非线性回归支持向量机的 SMO 算法改进

赵长春 姜晓爱 金英汉

(北京航空航天大学 航空科学与工程学院, 北京 100191) (西北工业大学 航天学院, 西安 710072)

摘 要: 为了解决非线性数据和非线性函数的回归问题,采用了支持向量机序列最小优化算法.原始序列最小优化(SMO,Sequential Minimal Optimization)算法存在训练速度慢和训练结果不稳定的缺点,为了能加快 SMO 算法的训练速度和提高训练结果稳定性,通过改进优化乘子更新方法、采用双阈值法、预存核函数、增加停机准则等方法对 SMO 算法做了改进.仿真实验表明,改进的算法能很好地对非线性数据和非线性函数进行回归,具有比原始 SMO 算法更快的训练速度和稳定的训练结果.

关 键 词: 支持向量机; 回归; 非线性数据; 非线性函数; 序列最小优化算法

中图分类号: TP 301.6

文献标识码: A **文 章 编 号**: 1001-5965(2014)01-0125-06

Improved SMO algorithm of nonlinear regression support vector machine

Zhao Changchun Jiang Xiaoai

(School of Aeronautic Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Jin Yinghan

(School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: In order to solve the regression problems of nonlinear data and nonlinear function, the support vector machine (SVM) sequential minimal optimization (SMO) algorithm was adopted. The original SMO algorithm has deficiencies such as low training speed and instability training results. To accelerate the training process of SMO algorithm and promote training stability of the solution, the SMO algorithm was improved by updating the optimization multipliers method, using double threshold values, caching kernel function outputs, adding stop criterion. Simulation results show that the improved algorithm performs well for regression of nonlinear data and nonlinear function, and it has faster training speed and better training result stability than original SMO algorithm.

Key words: support vector machine; regression; nonlinear data; nonlinear function; sequential minimal optimization (SMO) algorithm

支持向量机(SVM,Support Vector Machine)理论是 Vapnik 等人提出的一种新型的机器学习算法,在分类问题和回归问题上应用较广泛.由于 SVM 训练中需要求解二次规划问题,在训练大规模数据时占用内存空间较大,文献[1]提出了序列最小优化(SMO,Sequential Minimal Optimization)算法,该算法将问题分解为每次优化两个样本的 Lagrange 乘子,避免了求解二次规划问题,提高了训练速度.文献[2]详细介绍了 SMO 回归算法的实现方法,由于该算法训练时间较长,出现了许多对 SMO 算法的改进方法[3-5],以缩短训练时间.此外,支持向量机的参数选择对训练模型的精度和训练速度影响较大,通过选择最优的支持向量机参数可以提高训练模型的准确度和训练效率[6-7].参数优化方法的实质是利用测试样本对训练模型测试的精度来判断是否得到最优参数,

收稿日期: 2013-01-28;**网络出版时间**: 2013-05-07 11:38

网络出版地址: www.cnki.net/kcms/detail/11.2625.V.20130507.1138.002.html

作者简介: 赵长春(1987-),男,四川绵阳人,硕士生,zhaochangchun_buaa@163.com.

参数寻优的过程就是不断改变参数值和用新参数反复训练模型的过程.因此,研究提高算法自身的训练速度的方法,能在很大程度缩短训练时间.

本文对SMO算法的改进方法进行了研究,提出新的改进算法用于非线性数据和非线性函数的回归,通过改进优化乘子更新方法、采用双阈值法、预存核函数、增加停机准则等方法加快了训练速度,提高了训练结果稳定性,最后对支持向量机参数选择方法做了讨论.

1 支持向量机回归

支持向量机回归的求解方法是将原始最小化问题转化为求其对偶最大化问题,对于样本集(x_i,y_i),$i=1,2,\cdots,l$,求解的对偶问题[2]可表示如下:

$$\max\left[-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(x_i,x_j)+\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)y_i-\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\varepsilon\right] \tag{1}$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0\\0\leqslant\alpha_i\leqslant C\\0\leqslant\alpha_i^*\leqslant C\\i=1,2,\cdots,l\end{cases} \tag{2}$$

式中,ε为不敏感损失函数(示意图见图1);C为规则化参数,用于平衡回归函数的平坦程度;α_i,α_i^*为拉格朗日乘子;$k(x_i,x_j)=\Phi(x_i)\cdot\Phi(x_j)$,记为$k_{ij}$,为满足Mercer条件的核函数,$\Phi(\cdot)$为非线性映射.常用的核函数有线性核函数、多项式核函数、径向基核函数、Sigmoid核函数等.

回归函数可表示为

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_i,x_j)+b \tag{3}$$

式中,b为阈值.

图1中,ε用于控制回归逼近误差范围,$\xi\geqslant0$为松弛因子,表示回归函数值$f(x_i)$与样本值y_i之差的绝对值$|f(x_i)-y_i|$,当绝对值小于或等于ε时,取0,当绝对值大于ε时,取ξ.

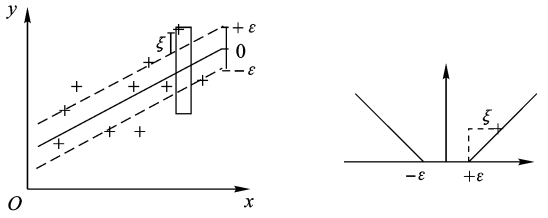


图1 ε不敏感损失函数示意图

2 回归SMO算法理论

作变量代换,令

$$\left.\begin{aligned}\bar{\alpha}_i&=\alpha_i-\alpha_i^*\\|\bar{\alpha}_i|&=\alpha_i+\alpha_i^*\end{aligned}\right\} \tag{4}$$

SMO算法每次只优化两个变量,固定其他变量值,设2个待优化变量分别为$\bar{\alpha}_u$,$\bar{\alpha}_v$,优化目标函数可表示成关于$\bar{\alpha}_u$,$\bar{\alpha}_v$的函数:

$$L(\bar{\alpha}_u,\bar{\alpha}_v)=\frac{1}{2}\bar{\alpha}_u^2k_{uu}+\frac{1}{2}\bar{\alpha}_v^2k_{vv}+\bar{\alpha}_u\bar{\alpha}_vk_{uv}+\varepsilon(|\bar{\alpha}_u|+|\bar{\alpha}_v|)-\bar{\alpha}_uy_u-\bar{\alpha}_vy_v+\bar{\alpha}_uz_u^{\text{old}}+\bar{\alpha}_vz_v^{\text{old}}+L_0 \tag{5}$$

式中,$z_i^{\text{old}}=f_i^{\text{old}}-\bar{\alpha}_u^{\text{old}}k_{ui}-\bar{\alpha}_v^{\text{old}}k_{vi}-b^{\text{old}}$,old标识的变量指本次迭代初始值;$L_0$是不含$\bar{\alpha}_u$,$\bar{\alpha}_v$的项,即在当前2个变量更新过程中恒为常数.由SMO算法原理[1]可知,$\bar{\alpha}_u$与$\bar{\alpha}_v$之和保持不变,即

$$s^{\text{old}}=\bar{\alpha}_u^{\text{old}}+\bar{\alpha}_v^{\text{old}}=\bar{\alpha}_u+\bar{\alpha}_v \tag{6}$$

将式(6)代入式(5)得到关于$\bar{\alpha}_v$的函数,记

$$\mathrm{d}|x|/\mathrm{d}x=\operatorname{sgn}x$$

对$\bar{\alpha}_v$求导可得

$$\partial L(\bar{\alpha}_v)/\partial\bar{\alpha}_v=(\bar{\alpha}_v-s^{\text{old}})k_{uu}+\bar{\alpha}_vk_{vv}+(s^{\text{old}}-2\bar{\alpha}_v)k_{uv}+\varepsilon(\operatorname{sgn}\bar{\alpha}_v-\operatorname{sgn}(s^{\text{old}}-\bar{\alpha}_v))+y_u-y_v-z_u^{\text{old}}+z_v^{\text{old}} \tag{7}$$

令$\partial L(\bar{\alpha}_v)/\partial\bar{\alpha}_v=0$,可推出$\bar{\alpha}_v$的更新公式:

$$\bar{\alpha}_v=\bar{\alpha}_v^{\text{old}}+[y_v-y_u+f_u^{\text{old}}-f_v^{\text{old}}+\varepsilon(\operatorname{sgn}\bar{\alpha}_v-\operatorname{sgn}(s^{\text{old}}-\bar{\alpha}_v))]/\eta \tag{8}$$

式中,$\eta=k_{uu}-2k_{uv}+k_{vv}$,当核函数满足Mercer条件时,$\eta\geqslant0$成立.

式(8)存在5个可能的解,依据$\operatorname{sgn}\bar{\alpha}_v-\operatorname{sgn}(s^{\text{old}}-\bar{\alpha}_v)$分别取值为$-2$,0,2的情况有3个可能的解,另2个可能的解是$\bar{\alpha}_v$取0和$s^{\text{old}}$.

3 SMO算法改进

3.1 采用双阈值b_{low}和b_{up}

文献[3]中指出,由于阈值b正确性不确定,使得在优化中对已达到最优值的乘子$\bar{\alpha}_i$更新,浪费执行时间,因此采用双阈值b_{low}和b_{up}的方法对算法做了改进.

令 $F_i = y_i - \langle \omega, \Phi(x_i) \rangle$,其中$\langle \cdot, \cdot \rangle$ 表示内积,$\omega = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \Phi(x_i)$,有以下 5 种情况:

$$\left.\begin{array}{ll} -\varepsilon \leqslant (F_i - b) \leqslant \varepsilon & \alpha_i = \alpha_i^* = 0 \\ \varepsilon \leqslant (F_i - b) & \alpha_i = C \\ (F_i - b) \leqslant -\varepsilon & \alpha_i^* = C \\ (F_i - b) = \varepsilon & 0 < \alpha_i < C \\ (F_i - b) = -\varepsilon & 0 < \alpha_i^* < C \end{array}\right\} \quad (9)$$

根据式(9)的情况可定义 $\bar{\alpha}_i$ 集合:$I_{0a} = \{i: 0 < \bar{\alpha}_i < C\}$;$I_{0b} = \{i: -C < \bar{\alpha}_i < 0\}$;$I_1 = \{i: \bar{\alpha}_i = 0\}$;$I_2 = \{i: \bar{\alpha}_i = -C\}$;$I_3 = \{i: \bar{\alpha}_i = C\}$. $I_0 = I_{0a} \cup I_{0b}$,定义 $\tilde{F}_i$ 和 $\bar{F}_i$ 如下:

$$\left.\begin{array}{ll} \tilde{F}_i = F_i + \varepsilon & i \in I_{0b} \cup I_2 \\ \tilde{F}_i = F_i - \varepsilon & i \in I_{0a} \cup I_1 \\ \bar{F}_i = F_i - \varepsilon & i \in I_{0a} \cup I_3 \\ \bar{F}_i = F_i + \varepsilon & i \in I_{0b} \cup I_1 \end{array}\right\} \quad (10)$$

根据式(9)和式(10)可定义 b 的上、下界,如下:

$$\left.\begin{array}{ll} b_{\text{low}} = \max\{\tilde{F}_i\} & i \in I_0 \cup I_1 \cup I_2 \\ b_{\text{up}} = \min\{\bar{F}_i\} & i \in I_0 \cup I_1 \cup I_3 \end{array}\right\} \quad (11)$$

当 $\bar{\alpha}_i$ 满足最优条件时,如下关系式成立:

$$b_{\text{low}} \leqslant b_{\text{up}} \quad (12)$$

由于数值解原因,通常不可能获得准确的优化结果,因此,通过增加一项正误差 τ 获取近似优化结果,重新写式(12),如下:

$$b_{\text{low}} \leqslant b_{\text{up}} + 2\tau \quad (13)$$

查找违反最优条件式(13)的样本作为待优化样本,即满足如下条件之一的样本为待优化样本:

$$\left.\begin{array}{l} \tilde{F}_i > \bar{F}_j + 2\tau \\ \qquad i \in I_0 \cup I_1 \cup I_2; j \in I_0 \cup I_1 \cup I_3 \\ \bar{F}_i < \tilde{F}_j - 2\tau \\ \qquad i \in I_0 \cup I_1 \cup I_3; j \in I_0 \cup I_1 \cup I_2 \end{array}\right\} \quad (14)$$

3.2　优化乘子更新

文献[4]指出用 $\operatorname{sgn} \bar{\alpha}_v^{\text{old}} - \operatorname{sgn}(s^{\text{old}} - \bar{\alpha}_v^{\text{old}})$ 代替 $\operatorname{sgn} \bar{\alpha}_v - \operatorname{sgn}(s^{\text{old}} - \bar{\alpha}_v)$ 计算是多数 SMO 算法中的错误,并对乘子的更新方法做了改进.

由于 $F_u - F_v$ 的值与 b 无关,通过其更新乘子可避免乘子优化结果受不准确的 b 值的影响,乘子更新步骤如下:

1) 更新优化乘子分下面 3 个步骤.

① 按如下式子进行更新:

$$\left.\begin{array}{l} \bar{\alpha}_v = \bar{\alpha}_v^{\text{old}} - (F_u - F_v)/\eta \\ \bar{\alpha}_u = s^{\text{old}} - \bar{\alpha}_v \end{array}\right\} \quad (15)$$

② 如果更新后两个乘子符号相同则得到本次更新结果,反之,定义调节变量 $\Delta = 2\varepsilon/\eta$,如果根据 Δ 更新后的乘子 $\bar{\alpha}_u$ 和 $\bar{\alpha}_v$ 符号不改变,按式(16)进行:

$$\bar{\alpha}_v = \bar{\alpha}_v^{\text{old}} - \operatorname{sgn} \bar{\alpha}_v \cdot \Delta \quad (16)$$

③ 如果更新后的乘子符号发生改变,则设置乘子为 0 或 s^{old},按如下条件进行:

$$\bar{\alpha}_v = \begin{cases} s^{\text{old}} & \bar{\alpha}_v > 0; \bar{\alpha}_u < 0; s^{\text{old}} \geqslant 0 \\ 0 & \bar{\alpha}_v > 0; \bar{\alpha}_u < 0; s^{\text{old}} < 0 \\ s^{\text{old}} & \bar{\alpha}_v < 0; \bar{\alpha}_u > 0; s^{\text{old}} < 0 \\ 0 & \bar{\alpha}_v < 0; \bar{\alpha}_u > 0; s^{\text{old}} \geqslant 0 \end{cases} \quad (17)$$

2) 约束优化乘子范围.

确保变量在 $\pm C$ 之间,根据下面式子修正 $\bar{\alpha}_u$ 与 $\bar{\alpha}_v$,确保变量在区间$[L, H]$之内,定义如下:

$$\left.\begin{array}{l} L = \max(s^{\text{old}} - C, -C) \\ H = \min(s^{\text{old}} + C, C) \end{array}\right\} \quad (18)$$

3.3　停机准则

目标问题与其对偶问题的对偶间隙为 0 时,达到全局最优解,但数值求解只能得到近似全局最优解,因此将对偶间隙作为停机准则,设定训练精度在达到预定值后停止计算[8-9],减少训练时间.

定义如下关系:

$$\left.\begin{array}{l} \xi_i = \max(0, y_i - \langle \omega, \Phi(x_i) \rangle - b - \varepsilon) \\ \xi_i^* = \max(0, \langle \omega, \Phi(x_i) \rangle + b - y_i - \varepsilon) \end{array}\right\} \quad (19)$$

记目标优化问题为函数 $U(\omega, \xi, \xi^*)$,对偶问题为函数 $W(\bar{\alpha})$,则函数表达式为

$$U(\omega, \xi, \xi^*) = \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \quad (20)$$

$$W(\bar{\alpha}) = \frac{1}{2} \sum_{i,j=1}^{l} \bar{\alpha}_i \bar{\alpha}_j k(x_i, x_j) - \sum_{i=1}^{l} | \bar{\alpha}_i | y_i + \sum_{i=1}^{l} \bar{\alpha}_i \varepsilon \quad (21)$$

停机准则计算式为

$$\sigma = \lg\left(\frac{U(\omega,\xi,\xi^*) - W(\bar{\alpha})}{U(\omega,\xi,\xi^*) + 1}\right) \tag{22}$$

当 σ 小于给定精度后停止训练,能缩短训练时间,将 σ 停机准则和式(14)的违反条件共同判断停止训练条件.

3.4 核函数矩阵计算和存储

在训练过程中,为避免重复计算核函数,训练开始时进行核函数矩阵存储. 核函数矩阵是对称矩阵,存储时只需存储对称部分,假设有 N 个样本,则可用大小 $N(N+1)/2$ 的一维数组 a_J 存储核函数矩阵,从主对角元素开始按主对角线方向顺序存储矩阵元素,如图 2 所示,元素 k_{mn} 在矩阵中的位置与在数组 a_J 中位置对应关系如下:

$$J = \left[N + \frac{1-(n-m)}{2}\right](n-m) + m \tag{23}$$

$$m,n = 1,2,\cdots,N; J = 1,2,\cdots,N/(N+1)/2$$

式中,J 为一维数组元素的下标;m,n 分别为核函数矩阵元素 k_{mn} 的行号和列号. 核函数矩阵对称,只需计算主对角线和其上方的元素,其下方的元素与上方元素对称. 规定 $m \leqslant n$,如果 $m > n$,交换 m 和 n 的值后再计算.

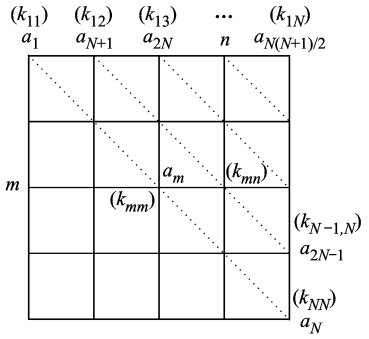


图 2 核矩阵元素存储

3.5 改进的 SMO 算法描述

改进后的 SMO 算法具体步骤如下:

1) 初始化 $\bar{\alpha}_i$,b 为 0,计算核函数矩阵并存储为一维数组.

2) 从整个样本中随机选择一个样本 i,令 $b_{\text{low}} = y_i - \varepsilon$,$b_{\text{up}} = y_i + \varepsilon$,$i_{\text{low}} = i$,$i_{\text{up}} = i$.

3) SMO 算法采用内、外两层循环选择待优化样本,步骤如下:

① 外循环查找第 1 个待优化样本,样本乘子为 $\bar{\alpha}_v$. 首先遍历所有非边界样本,查找违反最优条件的样本作为第 1 个待优化样本;如果没有可以优化的样本,则遍历所有样本;如果两次遍历查找都没有可以优化的样本,则转 6).

② 内循环查找第 2 个待优化样本,样本乘子为 $\bar{\alpha}_u$. 选择违反最优条件最严重的作为第 2 个待优化样本,例如,选择 $i_2 = i_{\text{low}}$,$i_1 = i_{\text{up}}$.

4) 优化乘子更新步骤:采用 3.2 节乘子更新法更新优化样本的乘子. 如果优化乘子的改变量大于一定值,则更新优化样本乘子和界内样本的误差值 $F_i = y_i - \langle \omega, \Phi(x_i) \rangle$,并按式(11)在界内样本中寻找新的 $b_{\text{low}}, b_{\text{up}}, i_{\text{low}}, i_{\text{up}}$.

5) 在界内样本或所有样本遍历循环完成若干次迭代后计算一次停机准则 σ 值.

6) 为防止优化程序出现死循环,当满足以下 3 个条件之一就结束算法:优化成功的样本数量为 0、停机准则 σ 小于给定精度、迭代次数大于给定次数;如果没有满足条件的,则转 3).

4 仿真实验及结果分析

将改进的算法与原始的 SMO 算法做对比,实验使用两个不同的拟合数据. 通过对比达到相同停机准则时的时间和测试均方根误差(RMSE, Root Mean Square Error)来评价两个算法的优越性,RMSE 表达式为

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{24}$$

式中,N 为测试样本总数;y_i 为样本目标值;$\hat{y}_i$ 为回归值.

4.1 仿真实验 1

选取文献[10]中的训练样本作为算例,该数据具有非线性特点,样本如下:

$$\boldsymbol{X} = [1.0;3.0;4.0;5.6;7.8;10.2;11.0;11.5;12.7]$$

$$\boldsymbol{Y} = [-1.6;-1.8;-1.0;1.2;2.2;6.8;10.0;10.0;10.0]$$

所选择的测试样本如下:

$$\boldsymbol{X}_{\text{t}} = [2.0;4.0;6.0;8.0;10.0;12.0]$$

$$\boldsymbol{Y}_{\text{t}} = [-1.74;-0.7;1.22;2.7;6.6;9.9]$$

选用径向基核函数,表达式如下:

$$K_{ij} = \exp(-\|x_i - x_j\|^2/2\gamma^2) \tag{25}$$

参数设置基本原则:优化的参数对 (C,γ^2) 的组合有很多[7],样本数据变动趋势较平坦时,γ^2 取较大值,常取大于 1 的值;反之,取较小值,常取小于 1 的值;C 值用于平衡回归函数平坦程度,一般取样本均值附近的值;ε 控制回归函数逼近误差范围,根据实际需求取值;较小的 τ 值会增加训练时间,取值应当比 ε 取值小 1 ~ 2 个数量级;σ 取值决定了训练精度,不宜过小,否则会增加训练时间.

根据上述基本原则和多次实验设置参数:$C=4,\gamma^2=2.1,\varepsilon=0.3,\tau=0.0001$. 采用在不同停机准则值下所用的训练时间和 RMSE 值作对比,表 1为 2 种算法的对比.

表 1　2 种算法比较

停机准则 σ 值设置	原始 SMO 算法			改进 SMO 算法		
	耗时/s	均方根误差值	训练结束时的 σ 值	耗时/s	均方根误差值	训练结束时的 σ 值
-2	3.16	0.04928	-2.127	0.15	0.03166	-2.263
-3	4.7	0.02764	-3.036	0.26	0.02396	-3.046
-4	5.97	0.02640	-3.559	0.45	0.02657	-4.242
-5	7.1	0.02619	-3.492	0.53	0.02644	-3.978

从表 1 可以看出,与原始算法相比,改进后的算法训练时间明显缩短,但随着停机准则 σ 值减小而增加,因为停机准则 σ 值越小表示对偶间隙值越小,则训练时间会增加;2 种算法的 RMSE 值随停机准则 σ 值减小而减小,在相同训练精度条件下,改进的算法能有效加快训练速度.

用改进的 SMO 算法训练非线性数据的回归结果如图 3 所示,从图中可看出,SMO 算法对非线性数据拟合效果较好.

4.2　仿真实验 2

选择函数进行拟合,函数表达式如下:

$$y(x) = 2\sin x + x \qquad (26)$$

产生 100 个随机数点 X,代入式(26)得到 Y,组成训练样本(X,Y),产生 100 个点随机数 X_t,代入式(26)得到 Y_t,组成测试样本(X_t,Y_t).

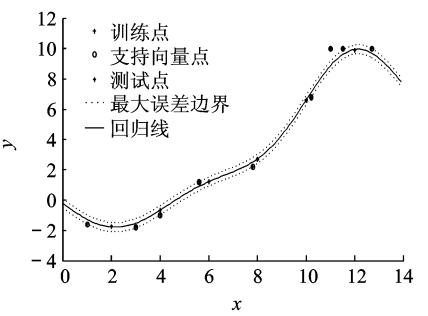


图 3　非线性数据回归结果($C=4,\gamma^2=2.1$)

选用径向基核函数,根据参数选择基本原则和多次实验设置参数:$\sigma=-2,\varepsilon=0.01,\tau=0.001$. 分别固定 C 值和固定 γ^2 值,在这 2 种情况下对比 2 种算法的支持向量数目、训练时间、RMSE 值.

从表 2、表 3 可以看出,能得到较小 RMSE 值的最佳 C 值为 1 ~ 1.5 之间,最佳 γ^2 值为 0.1;选取最佳 C 和 γ^2 时,2 种算法都能得到较小的 RMSE 值和较少的支持向量,较小的 γ^2 值和较大的 C 值都会增加训练时间;与原始 SMO 算法相比,在达到相同的训练精度条件下,改进的算法明显缩短训练时间;原始算法使用了随机选择优化样本的方法,所以训练结果不稳定,同时也增加了训练时间,而改进后的算法使训练结果具有稳定性,使训练速度更快.

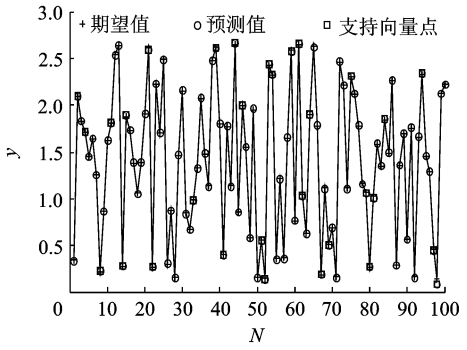
用改进的 SMO 算法对非线性函数回归的结果见图 4,从图中可看出,SMO 算法对非线性函数回归效果较好,随机测试样本测试误差较小.

表 2　固定 $C=1.2$ 情况下 2 种算法比较

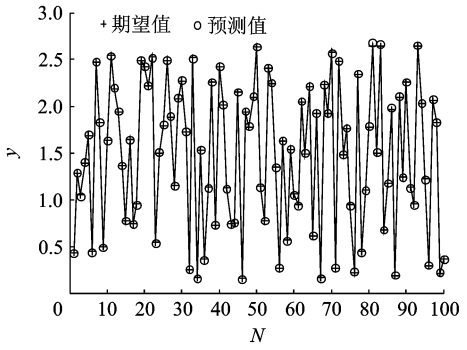
γ^2 值	原始 SMO 算法				改进 SMO 算法			
	支持向量	耗时/s	均方根误差值	训练结束时的 σ 值	支持向量	耗时/s	均方根误差值	训练结束时的 σ 值
0.01	73	414.8	0.1658	-2.141	79	31.9	0.1268	-2.067
0.05	43	156.1	0.0119	-2.105	41	14.9	0.0481	-2.075
0.1	23	223.8	0.0283	-2.079	31	8.99	0.0185	-2.118
0.5	37	89.5	0.0094	-2.010	2	0.12	0.2340	-0.044
1	63	37.3	0.0151	-2.016	4	0.17	0.3726	-0.057
5	100	15.3	0.5018	-1.409	12	0.40	0.7236	-0.090
10	100	11.3	0.7155	-1.298	14	0.42	0.7871	-0.102

表 3　固定 $\gamma^2=0.1$ 情况下 2 种算法比较

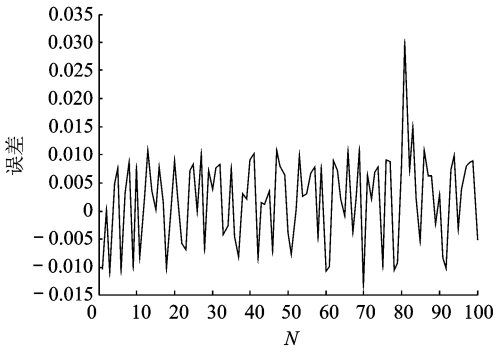
C 值	原始 SMO 算法				改进 SMO 算法			
	支持向量	耗时/s	均方根误差值	训练结束时的 σ 值	支持向量	耗时/s	均方根误差值	训练结束时的 σ 值
0.1	54	29.8	0.1026	-2.036	6	0.23	0.6492	-0.115
0.5	30	112.0	0.0333	-2.163	4	0.18	0.3246	-0.079
0.8	25	122.8	0.0126	-2.067	29	9.62	0.0288	-2.003
1	25	120.8	0.0276	-2.205	28	11.2	0.0119	-2.072
1.5	25	93.4	0.0269	-2.005	25	28.8	0.0139	-2.076
5	23	135.1	0.0095	-2.019	22	71.1	0.0088	-2.084
15	17	130.7	0.0130	-2.014	16	121.4	0.0102	-0.472



a 训练结果



b 测试结果



c 测试样本测试误差

图 4 非线性函数回归结果($C=1.2$,$\gamma^2=0.1$)

5 结 论

通过仿真实验,验证了支持向量机算法具有较强的非线性数据拟合功能,改进的 SMO 算法在训练速度和结果稳定性上得到提高.

实验表明,参数 C 和 γ^2,停机准则 σ 值对训练模型精度和训练速度影响较大,参数选择具有数据依赖性,仿真实验 1 中数据分布值域广,数据变化趋势较平坦,则 γ^2 取较大值,仿真实验 2 中数据分布值域窄,数据变化较剧烈,则 γ^2 取较小值,参数 C 取值大致位于最大值和最小值的中间;停机准则σ值越小,训练时间越长,精度越高,可根据训练要求取不同值,如何更好地选择参数 C 和 γ^2 是值得继续研究的方向.

参考文献(References)

[1] Platt J C. Fast training of support vector machines using sequential minimal optimization [R]. MSR-TR-98-14,1998

[2] Smola A J,Scholkopf B. A tutorial on support vector regression [R]. NC2-TR-1998-030,1998

[3] Shevade S K,Keerthi S S,Bhattacharyya C,et al. Improvements to SMO algorithm for SVM regression [J]. IEEE Transactions on Neural Networks,2000,11(5):1188 – 1193

[4] Flake G W,Lawrence S. Efficient SVM regression training with SMO [J]. Machine Learning,2002,46(1-3):271 – 290

[5] 张浩然,韩正之. 回归支持向量机的改进序列最小优化学习算法 [J]. 软件学报,2003,14(12):2006 – 2013
Zhang Haoran,Han Zhengzhi. An improved sequential minimal optimization learning algorithm for regression support vector machine [J]. Journal of Software,2003,14(12):2006 – 2013 (in Chinese)

[6] 刘胜,李妍妍. 自适应 GA-SVM 参数选择算法研究[J]. 哈尔滨工程大学学报,2007,28(4):398 – 402
Liu Sheng,Li Yanyan. Parameter selection algorithm for support Vector machines based on adaptive genetic algorithm [J]. Journal of Harbin Engineering University,2007,28(4):398 – 402 (in Chinese)

[7] 闫国华,朱永生. 支持向量机回归的参数选择方法 [J]. 计算机工程,2009,35(14):218 – 220
Yan Guohua, Zhu Yongsheng. Parameters selection method for support vector machine regression [J]. Computer Engineering, 2009,35(14):218 – 220 (in Chinese)

[8] 董磊,任章,李清东. 基于 SMO-SVR 的飞机舵面损伤故障趋势预测 [J]. 北京航空航天大学学报,2012,38(10):1300 – 1305
Dong Lei,Ren Zhang,Li Qingdong. Fault prediction for aircraft control surface damage based on SMO-SVR [J]. Journal of Beijing University of Aeronautics and Astronautics,2012,38(10):1300 – 1305 (in Chinese)

[9] 王书舟,伞冶,张允昌. 基于支持向量机改进 SMO 算法的直升机旋翼自转着陆过程建模[J]. 航空学报,2009,30(1):46 – 51
Wang Shuzhou,San Ye,Zhang Yunchang. Modeling for landing process of helicopter with rotator self-rotating based on modified SMO algorithm of support vector machine[J]. Acta Aeronautica et Astronautica Sinica,2009,30(1):46 – 51(in Chinese)

[10] 王定成. 支持向量机建模预测与控制[M]. 北京:气象出版社,2009:48 – 49
Wang Dingcheng. Prediction and control based on support vector machine modelling [M]. Beijing: Meteorological Press, 2009:48 – 49(in Chinese)