

文章编号: 1001-0920(2008)12-1347-06

一种快速稀疏最小二乘支持向量回归机

赵永平, 孙健国

(南京航空航天大学 能源与动力学院, 南京 210016)

摘要: 将 Jiao 法直接应用于最小二乘支持向量回归机上的效果并不理想, 为此采用不完全抛弃的策略, 提出了改进的 Jiao 法, 并将其应用于最小二乘支持向量回归机. 数据集测试的结果表明, 基于改进 Jiao 法的稀疏最小二乘支持向量回归机, 无论在支持向量个数和训练时间上都取得了一定的优势. 与其他剪枝算法相比, 在不丧失回归精度的情况下, 改进的 Jiao 法可大大缩短训练时间. 另外, 改进的 Jiao 法同样适用于分类问题.

关键词: 最小二乘支持向量回归机; Jiao 法; 剪枝算法; 分类

中图分类号: TP301.6 **文献标识码:** A

Fast method for sparse least squares support vector regression machine

ZHAO Yong-ping, SUN Jian-guo

(College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China. Correspondent: ZHAO Yong-ping, E-mail: y.p.zhao@nuaa.edu.cn)

Abstract: Jiao's method is directly used for least squares support vector regression machine (LS-SVRM), but the effectiveness isn't satisfying. Therefore, an improved Jiao's method built on partial reduction is proposed, and a fast sparse least squares support vector regression machine (FPLS-SVRM) on the basis of the improved Jiao's method is presented. Simulation results of data sets demonstrate that FPLS-SVRM gains advantages over LS-SVRM based on Jiao's method in the number of support vectors and training time in a way. Meanwhile, in comparison with other pruning algorithms, FPLS-SVRM reduces training time without loss of regression accuracy. In addition, the improved Jiao's method is similarly applied to classification.

Key words: Least squares support vector regression machine; Jiao's method; Pruning algorithm; Classification

1 引言

支持向量机^[1]是解决数据挖掘、分类和回归估计等^[2,3]的高效方法. 它建立在结构风险最小化的基础上, 具有全局最优性. 原始的支持向量机是基于不等式约束, 其解具有稀疏性和泛化能力强的优点. 将原支持向量机的不等式约束变成等式约束, Suykens 等提出了最小二乘支持向量机^[4], 并对其进行了深入的研究^[5]. 一个数学模型要有较强的泛化能力一般通过两个途径来实现^[6]: 1) 引入正则化参数; 2) 降低模型的复杂度, 减少回归模型的输入项. 由于最小二乘支持向量机的解缺乏稀疏性^[7], 其泛化性较差, 而且增加了回归预测的时间. 因此, 实现最小二乘支持向量机解的稀疏性很重要. 稀疏性

简化了最小二乘支持向量机的模型, 提高了泛化能力, 缩短了预测时间.

为了得到解的稀疏性, 人们进行了一些研究, 其中一个重要方面是对剪枝算法的研究. Suykens 等^[8]提出了基于最小支持权重的剪枝算法; Kruif 等^[9]提出了基于最小化模型预报误差的剪枝算法; Hoegaerts 等^[10]将 Kruif 等提出的剪枝准则进行了改进, 并对这些剪枝算法进行了比较. 序贯最小优化法 (SMO)^[11]原本是用来解决大量数据下支持向量机训练问题的, 但 Keerthi 等^[12]则将其应用于最小二乘支持向量机, 在此基础上, Zeng 等^[13]提出了基于 SMO 的剪枝算法. 最近, Jiao 等^[14]提出了一种用于分类的快速剪枝算法. 剪枝算法首先应确定剪枝

收稿日期: 2007-09-22; 修回日期: 2008-01-01.

基金项目: 国家自然科学基金项目 (50576033).

作者简介: 赵永平 (1982—), 男, 河南南阳人, 博士生, 从事机器学习、控制理论和控制方法的研究; 孙健国 (1939—), 男, 浙江海宁人, 教授, 博士生导师, 从事控制、建模和故障诊断等研究.

点,剪枝后需重新训练,这都将要浪费大量的时间.如何快速找到剪枝点,并在剪枝后迅速建立起数学模型,这些都是加快最小二乘支持向量机学习行之有效的办法.

本文将 Jiao 等提出的快速剪枝算法应用于回归,效果并不理想.为此,提出了基于改进 Jiao 法的快速稀疏最小二乘支持向量回归机,并通过数据集的测试验证了改进的有效性.

2 最小二乘支持向量回归机

通过将支持向量机的不等式约束改成等式约束, Suykens 等给出了最小二乘支持向量回归机的数学模型

$$\min_{W, e} J(W, e) = \frac{1}{2} W^T W + \frac{C}{2} \sum_{i=1}^N e_i^2, \quad (1)$$

s. t. $d_i = W^T(x_i) + b + e_i.$

其中 e_i 代表系统测量值与真实输出之间的误差, C

R 为正则化参数, (\cdot) 为输入空间到特征空间的非线性映射. 式(1)的 Wolfe 对偶优化问题为

$$\min \left\{ \frac{1}{2} \sum_{i,j=1}^N i_j (x_i)^T (x_j) + \sum_{i=1}^N \frac{2}{2C} \sum_{i=1}^N i y_i \right\}, \quad (2)$$

s. t. $\sum_{i=1}^N i = 0.$

将式(2)的约束优化转化为无约束优化问题

$$\min \left\{ \frac{1}{2} \sum_{i,j=1}^N i_j (x_i)^T (x_j) + \sum_{i=1}^N \frac{2}{2C} \sum_{i=1}^N i y_i + b \sum_{i=1}^N i \right\}. \quad (3)$$

式(3)与(2)是完全等价的.用满足 Mercer 条件的核函数 $k(\cdot, \cdot)$ 代替 $(x_i)^T (x_j)$, 即 $k(x_i, x_j) = (x_i)^T (x_j)$, 其中常用的核函数有 RBF 核、多项式核和 Sigmoid 核等. 将式(3)变形, 可得

$$\min \left\{ g = \frac{1}{2} [b \quad \mathbf{1}^T] \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \bar{K} \end{bmatrix} \begin{bmatrix} b \\ \mathbf{d} \end{bmatrix} \right\}. \quad (4)$$

其中: $\bar{K} = K + (1/C)I$, $K_{ij} = k(x_i, x_j)$, I 为与 K 同阶的单位矩阵, $\mathbf{1} = [1, 1, \dots, 1]^T$. 于是, 可得到最小二乘支持向量回归机的数学模型为

$$y = f(x) = \sum_{i=1}^N i K(x, x_i) + b. \quad (5)$$

其中 b 和 $\mathbf{d} = [d_1, d_2, \dots, d_N]^T$ 是下式的解:

$$\begin{bmatrix} b \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \bar{K} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{d} \end{bmatrix}. \quad (6)$$

由上述建模过程可知, 训练数据集中的每一个

数据, 对目标函数(1)都产生一个有效约束, 也就是说, 最小二乘支持向量回归机的解丧失了稀疏性的优点. 这就导致了最小二乘支持向量回归机预测时间较长的缺陷, 同时在一定程度上影响了它的泛化能力. 基于此, 本文提出一种快速稀疏最小二乘支持向量回归机.

3 Jiao 方法

首先定义两个集合 $P = \emptyset$ 和 $Q = \{1, 2, \dots, N\}$, P 和 Q 分别代表工作集和训练数据集的索引; 然后按一定的标准从训练数据集中选择一个数据 x_s 到工作集中, 则 $Q = Q - \{s\} = \{1, \dots, s-1, s+1, \dots, N\}$, $P = \{s\}$. 重复以上步骤, 一直达到停机标准为止, 此时 P 中是支持向量的索引, Q 中是非支持向量的索引. 这便实现了最小二乘支持向量回归机的稀疏性.

3.1 核矩阵的迭代求逆

如果在 $(n+1)$ 步数据 x_s 被选中, 此时 $|P| = n$, 则式(6)中矩阵的逆

$$R^{n+1} = \begin{bmatrix} 0 & \mathbf{1}_{/P}^T & \mathbf{1} \\ \mathbf{1}_{/P} & \bar{K}_{PP} & \bar{k}_s \\ 1 & \bar{k}_s & \bar{K}_{ss} \end{bmatrix}^{-1}. \quad (7)$$

其中: $\bar{k}_s = [\bar{K}_{P_1s}, \bar{K}_{P_2s}, \dots, \bar{K}_{P_ns}]^T$, $/P$ 代表集合的势, P_i 代表集合 P 中第 i 个元素. 若已计算出

$$R^n = \begin{bmatrix} 0 & \mathbf{1}_{/P}^T \\ \mathbf{1}_{/P} & \bar{K}_{PP} \end{bmatrix},$$

则由 Sherman-Morrion 公式^[15]可知

$$R^{n+1} = R^n + \begin{bmatrix} & & \\ & & \\ \mathbf{1} & \mathbf{1}^T & 1 \end{bmatrix}. \quad (8)$$

其中

$$= R^n \begin{bmatrix} 1 \\ \mathbf{1} \end{bmatrix}, \quad = (\bar{K}_{ss} - [\mathbf{1} \quad \bar{k}_s^T])^{-1}.$$

根据式(8), 第 $n+1$ 步的 b 和 \mathbf{d} 为

$$\begin{bmatrix} b^{n+1} \\ \mathbf{d}_P^{n+1} \\ \mathbf{d}_s^{n+1} \end{bmatrix} = R^{n+1} \begin{bmatrix} 0 \\ \mathbf{d}_P \\ \mathbf{d}_s \end{bmatrix} = \begin{bmatrix} R^n \begin{bmatrix} 0 \\ \mathbf{d}_P \\ 0 \end{bmatrix} \\ \mathbf{1}^T \begin{bmatrix} 0 \\ \mathbf{d}_P \\ \mathbf{d}_s \end{bmatrix} - \mathbf{d}_s \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ - \mathbf{1} \end{bmatrix}. \quad (9)$$

因为 $\begin{bmatrix} b^n \\ \mathbf{d}_P^n \end{bmatrix} = R^n \begin{bmatrix} 0 \\ \mathbf{d}_P \end{bmatrix}$, 所以

$$\begin{bmatrix} b^{n+1} \\ \mathbf{d}_P^{n+1} \\ \mathbf{d}_s^{n+1} \end{bmatrix} = \begin{bmatrix} b^n \\ \mathbf{d}_P^n \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{1}^T \begin{bmatrix} 0 \\ \mathbf{d}_P \\ \mathbf{d}_s \end{bmatrix} - \mathbf{d}_s \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ - \mathbf{1} \end{bmatrix}. \quad (10)$$

3.2 选择支持向量标准和停机标准

由式(4)可知, 第 n 步 g 的最优值为

$$g^n = -\frac{1}{2} [0 \quad d^T] \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1}_{/p} & \overline{K}_{pp} \end{bmatrix} \begin{bmatrix} 0 \\ d \end{bmatrix}. \quad (11)$$

若保持 n_p 和 b 不变, 则在第 $n+1$ 步从训练集中选择一个数据作为支持向量时, 式(11) 可简化为

$$g_i^{n+1} = \min_i \left\{ \frac{1}{2} \left(k(x_i, x_i) + \frac{1}{C} \right)^2 + r_i^n \right\}, \quad i \in Q. \quad (12)$$

其中

$$r_i^n = \begin{cases} -d_i, & n = 0; \\ \sum_{l=1}^n p_l k(x_i, x_{p_l}) + b^n - d_i, & n > 0. \end{cases} \quad (13)$$

式(12) 的最优值为

$$g_i^{n+1} = -\frac{(r_i^n)^2}{2\left(k(x_i, x_i) + \frac{1}{C}\right)}, \quad i \in Q. \quad (14)$$

由此可得第 $n+1$ 步选择 s 的标准为

$$s = \arg \min_i \left\{ \frac{-(r_i^n)^2}{2\left(k(x_i, x_i) + \frac{1}{C}\right)} \right\}. \quad (15)$$

选择的停机标准为

$$\max(|r_Q^n|) < \epsilon. \quad (16)$$

由式(16) 可知, ϵ 的大小可以控制选择支持向量的多少, ϵ 越小, 选择的支持向量越多, 反之则越少. $\epsilon = 0$ 时就是标准的最小二乘支持向量回归机.

4 快速稀疏最小二乘支持向量回归机

由式(7) ~ (9) 可知, 在计算 $[b^{n+1}, \frac{1}{p^{n+1}}, \frac{1}{s^{n+1}}]^T$ 时, 只利用了 $\{(x_i, d_i)\}_{i \in p}$ 中的信息, 而将 $\{(x_i, d_i)\}_{i \in Q}$ 中的信息完全抛弃, 这是一种不恰当的做法. 因为数据集 $\{(x_i, d_i)\}_{i \in Q}$ 中的数据对目标函数(1) 也产生有效约束, 从而影响目标函数(1) 的优化. 如果用数据集 $\{(x_i, d_i)\}_{i \in p}$ 建立最小二乘支持向量回归机时, 考虑到 $\{(x_i, d_i)\}_{i \in Q}$ 产生的约束对目标函数的影响, 则必然对最小二乘支持向量回归机模型的建立有帮助, 会在一定程度上增加其泛化能力.

令 $O = \{1, 2, \dots, N\}$, 则有

$$\overline{k}_s = \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1}_{/o} & \overline{K}_{op} \end{bmatrix}^T \begin{bmatrix} 1 \\ \overline{K}_{os} \end{bmatrix}, \quad (17)$$

$$\overline{K}_s = [1 \quad \overline{K}_s] \begin{bmatrix} 1 \\ \overline{K}_{os} \end{bmatrix}, \quad (18)$$

$$\tilde{d}_s = [1 \quad \overline{K}_s] \begin{bmatrix} 0 \\ d \end{bmatrix}, \quad (19)$$

$$\tilde{0} = \sum_{i=1}^N d_i. \quad (20)$$

若 $\tilde{R}^n = \overline{K}_{pp}^{-1}$ 已知, 则

$$\tilde{R}^{n+1} = \begin{bmatrix} \tilde{R}^n & \tilde{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \begin{bmatrix} \tilde{\sim} \\ -1 \end{bmatrix} [\tilde{\sim}^T \quad -1]. \quad (21)$$

其中: $\tilde{\sim} = \tilde{R}^n \tilde{k}_s$, $\tilde{\sim} = (\overline{K}_s - \tilde{k}_s^T \tilde{\sim})^{-1}$. 由此可得

$$\begin{bmatrix} \tilde{b}^{n+1} \\ \tilde{\sim}^{n+1} \\ \tilde{\sim}^{n+1} \end{bmatrix} = \tilde{R}^{n+1} \begin{bmatrix} \tilde{0} \\ \tilde{d}_p \\ \tilde{d}_s \end{bmatrix} = \begin{bmatrix} \tilde{R}^n \begin{bmatrix} \tilde{0} \\ \tilde{d}_p \end{bmatrix} \\ \tilde{\sim}^T \begin{bmatrix} \tilde{0} \\ \tilde{d}_p \end{bmatrix} - \tilde{d}_s \\ 0 \end{bmatrix} + \begin{bmatrix} \tilde{\sim} \\ -1 \end{bmatrix} \begin{bmatrix} \tilde{\sim} \\ -1 \end{bmatrix}. \quad (22)$$

将 $\tilde{\sim}$ 和 \tilde{b} 分别代入式(13) ~ (16), 可得到改进后的选择支持向量的标准和停机标准.

下面证明改进的合理性:

由数据集 $\{(x_i, d_i)\}_{i=1}^N$ 建立最小二乘支持向量回归机, 最后归结为求解

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \overline{K} \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix}. \quad (23)$$

解式(23) 相当于解如下优化问题:

$$\min_b L = \left\| \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \overline{K} \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2^2. \quad (24)$$

当迭代到第 n 步时, 式(24) 的优化问题变为如下优化问题:

$$\min_{p,b} L_p = \left\| \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \begin{bmatrix} b \\ d_p \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2^2. \quad (25)$$

令 $\partial L_p / \partial \tilde{\sim} = 0$, 可得

$$\begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix}^T \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \begin{bmatrix} b \\ d_p \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix}^T \begin{bmatrix} 0 \\ d \end{bmatrix}, \quad (26)$$

其中 $\tilde{\sim}_p = \begin{bmatrix} b \\ d_p \end{bmatrix}$. 由式(17) ~ (20) 可知, 改进后的

迭代过程就是通过迭代建立式(26) 来求解 $\tilde{\sim}_p$. 另外, 若 $\tilde{\sim}_p$ 和 b 是式(26) 的解, a 和 b 是改进前的解, 即是下式的解:

$$\begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{pp} \end{bmatrix}^T \begin{bmatrix} b \\ d_p \end{bmatrix} = \begin{bmatrix} 0 \\ d_p \end{bmatrix},$$

则

$$\begin{aligned} & \left\| \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \begin{bmatrix} b \\ d_p \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2^2 = \\ & \left\| \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \left(\begin{bmatrix} b \\ d_p \end{bmatrix} - \begin{bmatrix} b \\ d_p \end{bmatrix} \right) \right\|_2^2 + \\ & \left\| \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \begin{bmatrix} b \\ d_p \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2^2 = \\ & \left\| \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \left(\begin{bmatrix} b \\ d_p \end{bmatrix} - \begin{bmatrix} b \\ d_p \end{bmatrix} \right) \right\|_2^2 + \\ & \left\| \begin{bmatrix} 0 & \mathbf{1}_{/p}^T \\ \mathbf{1} & \overline{K}_{op} \end{bmatrix} \begin{bmatrix} b \\ d_p \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2^2 + \end{aligned}$$

$$2 \left(\begin{bmatrix} b \\ \rho \end{bmatrix} - \begin{bmatrix} b \\ \rho \end{bmatrix} \right)^T \begin{bmatrix} 0 & \mathbf{1}_{/P}^T \\ \mathbf{1} & \overline{K_{OP}} \end{bmatrix}^T \times \left(\begin{bmatrix} 0 & \mathbf{1}_{/P}^T \\ \mathbf{1} & \overline{K_{OP}} \end{bmatrix} \begin{bmatrix} b \\ \rho \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right) = \left\| \begin{bmatrix} 0 & \mathbf{1}_{/P}^T \\ \mathbf{1} & \overline{K_{OP}} \end{bmatrix} \left(\begin{bmatrix} b \\ \rho \end{bmatrix} - \begin{bmatrix} b \\ \rho \end{bmatrix} \right) \right\|_2^2 + \left\| \begin{bmatrix} 0 & \mathbf{1}_{/P}^T \\ \mathbf{1} & \overline{K_{OP}} \end{bmatrix} \begin{bmatrix} b \\ \rho \end{bmatrix} - \begin{bmatrix} 0 \\ d \end{bmatrix} \right\|_2^2 \quad (27)$$

由式(27)可知,改进后的解 $[b \quad \rho]^T$ 对式(25)的优化效果不会低于改进前的解 $[b \quad \rho]^T$. Jiao 法改进前后的最大区别在于,改进前在迭代每一步计算 b 和 ρ 时,仅从数据集 $\{(x_i, d_i)\}_{i \in P}$ 对优化目标函数的影响考虑,完全抛弃了 $\{(x_i, d_i)\}_{i \in Q}$ 产生的约束对目标函数的影响;改进后在迭代每一步计算 b 和 ρ 时,则既要考虑 $\{(x_i, d_i)\}_{i \in P}$ 产生的约束对优化目标函数的影响,也要考虑 $\{(x_i, d_i)\}_{i \in Q}$ 对目标函数的影响. 因此,用改进后 Jiao 法建立的最小二乘支持向量回归机模型的泛化能力比较强,至少不会比改进前差. 改进前的 Jiao 法要达到与改进后一样的泛化能力,只有通过增加支持向量的数目,即考虑其他数据产生的约束对目标函数的影响来实现.

现将改进后算法的实现步骤描述如下:

Step1: 令 $\tilde{b} = \mathbf{0}^T, \tilde{\rho} = 0, r^0 = -d, O = Q = \{1, 2, \dots, N\}, P = \emptyset, \tilde{0} = \sum_{i=1}^N d_i, n = 0;$

Step2: 如果 $Q = \emptyset$ 或 $\max(|r_Q^n|) < \epsilon$, 则停机;

Step3: $s = \arg \min_Q (-r_Q^n)^2 / (2(k(x_i, x_i) + \frac{1}{C}))$;

Step4: 如果 $n = 0$, 则

$$\tilde{R}^{n+1} = \left(\begin{bmatrix} 0 & \mathbf{1} \\ \mathbf{1}_{/O} & \overline{K_{OP}} \end{bmatrix}^T \begin{bmatrix} 0 & \mathbf{1} \\ \mathbf{1}_{/O} & \overline{K_{OP}} \end{bmatrix} \right)^{-1},$$

$$\begin{bmatrix} \tilde{b}^{n+1} \\ \tilde{\rho}^{n+1} \end{bmatrix} = \tilde{R}^{n+1} \begin{bmatrix} \tilde{0} \\ d \end{bmatrix};$$

Step5: 如果 $n > 0$, 则根据式(17) ~ (22) 计算 $\tilde{R}^{n+1}, \tilde{P}^{n+1}, \tilde{S}^{n+1}$ 和 b^{n+1} ;

Step6: $Q = Q - \{s\}, P = P + \{s\}$;

Step7: $r_Q^{n+1} = K_{QP} \tilde{P}^{n+1} + \tilde{b}^{n+1}, n = n + 1$, 转 Step2.

5 实例仿真

5.1 人工数据集

墨西哥草帽函数 MHF

$$y = f(x_1, x_2) = \frac{\sin(\sqrt{x_1^2 + x_2^2})}{\sqrt{x_1^2 + x_2^2}} + \dots$$

其中: $\epsilon \sim N(0, 0.1^2)$ 为噪声, $C = 14$, 选择的是 RBF 核

$$k(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}.$$

其中: 核参数是通过交叉验证^[16,17]取得的, $\sigma = 0.90$, 训练数据集的大小为 20×20 , 测试数据集的大小为 23×23 . 定义 1 个性能指标 (Root mean squared error)

$$RMSE = \sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2}.$$

其中: N_{test} 代表测试数据的个数, y_i 代表目标值, \hat{y}_i 代表模型的预测值. 采用基于改进 Jiao 法的稀疏最小二乘支持向量回归机, 仿真效果如图 1 所示.

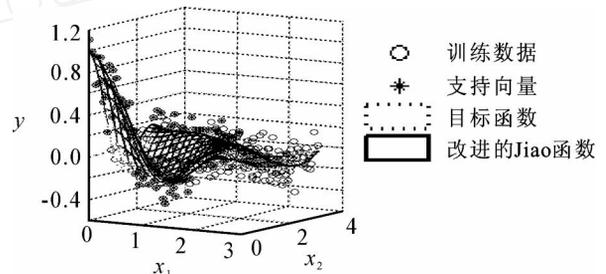


图 1 用 FLS-SVRM 对 MHF 的回归结果

各种剪枝算法仿真结果的对比如表 1 所示.

表 1 MHF 回归结果的比较

方 法	RMSE	teTime/s	trTime/s	NSV
Suykens 法	0.0246	14.09	262.4	252
Kruif 法	0.0247	11.98	807.5	205
Hoegaerts 法	0.0238	15.02	717.8	252
Zeng 法	0.0246	11.70	1224.7	196
Jiao 法	0.0236	23.33	39.4	384
Jiao 法	0.037	10.20	19.4	167
改进 Jiao 法	0.0217	9.33	17.8	150

表 1 中: teTime 代表预测时间(在 Matlab6.5, CPU Celeron(R) 1.70 GHz, 256M 内存, 80G 硬盘环境下测得), trTime 代表训练时间, NSV 代表支持向量个数. 表 1 中第 6 和 7 两行是用 Jiao 法选择不同个数的支持向量时得到的结果.

5.2 UCI 数据集^[18]

UCI 数据集是检测机器学习效果的标准数据集, 本文采用 4 个 UCI 数据集, 现将各个数据集描述如下:

1) machineCPU dataset. 此数据集有 209 个样本, 6 个属性, 用 150 个样本进行训练, 59 个样本进行测试;

2) Auto-MPG dataset. 此数据集有 392 个样本,7 个属性,用 350 个样本进行训练,42 个样本进行测试;

3) Boston housing dataset. 此数据集有 506 个样本,13 个属性,用 400 个样本进行训练,106 个样本进行测试;

4) Abalone dataset. 此数据集有 4177 个样本,8 个属性,用 2500 个样本进行训练,1677 个样本进行测试.

用各种剪枝算法建立稀疏最小二乘支持向量回归机对 UCI 数据集的学习效果如表 2 ~ 表 5 所示.

表 2 machineCPU 数据集测试的结果

方 法	RMSE	teTime/ s	trTime/ s	NSV
Suykens 法	37.65	0.48	23.03	73
Kruif 法	37.24	0.42	36.17	63
Hoegaerts 法	37.31	0.42	37.38	60
Zeng 法	38.03	0.67	1734.50	96
Jiao 法	36.76	0.73	4.11	109
Jiao 法	44.35	0.42	2.67	59
改进 Jiao 法	35.56	0.42	2.55	58

表 3 AutoMPG 数据集测试的结果

方 法	RMSE	teTime/ s	trTime/ s	NSV
Suykens 法	2.25	0.52	103.42	101
Kruif 法	2.19	0.41	402.83	86
Hoegaerts 法	2.23	0.52	414.10	96
Zeng 法	2.22	0.80	969.11	146
Jiao 法	2.28	1.55	27.00	289
Jiao 法	3.28	0.61	11.64	111
改进 Jiao 法	2.35	0.61	11.83	112

表 4 Boston housing 数据集测试的结果

方 法	RMSE	teTime/ s	trTime/ s	NSV
Suykens 法	2.70	2.23	143.9	185
Kruif 法	2.76	2.36	625.6	195
Hoegaerts 法	2.64	2.19	638.4	176
Zeng 法	2.80	3.20	3253.1	256
Jiao 法	2.77	4.92	39.7	381
Jiao 法	3.49	2.86	25.4	227
改进 Jiao 法	2.71	2.80	25.9	220

表 5 Abalone 数据集测试的结果

方 法	RMSE	teTime/ s	trTime/ s	NSV
Jiao 法	2.17	206.09	894.03	979
Jiao 法	2.58	85.66	372.72	442
改进 Jiao 法	2.09	88.23	396.64	424

示.

由于其他几种剪枝算法测试时间太长,在表 5 中未列出,但这并不影响 Jiao 法改进前后效果的比较.

由表 1 ~ 表 5 可知,本文将 Jiao 法用在最小二乘支持向量回归机所取得的效果并不明显,而采用改进 Jiao 法则取得了较好的效果.这主要表现在以下 2 方面:1) 由表 2 ~ 表 4 第 6 和 8 行(表 5 第 1 和 3 行)可知,在相同的回归精度下,改进 Jiao 法需要较少的支持向量和较短的训练时间,这主要是因为,在计算 \tilde{K} 和 \tilde{b} 时,充分考虑了非支持向量产生的约束对目标函数的影响,以达到增强最小二乘支持向量回归机泛化能力的目的,改进前要达到与改进后一样的泛化能力,只能通过增加支持向量的数目来实现;2) 由表 2 ~ 表 4 第 7 和 8 行(表 5 第 2 和 3 行)可知,在选择基本相同个数的支持向量时,改进 Jiao 法的回归精度高,而且不增加训练时间.出现这样的结果一是充分考虑了非支持向量产生的约束对优化目标函数的影响,增强了模型的泛化能力;二是看似计算 \tilde{K} 和 \tilde{K}_s 浪费了时间,但在后面计算 K_{QP} 时,可充分利用计算 \tilde{K} 和 \tilde{K}_s 时 $\overline{K_{OP}}$ 中的信息(不需再计算 K_{QP} ,节省了计算量),从而保持训练时间基本不变.同时,本文提出的改进 Jiao 法与其他方法相比,在预测精度基本保持不变的情况下,大大缩短了训练时间.

另外,将本文改进的 Jiao 法推广到分类问题,同样可以取得与回归相似的效果.在分类精度几乎不变的情况下,改进 Jiao 法需要较少的支持向量和较短的训练时间;在支持向量数目基本相同的情况下,改进 Jiao 法的分类精度高.作者用 4 个 UCI 数据集(wine, glass, waveform, waveform + noise)验证了改进 Jiao 法的分类效果.

6 结 论

将 Jiao 法应用于最小二乘支持向量回归机,回归效果并不理想.本文在此基础上提出了改进的 Jiao 法,并将其应用于最小二乘支持向量回归机,在回归精度基本不变的情况下,支持向量数目和训练时间都取得了明显的优势.在支持向量数目几乎相同的情况下,改进 Jiao 法的回归精度高,并保持了训练时间基本不变.与其他剪枝算法相比较,在预测精度基本不变的前提下,本文提出的改进 Jiao 法大大缩短了训练时间,这无疑是一种快速稀疏最小二乘支持向量回归机.另外,将改进的 Jiao 法推广到分类问题,可取得与回归相似的结果.

参考文献(References)

- [1] Vapnik V N. The nature of statistical learning theory [M]. New York: Springer-Verlag, 1995.
- [2] Meng Z M, Peng L F, Zhou G G, et al. A multi-classification method of temporal data based on support vector machine [C]. Proc of LNAI. Berlin: Springer-Verlag, 2007: 240-249.
- [3] Huang C L, Chen M C, Wang C J. Credit scoring with a data mining approach based on support vector machines[J]. Expert Systems with Application, 2007, 33(4): 847-856.
- [4] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999, 9(3): 293-300.
- [5] Suykens J A K, Van Gestel T, De Brabanter J, et al. Least squares support vector machines[M]. Singapore: World Scientific, 2002.
- [6] Gao J B, Shi D, Liu X M. Significant vector learning to construct sparse kernel regression models[J]. Neural Networks, 2007, 20(7): 791-798.
- [7] Suykens J A K, Brabanter J D, Lukas L, et al. Weighted least squares support vector machines: Robustness and sparse approximation[J]. Neurocomputing, 2002, 48(1-4): 85-105.
- [8] Suykens J A K, Lukas L, Vandewalle J. Sparse approximation using least squares support vector machines[C]. Proc of IEEE Int Symposium on Circuits and System. Piscataway: IEEE, 2000: 757-760.
- [9] Kruijff B J, Vries J A. Pruning error minimization in least squares support vector machines[J]. IEEE Trans on Neural Networks, 2003, 14(3): 696-702.
- [10] Hoegaerts L, Suykens J A K, Vandewalle J, et al. A comparison of pruning algorithms for sparse least squares support vector machines[C]. Proc of Int Conf on Neural Information Processing 2004. Berlin: Springer-Verlag, 2004: 1247-1253.
- [11] Platt J. Fast training of support vector machines using sequential minimal optimization[C]. Proc of Advances in Kernel Methods-Support Vector Machine. Cambridge: MIT Press, 1999: 185-208.
- [12] Keerthi S S, Shevade S K. SMO algorithm for least squares SVM[C]. Proc of the Int Joint Conf on Neural Networks 2003. Piscataway: IEEE, 2003: 2088-2093.
- [13] Zeng X Y, Chen X W. SMO-based pruning methods for sparse least squares support vector machines[J]. IEEE Trans on Neural Networks, 2005, 16(6): 1541-1546.
- [14] Jiao L C, Bo L F, Wang L. Fast sparse approximation for least squares support vector machine [J]. IEEE Trans on Neural Networks, 2007, 18(3): 685-697.
- [15] 黄红选, 韩继业. 数学规划[M]. 北京: 清华大学出版社, 2006.
(Hang H X, Han J Y. Mathematical programming [M]. Beijing: Tsinghua University Press, 2006.)
- [16] Zhao Y, Keong K C. Fast leave-one-out evaluation and improvement on inference for LS-SVMs[C]. Proc of the 17th Int Conf on Pattern Recognition. Cambridge: IEEE, 2004: 494-497.
- [17] An S J, Liu W Q, Venkatesh S. Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression[J]. Pattern Recognition, 2007, 40(8): 2154-2162.
- [18] Murphu P M, Aha D W. UCI repository of machine learning database [Z]. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

(上接第 1346 页)

- [14] 车录锋, 周晓军, 徐志农, 等. 可拓方法在多传感器信息融合工件识别中的应用[J]. 系统工程理论与实践, 2000, 20(8): 91-94.
(Che L F, Zhou X J, Xu Z R. et al. Application of extension method in multi-sensor data fusion for parts recognition [J]. System Engineering Theory and Practice, 2000, 20(8): 91-94.)
- [15] Hwang C L, Yoon K. Multiple attribute decision making methods and applications [M]. New York: Springer, 1981.
- [16] 邱菀华. 管理决策与应用熵学[M]. 北京: 机械工业出版社, 2002.
(Qiu W H. Management decision and applied entropy [M]. Beijing: China Machine Press, 2002.)