机器学习方法的回归性能[21],其表达式为

$$y=\sin c(x_i)=\frac{\sin x_i}{x_i}+e_i, x_i\sim U[-3\pi,3\pi] \tag{28}$$

为了更有效地测试我们的算法,在训练样本中分别加入具有两种不同异方差结构的噪声,它们分别为:(1) Type A: $e_i=\left(-\dfrac{|x_i|}{8\pi}+0.5\right)\varepsilon_i,\varepsilon_i\sim U[-0.5,0.5]$;(2) Type B: $e_i=\left(-\dfrac{|x_i|}{8\pi}+0.5\right)\varepsilon_i,\varepsilon_i\sim N(0,0.25^2)$.

其中,$U[a,b]$表示的是在$[a,b]$上的均匀随机变量;$N(c,d^2)$表示的是均值为 c、方差为 d^2 的高斯随机变量.为了提高比较结果的可靠性,用 Matlab 工具箱对每种噪声分别产生 10 组噪声样本,每组噪声样本包括 500 个训练样本和 500 个测试样本.表 2 是 par-v-SVR,TSVR,TPISVR 和 LSTPISVR 分别运行 10 次的平均结果.图 1 和图 2 分别是 par-v-SVR,TSVR,TPISVR 和 LSTPISVR 对带不同噪声的 sin$c$ 函数运行一次的结果.

从表 2 中我们看出:对于带有 Type A 噪声的 sin$c$ 函数,与其他 3 种算法相比,LSTPISVR 可以在更短的时间内获得更好的回归效果;而对于带有 Type B 噪声的 sin$c$ 函数而言,虽然 TPISVR 的 *RMSE* 值要优于 LSTPISVR 的,但 LSTPISVR 的运行效率要高于 TPISVR,并且 LSTPISVR 的 *RMSE* 值已经接近于 TPISVR 的值,还在可以接受的范围.总之,LSTPISVR 的性能还是优于其他 3 种算法.

Table 2 Results of four algorithms on sin$c$ function with two types of noises

表 2 4 种算法在带有两种不同类型噪声 sin$c$ 函数下的比较结果

噪声	评价指标	LSTPISVR	TPISVR	TSVR	par-v-SVR
Type A	*RMSE*	0.0954±0.0023	0.0957±0.0015	0.1028±0.0030	0.0957±0.0017
	时间(s)	0.078 5	0.082 5	0.091 8	0.361 2
Type B	*RMSE*	0.0858±0.0038	0.0856±0.0054	0.0879±0.0024	0.0872±0.0027
	时间(s)	0.068 7	0.078 1	0.081 5	0.371 2

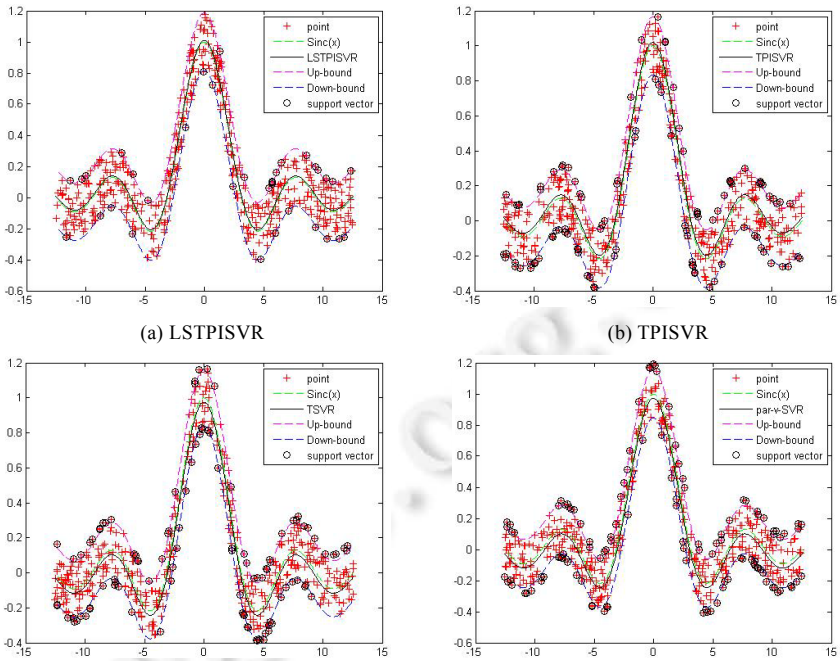


Fig.1 Fitting results of LSTPISVR, TPISVR, TSVR and par-*v*-SVR on sinc(*x*)with noise Type A

图 1 LSTPISVR,TPISVR,TSVR 和 par-*v*-SVR 对带有 Type A 噪声的 sinc(*x*)的拟合结果

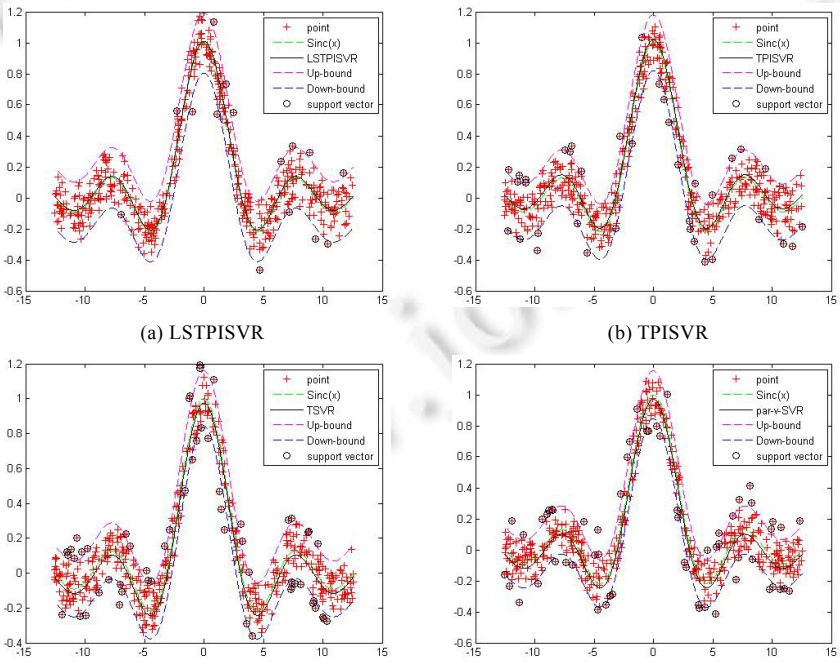


Fig.2 Fitting results of LSTPISVR, TPISVR, TSVR and par-*v*-SVR on sinc(*x*)with noise Type B

图 2 LSTPISVR,TPISVR,TSVR 和 par-*v*-SVR 对带有 Type B 噪声的 sinc(*x*)的拟合结果

从图 1 和图 2 可以明显看出:对于这两种噪声的数据集,LSTPISVR 算法的支持向量是比较少的,并且它的拟合效果比其他 3 种算法要好.

3.2 UCI数据集上的实验

为了更进一步地测试算法性能,我们将对文献[21]中的 6 个 UCI 数据集进行测试.这 6 个 UCI 数据集分别是 Motorcycle,Boston,Auto-Mpg,Machine CPU,Servo 和 Auto price.表 3 显示了在线性情况下,par-v-SVR,TSVR,TPISVR 和 LSTPISVR 分别对这 6 个数据集运行 10 次的平均结果.表 4 显示了在非线性情况下,par-v-SVR,TSVR,TPISVR 和 LSTPISVR 采用高斯核分别对这 6 个数据集运行 10 次的平均结果.图 3 表示在 LSTPISVR 中,混沌布谷鸟算法的鸟窝个数、迭代次数与适应度值之间的关系.

从表 3 和表 4 的实验结果可以看出:线性核和高斯核的情况下,对于这 6 个 UCI 数据集,LSTPISVR 都可以在更短的时间内获得更好的 *RMSE* 值.对于一些数据集,采用高斯核的时候,LSTPISVR 的 *RMSE* 值要好于采用线性核的情况,但其时间消耗要多一些.图 3 分析了 LSTPISVR 中混沌布谷鸟算法鸟窝个数、迭代次数与适应度值之间的关系.从图 3(a)、图 3(b)可以看出:在鸟窝个数选取 100 的情况下,线性和非线性都达到了稳定状态,因此在本文中,鸟窝个数的初始值为 100;从图 3(c)、图 3(d)也可以看出:在迭代次数为 100 之后,算法的适应度值几乎不再有变化,达到了稳定状态,因此在本文中,迭代次数定为 100.同时我们也可以看出:采用混沌布谷鸟算法来优化 LSTPISVR,算法很快就达到了稳定状态.这些实验结果表明,LSTPISVR 的性能要优于其他 3 种算法.

Table 3 Results of four algorithms on UCI dataset with linear kernel

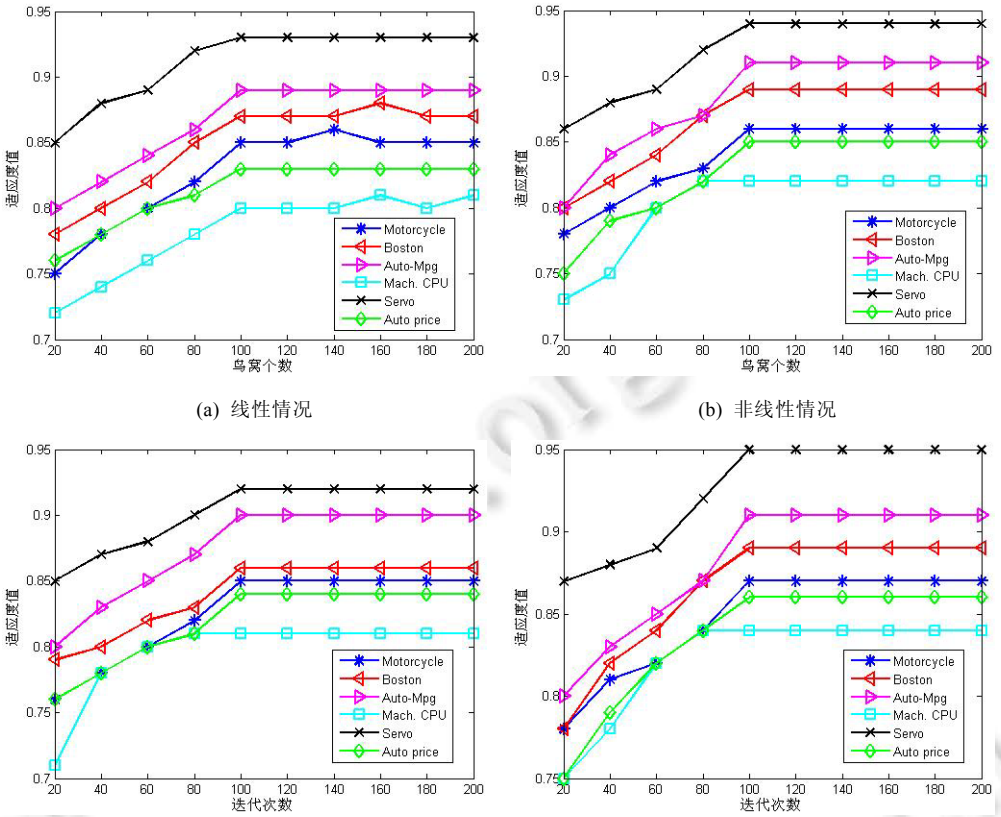
表 3 基于线性核的 4 种算法在 UCI 数据集的测试结果

数据集	评价指标	LSTPISVR	TPISVR	TSVR	par-v-SVR
Motorcycle (133×2)	*RMSE*	0.2185±0.0023	0.2205±0.0025	0.2296±0.0031	0.2249±0.0028
	时间(s)	0.042 5	0.051 9	0.045 7	0.154 2
Boston (506×14)	*RMSE*	0.1523±0.0422	0.1569±0.0463	0.1641±0.0378	0.1633±0.0402
	时间(s)	0.106 8	0.155 2	0.159 6	0.523 6
Auto-Mpg (398×8)	*RMSE*	0.1341±0.0452	0.1352±0.0403	0.1365±0.0395	0.1369±0.0463
	时间(s)	0.073 6	0.087 5	0.093 6	0.372 8
Mach. CPU (209×7)	*RMSE*	0.0928±0.0065	0.0942±0.0068	0.0933±0.0073	0.0949±0.0091
	时间(s)	0.018 5	0.019 7	0.019 2	0.080 5
Servo (167×5)	*RMSE*	0.1819±0.0925	0.1836±0.0957	0.1912±0.1043	0.1854±0.0938
	时间(s)	0.023 5	0.025 4	0.025 9	0.114 5
Auto price (159×16)	*RMSE*	0.1609±0.0439	0.1627±0.0451	0.1653±0.0452	0.1641±0.0496
	时间(s)	0.024 3	0.026 8	0.027 5	0.115 2

Table 4 Results of four algorithms on UCI dataset with Gaussian kernel

表 4 基于高斯核的 4 种算法在 UCI 数据集的测试结果

数据集	评价指标	LSTPISVR	TPISVR	TSVR	par-v-SVR
Motorcycle (133×2)	*RMSE*	0.2178±0.0017	0.2202±0.0015	0.2292±0.0026	0.2245±0.0022
	时间(s)	0.042 8	0.052 3	0.045 9	0.154 5
Boston (506×14)	*RMSE*	0.1518±0.0426	0.1565±0.0451	0.1634±0.0365	0.1619±0.0407
	时间(s)	0.107 5	0.155 6	0.160 8	0.524 1
Auto-Mpg (398×8)	*RMSE*	0.1334±0.0446	0.1345±0.0405	0.1358±0.0387	0.1362±0.0459
	时间(s)	0.074 2	0.087 9	0.094 5	0.374 1
Mach. CPU (209×7)	*RMSE*	0.0913±0.0067	0.0928±0.0065	0.0931±0.0076	0.0943±0.0093
	时间(s)	0.019 3	0.021 4	0.019 8	0.082 8
Servo (167×5)	*RMSE*	0.1812±0.0922	0.1821±0.0955	0.1902±0.1045	0.1842±0.0943
	时间(s)	0.023 8	0.026 3	0.026 7	0.117 6
Auto price (159×16)	*RMSE*	0.1564±0.0433	0.1602±0.0455	0.1624±0.0458	0.1637±0.0454
	时间(s)	0.024 8	0.027 5	0.027 9	0.115 7



(a) 线性情况 (b) 非线性情况

(c) 线性情况 (d) 非线性情况

Fig.3 The influence of the number of nest to the fitness and the influence of the total iteration to the fitness of Chaos cuckoo algorithm in LSTPISVR

图 3 在 LSTPISVR 中,混沌布谷鸟算法的鸟窝个数、迭代次数与适应度值之间的关系

4 结束语

标准孪生参数化不敏感支持向量回归机的模型是在对偶空间求解一对带有不等式约束的二次规划问题,然而这种求解方法的时间消耗比较大.为了提高 TPISVR 的训练速度,在本文中,我们引入最小二乘方法,将不等式约束条件转化为等式约束条件,并把两个 QPP 转化为两个线性方程且在原空间上直接进行求解,提出了最小二乘孪生参数化不敏感支持向量回归机.与 TPISVR 一样,TPISVR 有 4 个参数需要选择.对于机器学习算法而言,其参数选择合适与否,将影响到其学习性能.本文提出具有较强寻优能力的混沌布谷鸟算法作为 LSTPISVR 的参数选择方法.在人工数据集和 UCI 数据集上的实验表明:LSTPISVR 的训练速度比 TPISVR 的快,而且回归精度并没有变差.鉴于 LSTPISVR 的良好性能,将其应用于时间序列领域,将是下一步的研究工作.

References:

[1] Vapnik VN. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995. [doi: 10.1007/978-1-4757-3264-1]

[2] Xu XZ, Ding SF, Shi ZZ, Zhu H. Optimizing radial basis function neural network based on rough sets and affinity propagation clustering algorithm. Journal of Zhejiang University—SCIENCE C, 2012,13(2):131−138. [doi: 10.1631/jzus.C1100176]

[3] Wang XB, Zhou DL, Wang SJ. Constructive neuron networks classification algorithm based on biomimetic pattern recognition. Chinese Journal of Computers, 2007,30(12):2109−2114 (in Chinese with English abstract). [doi: 10.3321/j.issn:0254-4164.2007.12.006]

[4] Pan H, Zhu YP, Xia LZ. Efficient and accurate face detection using heterogeneous feature descriptors and feature selection. Computer Vision and Image Understanding, 2013,117(1):12−28. [doi: 10.1016/j.cviu.2012.09.003]

[5] Chen ZY, Zhi ZP. Distributed customer behavior prediction using multiplex data: A collaborative MK-SVM approach. Knowledge-Based Systems, 2012,35:111−119. [doi: 10.1016/j.knosys.2012.04.023]

[6] Moraes R, Valiati JF, Gaviao N, Wilson P. Document-Level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Application, 2013,40(2):621−633. [doi: 10.1016/j.eswa.2012.07.059]

[7] Wu JX. Efficient HIK SVM learning for image classification. IEEE Trans. on Image Processing, 2012,21(10):4442−4453. [doi: 10.1109/TIP.2012.2207392]

[8] Cortes C, Vapnik VN. Support vector networks. Machine Learning, 1995,20:273−297. [doi: 10.1007/BF00994018]

[9] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines. In: Proc. of the '97 IEEE Workshop on Neural Networks for Signal Processing. New York: IEEE Press, 1997. 276−285. [doi: 10.1109/NNSP.1997.622408]

[10] Platt JC. Using analytic QP and sparseness to speed training of support vector machines. In: Kearns M, Solla S, Cohn D, eds. Proc. of the Advances in Neural Information Processing Systems 11. Cambridge: MIT Press, 1999. 557−563.

[11] Ding SF, Huang HJ, Shi ZZ. Weighted smooth CHKS twin support vector machines. Ruan Jian Xue Bao/Journal of Software, 2013, 24(11):2548−2557 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4475.htm[doi: 10.3724/SP.J.1001.2013.04475]

[12] Ding SF, Huang HJ, Xu XZ, Wang J. Polynomial smooth twin support vector machines. Applied Mathematics & Information Sciences, 2014,8(4):2063−2071. [doi: 10.12785/amis/080465]

[13] Jayadeva KR, Suresh C. Twin support vector machines for pattern classification. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2007,29(5):905−910. [doi: 10.1109/TPAMI.2007.1068]

[14] Cong HH, Yang CF, Pu XR. Efficient speaker recognition based on multi-class twin support vector machines and GMMs. In: Proc. of the 2008 IEEE Conf. on Robotics, Automation and Mechatronics. 2008. 348−352. [doi: 10.1109/RAMECH.2008.4681433]

[15] Zhang XS, Gao XB, Wang Y. Twin support tensor machines for MCs detection. Journal of Electronics (China), 2009,26(3): 318−325. [doi: 10.1007/s11767-007-0211-0]

[16] Xu YT. *K*-Nearest neighbor-based weighted multi-class twin support vector machine. Neurocomputing, 2016,205:430−438. [doi: 10.1016/j.neucom.2016.04.024]

[17] Reshma K, Pooja S, Suresh C. Improvements on *v*-twin support vector machine. Neural Networks, 2016,79:97−107. [doi: 10.1016/j.neunet.2016.03.011]

[18] Peng XJ. TSVR: An efficient twin support vector machine for regression. Neural Networks, 2010,23:365−372. [doi: 10.1016/j.neunet.2009.07.002]

[19] Tanveer M, Shubham K, Aldhaifallah M, Ho SS. An efficient regularized *K*-nearest neighbor based weighted twin support vector regression. Knowledge-Based Systems, 2016,94:70−87. [doi: 10.1016/j.knosys.2015.11.011]

[20] Hao PY. New support vector algorithms with parametric insensitive/margin model. Neural Networks, 2010,23(1):60−73. [doi: 10.1016/j.neunet.2009.08.001]

[21] Peng XJ. Efficient twin parametric insensitive support vector regression model. Neurocomputing, 2012,79:26−38. [doi: 10.1016/j.neucom.2011.09.021]

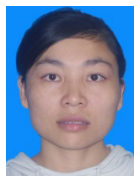
[22] Yang XS, Suash DEB. Cuckoo search via levy flights. In: Proc. of the World Cngress on Nature & Biologically Inspired Computing. India: IEEE Publications, 2009. 210−214.

附中文参考文献:

[3] 王宪保,周德龙,王守觉.基于仿生模式识别的构造型神经网络分类方法.计算机学报,2007,30(12):2109−2114. [doi: 10.3321/j.issn:0254-4164.2007.12.006]

[11] 丁世飞,黄华娟,史忠植.加权光滑 CHKS 孪生支持向量机.软件学报,2013,24(11):2548−2557. http://www.jos.org.cn/1000-9825/4475.htm [doi: 10.3724/SP.J.1001.2013.04475]

丁世飞(1963－),男,山东青岛人,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为人工智能,机器学习,数据挖掘,粒度计算.



黄华娟(1984－),女,博士,讲师,主要研究领域为机器学习,模式识别.