

基于核参数分时段调节型 LSSVM 的 在线过程辨识方法

周欣然^{1,2}, 滕召胜¹, 易 钊¹

(1. 湖南大学电气与信息工程学院, 湖南 长沙 410082;
2. 中南大学信息科学与工程学院, 湖南 长沙 410075)

摘要: 利用最小二乘支持向量机(least square support vector machine, LSSVM)在线辨识时变非线性过程时, 设定其核参数较困难, 设定的核参数不能适应过程变化而进行自动调节。针对此问题, 提出了一种基于核参数分时段调节型 LSSVM 的在线过程辨识方法。该方法利用了三个 LSSVM, 并将整个建模预测时期分为启动阶段和若干个工作周期。初始阶段末和每个工作周期末选定预测误差和最小的 LSSVM, 作为后续工作周期的工作 LSSVM, 同时根据启发式规则为另两个 LSSVM 设定核参数, 它们作为后续工作周期的比较 LSSVM。该方法设定核参数相对容易, 而且核参数具有一定的自动调节能力。数字仿真显示, 从统计角度而言, 所提方法比传统方法有更好的适应性。

关键词: 在线过程辨识; 时变非线性过程; 最小二乘支持向量机; 核参数

中图分类号: TP 273

文献标志码: A

Online process identifying based on time-division regulated kernel parameter LSSVM

ZHOU Xin-ran^{1,2}, TENG Zhao-sheng¹, YI Zhao¹

(1. Coll. of Electrical and Information Engineering, Hunan Univ., Changsha 410082, China;
2. School of Information Science and Engineering, Central South Univ., Changsha 410075, China)

Abstract: It is difficult setting the kernel parameter which is applied to online identify a time-varying nonlinear process by using least square support vector machine(LSSVM), and such setted one can not automatically adjust to adapt it to the varying process. In view of this situation, an online process identification approach based on time-division regulated kernel parameter LSSVM is proposed. Three LSSVMs are utilized and the whole modeling predicting times are divided into a starting stage and several working periods. The LSSVM with a smallest sum of predicting errors is selected as a working LSSVM for successive working periods at the end of both the starting stage and each working period. Moreover, kernel parameters are reset for other two LSSVMs according to heuristic rules, and they are used as comparative LSSVMs for the following working periods. The method is easy to set kernel parameters and has adjustability to a certain extent. The numerical simulation shows the adaptability of the method is better than that of traditional methods statistically.

Keywords: online process identification; time-varying nonlinear process; least square support vector machine (LSSVM); kernel parameter

0 引言

最小二乘支持向量机^[1-2](least squares support vector machine, LSSVM)用等式约束代替支持向量机^[3]的不等式约束, 因此其求解问题就变成一个线性方程组求解问题, 这比支持向量机训练中受约束的二次规划问题求解运算量减

少许多。文献[4]利用分块矩阵求逆公式设计了 LSSVM 的增量式和在线式学习算法, 该算法计算量小, 能跟踪过程动态特性, 因而适合训练 LSSVM 对时变非线性过程在线建模。目前用 LSSVM 对动态过程建模时其超参数(核参数、惩罚因子)的设定没有理论依据, 因此往往凭主观设定。而 LSSVM 的建模效果与超参数的值有关, 但当前常见的

超参数选择方法^[5-7]运算量都很大,只适合离线近似寻优,不适合在线寻优。设计一种建模效果对超参数初始设定值依赖性不强的建模方法很有必要,随着时间的推移,使超参数脱离不良的设定值,能动态变化以便适应过程当前动态特征。另一方面,时变过程的变化特征(系统结构或参数变化)一般很难预测,可能在一个时段按一种规律变化,在另一个时段又按另一种规律变化,也可能在某些时段又不变化;要使 LSSVM 在各种时段都取得较好的预测效果,LSSVM 的超参数也该相应地调整。

要使 LSSVM 模型运行过程中所有超参数同时动态寻优比较困难。本文设计了一种核参数能适用过程变化的在线型 LSSVM 建模方法。该方法将整个建模预测时期划分为启动阶段和若干个工作周期,在每个工作周期中,利用一个 LSSVM 在线建模预测,且称该 LSSVM 为工作 LSSVM;利用另两个核参数不同的 LSSVM 在线建模,但其预测值仅作参考用,且称该两 LSSVM 为比较 LSSVM。在每个工作周期末选出本周期预测误差和最小的 LSSVM 作为下一个工作周期的工作 LSSVM;并按下文规则为另两个 LSSVM 设定新的核参数,成为下一个工作周期用于比较的 LSSVM。当核参数初始值设置不当或被辨识过程动态特性变化时,本文方法能搜索到比较适合过程当前动态特征的 LSSVM 核参数,以便改善建模效果。

1 LSSVM 回归及其学习算法简介

1.1 LSSVM 回归

对于一组输入样本序列 $\{x_i, y_i\}, i = 1, \dots, l; x_i \in \mathbf{R}^d; y_i \in \mathbf{R}$, LSSVM 利用非线性映射 $\Psi(\cdot): X \rightarrow F$, 将输入数据映射到一个高维特征空间,使输入空间中的非线性函数估计问题转化为高维特征空间中的线性函数估计问题,回归函数形式为

$$y(x) = \mathbf{w}^T \Psi(x) + b \quad (1)$$

根据结构风险最小化原则,并使样本拟合误差最小化,回归问题变为约束优化问题

$$\min J(\mathbf{w}, \mathbf{e}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{C}{2} \sum_{i=1}^l e_i^2,$$

$$\text{s. t. } y_i = \mathbf{w}^T \Psi(x_i) + b + e_i, i = 1, 2, \dots, l \quad (2)$$

式中, C 是惩罚因子。为求解上述优化问题,把约束优化问题变成无约束优化问题,建立 Lagrange 函数

$$L(\mathbf{w}, b, \mathbf{e}, \mathbf{a}) = J(\mathbf{w}, \mathbf{e}) - \sum_{i=1}^l a_i (\mathbf{w}^T \Psi(x_i) + b + e_i - y_i) \quad (3)$$

根据 KKT 条件有

$$\begin{cases} \frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^l a_i \Psi(x_i) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^l a_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \rightarrow a_i = C e_i \\ \frac{\partial L}{\partial a_i} = 0 \rightarrow \mathbf{w}^T \Psi(x_i) + b + e_i - y_i = 0 \end{cases} \quad (4)$$

消去方程组(4)中 e_i, \mathbf{w} 后得到线性方程组:

$$\begin{bmatrix} 0 & \mathbf{e}1^T \\ \mathbf{e}1 & \mathbf{Q} + C^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \quad (5)$$

式中, $\mathbf{y} = (y_1, y_2, \dots, y_l)^T; \mathbf{e}1 = (1, 1, \dots, 1)^T; \mathbf{a} = (a_1, a_2, \dots, a_l)^T; \mathbf{Q}$ 称为核矩阵; $Q_{ij} = (\Psi(x_i) \cdot \Psi(x_j)) = k(x_i, x_j) (i, j = 1, 2, \dots, N), k(x_i, x_j)$ 是满足 Mercer 条件的核函数,常用的有线性函数、多项式函数、径向基函数、多层感知函数等,本文取径向基函数,即

$$k(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2 / \beta^2) \quad (6)$$

式中, $\beta > 0$, 称为核参数。

从式(5)解出 b, \mathbf{a} , 回归函数(1)化为

$$y(x) = \sum_{i=1}^l a_i k(x_i, x) + b \quad (7)$$

1.2 LSSVM 回归学习算法

用于时变过程在线建模的 LSSVM 利用滑动时间窗中的过程输入输出构成 LSSVM 学习的样本,时间窗不宜太长,因为位于时间窗前面的老数据与过程当前动态特性可能矛盾,对建模没有参考价值,甚至有负面影响。因此,样本数不大,利用解上面方程组(5)的方法一次性(批量式)训练 LSSVM 即可,而不宜采用适合大样本集的 SMO^[8] 训练算法。

在每一个预测步或控制周期解一次方程组(5)计算量仍然较大,文献[4]设计了增量式和在线式学习算法,每增加一个新样本或抛弃一个老样本时,求解方程组(5)在前一次求解结果的基础上进行,从而减少计算时间。综合运用它们得出了在线型 LSSVM 学习算法。现将它们简述如下。

1.2.1 LSSVM 的增量式学习算法

当样本逐个添加时, LSSVM 适合采用增量式学习算法。对于样本集 $\{(x_i, y_i)\}_{i=1}^l$, 随着新样本的到来,样本编号 l 逐步增大。记

$$\mathbf{y}(l) = (y_1, y_2, \dots, y_l)^T, \mathbf{a}(l) = (a_1, a_2, \dots, a_l)^T$$

$$Q(l)_{ij} = (\Psi(x_i) \cdot \Psi(x_j)) = k(x_i, x_j), i, j = 1, 2, \dots, l$$

$$\mathbf{H}(l) = \mathbf{Q}(l) + C^{-1} \mathbf{I}, b(l) = b_l$$

显然,它们的维数或值随 l 变化,则方程(5)表示为

$$\begin{bmatrix} 0 & \mathbf{e}1^T \\ \mathbf{e}1 & \mathbf{H}(l) \end{bmatrix} \begin{bmatrix} b(l) \\ \mathbf{a}(l) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y}(l) \end{bmatrix} \quad (8)$$

方程(8)的解为

$$b(l) = \frac{\mathbf{e}1^T \mathbf{H}(l)^{-1} \mathbf{y}(l)}{\mathbf{e}1^T \mathbf{H}(l)^{-1} \mathbf{e}1} \quad (9a)$$

$$\mathbf{a}(l) = \mathbf{H}(l)^{-1} \left(\mathbf{y}(l) - \frac{\mathbf{e}1 \mathbf{e}1^T \mathbf{H}(l)^{-1} \mathbf{y}(l)}{\mathbf{e}1^T \mathbf{H}(l)^{-1} \mathbf{e}1} \right) =$$

$$\mathbf{H}(l)^{-1} (\mathbf{y}(l) - \mathbf{e}1 b(l)) \quad (9b)$$

得出的回归模型为

$$y(x, l) = \sum_{i=1}^l a_i(l) k(x_i, x) + b(l) \quad (10)$$

从式(9)可知,求 $b(l), \mathbf{a}(l)$ 的时间主要花在计算 $\mathbf{H}(l)^{-1}$ 。每当新样本 (x_{l+1}, y_{l+1}) 加入样本集时有

$$\mathbf{H}(l+1) = \mathbf{Q}(l+1) + C^{-1} \mathbf{I} = \begin{bmatrix} \mathbf{H}(l) & \mathbf{V}(l+1) \\ \mathbf{V}(l+1)^T & v(l+1) \end{bmatrix} \quad (11)$$

式中, $\mathbf{V}(l+1) = [k(\mathbf{x}_{l+1}, \mathbf{x}_1), \dots, k(\mathbf{x}_{l+1}, \mathbf{x}_l)]^T$; $v(l+1) = k(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) + 1/C$ 。利用分块矩阵求逆公式^[4,9]得出

$$\mathbf{H}(l+1)^{-1} = \begin{bmatrix} \mathbf{H}(l) & \mathbf{V}(l+1) \\ \mathbf{V}(l+1)^T & v(l+1) \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{H}(l)^{-1} + \mathbf{H}(l)^{-1}\mathbf{V}(l+1) \cdot \rho^{-1}\mathbf{V}(l+1)^T\mathbf{H}(l)^{-1} & -\mathbf{H}(l)^{-1}\mathbf{V}(l+1)\rho^{-1} \\ -\rho^{-1}\mathbf{V}(l+1)^T\mathbf{H}(l)^{-1} & \rho^{-1} \end{bmatrix} \quad (12)$$

其中 $\rho = v(l+1) - \mathbf{V}(l+1)^T\mathbf{H}(l)^{-1}\mathbf{V}(l+1)$ 。可见不必重新计算 $\mathbf{H}(l+1)^{-1}$, 而是利用 $\mathbf{H}(l)^{-1}$ 递推计算, 从而减少计算量。

1.2.2 LSSVM 的在线式学习算法

设滑动时间窗的长度为 L , 则利用增量式学习算法学习完这初始 L 个样本后, LSSVM 要进行在线式学习, 即 $l \geq L$ 后, 总是利用最新的 L 个样本 $\{(\mathbf{x}_i, y_i)\}_{i=l-L+1}^l$ 训练 LSSVM。此后记

$$\begin{aligned} \mathbf{y}(l) &= (y_{l-L+1}, y_{l-L+2}, \dots, y_l)^T \\ \mathbf{a}(l) &= (a_{l-L+1}, a_{l-L+2}, \dots, a_l)^T, \end{aligned}$$

$$Q(t)_{ij} = k(\mathbf{x}_{l-L+i}, \mathbf{x}_{l-L+j}), i, j = 1, 2, \dots, L$$

得出的回归模型为

$$y(\mathbf{x}, l) = \sum_{i=l-L+1}^l a_i(t)k(\mathbf{x}_i, \mathbf{x}) + b(l), l \geq L \quad (13)$$

随后要抛弃样本集中最老的样本 $(\mathbf{x}_{l-L+1}, y_{l-L+1})$ 。记

$$\bar{Q}(l)_{ij} = k(\mathbf{x}_{l-L+1+i}, \mathbf{x}_{l-L+1+j}), i, j = 1, 2, \dots, L-1$$

$$\bar{\mathbf{H}}(l) = \bar{Q}(l) + C^{-1}\mathbf{I}$$

则 $\mathbf{H}(l)$ 可写成分块形式

$$\mathbf{H}(l) = Q(l) + C^{-1}\mathbf{I} = \begin{bmatrix} \bar{v}(l-L+1) & \bar{\mathbf{V}}(l) \\ \bar{\mathbf{V}}(l)^T & \bar{\mathbf{H}}(l) \end{bmatrix} \quad (14)$$

式中, $\bar{\mathbf{V}}(l) = [k(\mathbf{x}_{l-L+1}, \mathbf{x}_{l-L+2}), \dots, k(\mathbf{x}_{l-L+1}, \mathbf{x}_l)]$; $\bar{v}(l-L+1) = k(\mathbf{x}_{l-L+1}, \mathbf{x}_{l-L+1}) + 1/C$ 。利用分块矩阵求逆公式^[4,9]得出

$$\mathbf{H}(l)^{-1} = \begin{bmatrix} \bar{v}(l-L+1) & \bar{\mathbf{V}}(l) \\ \bar{\mathbf{V}}(l)^T & \bar{\mathbf{H}}(l) \end{bmatrix}^{-1} = \begin{bmatrix} \bar{\rho}^{-1} & -\bar{\rho}^{-1}\bar{\mathbf{V}}(l)\bar{\mathbf{H}}(l)^{-1} \\ -\bar{\mathbf{H}}(l)^{-1}\bar{\mathbf{V}}(l)^T\bar{\rho}^{-1} & \bar{\mathbf{H}}(l)^{-1} + \bar{\mathbf{H}}(l)^{-1}\bar{\mathbf{V}}(l)^T\bar{\rho}^{-1}\bar{\mathbf{V}}(l)\bar{\mathbf{H}}(l)^{-1} \end{bmatrix} \quad (15)$$

其中, $\bar{\rho} = \bar{v}(l-L+1) - \bar{\mathbf{V}}(l)\bar{\mathbf{H}}(l)^{-1}\bar{\mathbf{V}}(l)^T$ 。另将已用式(12)递推求出的 $\mathbf{H}(l)^{-1}$ 写成分块形式

$$\mathbf{H}(l)^{-1} = \begin{bmatrix} \eta & \mathbf{P}_{1 \times (L-1)} \\ \mathbf{P}^T & \mathbf{D}_{(L-1) \times (L-1)} \end{bmatrix} \quad (16)$$

比照式(15)和式(16), 它们的对应分块必相等, 直接利用 $\mathbf{H}(l)^{-1}$ 的分块求出

$$\bar{\mathbf{H}}(l)^{-1} = \mathbf{D} - \mathbf{P}^T\eta^{-1}\mathbf{P} \quad (17)$$

每当新样本 $(\mathbf{x}_{l+1}, y_{l+1})$ 加入样本集时, 样本集为 $\{(\mathbf{x}_i, y_i)\}_{i=l-L+2}^{l+1}$, 且有

$$\mathbf{H}(l+1) = \begin{bmatrix} \bar{\mathbf{H}}(l) & \mathbf{V}(l+1) \\ \mathbf{V}(l+1)^T & v(l+1) \end{bmatrix} \quad (18)$$

此时, $\mathbf{V}(l+1) = [k(\mathbf{x}_{l+1}, \mathbf{x}_{l-L+2}), \dots, k(\mathbf{x}_{l+1}, \mathbf{x}_l)]^T$, 仍有 $v(l+1) = k(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) + 1/C$ 。利用式(12)计算 $\mathbf{H}(l+1)^{-1}$, 利用式(9)求系数得出模型(13)。

$\mathbf{H}(l)$ 和 $\bar{\mathbf{H}}(l)$ 对称, 因此 $\mathbf{H}(l)^{-1}$ 和 $\bar{\mathbf{H}}(l)^{-1}$ 也是对称矩阵, 只计算它们的对角线元素和上三角部分元素即可; 在建模和预测时还有一些重复计算也可去掉。可以采用文献[10]的加速策略来提高算法速度。

2 基于核参数自调节型 LSSVM 的在线过程辨识方法

设过程的离散形式为: $y(t) = f_i(y(t-1), \dots, y(t-n), u(t-ud), \dots, u(t-m))$, 其中 $m \leq n$, ud 为非负整数, 它们为过程结构参数, 可分别取估计值 m', n' 和 ud' 。记 $\mathbf{x}_t = (y(t-1), \dots, y(t-n), u(t-ud), \dots, u(t-m))$, $y_t = y(t)$, 则 (\mathbf{x}_t, y_t) 就是 LSSVM 的一个学习样本。 f_i 在过程的不同变化时段(时刻)可能有不同的表达式, 用 LSSVM 建模, 则 $\Psi(\cdot)$ 的形式也应该相应变化才能取得好的逼近效果, 最后核函数 $k(\mathbf{x}_i, \mathbf{x}_j)$ 的形式也变化, 为简便见, 本文取定核函数形式, 只让核参数自适应变化; 下文算法用 σ 代表式(6)中 β^2 , 直接调整 σ 。

本文方法采用的最小二乘支持向量机记为 LSSVM, ($i=1, 2, 3$), 它们的 C 值相同, σ 分别记为 σ_i 。为了说明 LSSVM 的学习和工作过程, 将时间步序列如图 1 划分。从第 $n+1$ 时刻起进入启动阶段, 启动阶段的前 WL 步称为学习阶段, 3 个 LSSVM 从得到第一个样本开始进行增量式学习, 第 $n+2$ 时刻过去后, 3 个 LSSVM 都学习完 2 个样本, 就可以用来预测第 $n+3$ 时刻的过程输出值, 即 $\hat{y}_{n+3,i}$, 重复地进行学习和预测, 直到第 $n+WL$ 时刻过去后, 3 个 LSSVM 都学习完 WL 个样本, 都预测下一时刻的输出值 $\hat{y}_{n+WL+1,i}$, 然后都消除一个老样本。接下来进入 CL 步的在线式学习阶段, 称为比较阶段; 第 k ($k=n+WL+1, \dots, n+WL+CL$) 步测得 y_k 后, 计算预测误差绝对值 $|y_k - \hat{y}_{k,i}|$, 并将它累加到预测误差累计变量 ES_i 中, 然后都进行一次增量式学习, 然后计算 $\hat{y}_{k+1,i}$, 再都消除一个老样本; 重复这些操作 CL 次。

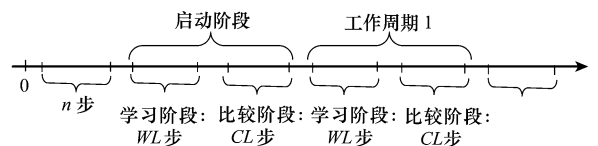


图 1 时间步序列划分图

启动阶段末对 $ES_i (i=1,2,3)$ 排序得: $ES_{i_1} \leq ES_{i_2} \leq ES_{i_3}$, LSSVM _{i_1} 将是工作周期 1 期间的工作 LSSVM, 它的 σ_{i_1} 不变; 清空 LSSVM _{i_2} 、LSSVM _{i_3} 的数据, 按如下启发式规则在区间 $[\sigma_{\min}, \sigma_{\max}]$ 中重新设定 σ_{i_2} 和 σ_{i_3} :

(1) 如果 $\sigma_{i_1} \leq \sigma_{i_2} \leq \sigma_{i_3}$, 说明用比 σ_{i_1} 更小的值作 σ 参数的 LSSVM 可能预测效果更好, 故在 $[\sigma_{\min}, \sigma_{i_1}]$ 试探搜索一参数; 但具体取值很难找到理论依据, 仿真发现取 $\sigma_{i_3} = (\sigma_{\min} + \sigma_{i_1})/2$ 可以得到满意的效果; 而让 σ_{i_2} 不变; 同理, 如果 $\sigma_{i_1} \leq \sigma_{i_3} \leq \sigma_{i_2}$, 则 $\sigma_{i_2} = (\sigma_{\min} + \sigma_{i_1})/2$, σ_{i_3} 不变;

(2) 如果 $\sigma_{i_3} \leq \sigma_{i_2} \leq \sigma_{i_1}$, 说明用比 σ_{i_1} 更大的值作 σ 参数的 LSSVM 可能预测效果更好, 故在 $[\sigma_{i_1}, \sigma_{\max}]$ 试探搜索一参数; 仿真发现取 $\sigma_{i_3} = \sigma_{i_1} + (\sigma_{\max} - \sigma_{i_1}) \times 3/4$ 可以得到满意的效果; 而让 σ_{i_2} 不变; 同理, 如果 $\sigma_{i_1} \leq \sigma_{i_3} \leq \sigma_{i_2}$, 则 $\sigma_{i_2} = \sigma_{i_1} + (\sigma_{\max} - \sigma_{i_1}) \times 3/4$, σ_{i_3} 不变;

(3) 如果 $\sigma_{i_2} \leq \sigma_{i_1} \leq \sigma_{i_3}$, 说明 σ 参数在 $[\sigma_{i_2}, \sigma_{i_1}]$ 或 $[\sigma_{i_1}, \sigma_{i_3}]$ 内取值的 LSSVM 可能预测效果更好; 仿真发现取 $\sigma_{i_2} = (\sigma'_{i_2} + \sigma_{i_1})/2$, $\sigma_{i_3} = (\sigma'_{i_3} + \sigma_{i_1})/2$ 可以得到满意的效果, σ'_{i_2} 、 σ'_{i_3} 分别表示 σ_{i_2} 、 σ_{i_3} 的原值; $\sigma_{i_3} \leq \sigma_{i_1} \leq \sigma_{i_2}$ 时类似处理。

在工作周期 1 的学习阶段, LSSVM _{i_1} 进行在线式学习, 并用它作预测模型; LSSVM _{i_2} 、LSSVM _{i_3} 进行增量式学习。在工作周期 1 的比较阶段, LSSVM _{i_1} 仍然进行在线式学习, 仍然用它计算过程输出预测值, 还要累计此阶段预测误差值; LSSVM _{i_2} 、LSSVM _{i_3} 也进行在线式学习, 计算它的预测值, 并累计预测值的误差; LSSVM _{i_2} 、LSSVM _{i_3} 仅供比较用, 故叫比较 LSSVM。

工作周期 1 的最后一步处理完毕, 又要对 $ES_i (i=1,2,3)$ 排序并选出具有最小预测误差累计值的最小二乘支持向量机, 其编号仍用 i_1 表示, 重新设置 σ_{i_2} 和 σ_{i_3} 。然后进入工作周期 2 的处理, 处理过程与前一工作周期相同。

工作周期无限重复, 每个工作周期之后要选出最好的 LSSVM 用于下一工作周期的建模, 并为另两个 LSSVM 设定一个可能较好的核参数值。

现将在线建模预测算法归纳如下:

步骤 1 设定有关参数: WL 、 CL 、 σ_{\min} 、 σ_{\max} 以及 C 、 $\sigma_i (i=1,2,3)$ 分别位于 $[\sigma_{\min}, \sigma_{\max}]$ 四等分点的附近, 各 LSSVM 对应的样本集大小 l 置 0。

步骤 2 执行启动阶段的操作

(1) LSSVM _{i} ($i=1,2,3$) 在学习阶段执行增量式学习算法: 若 LSSVM _{i} ($i=1,2,3$) 的样本数 $l=1$, 则 $\mathbf{H}(l)^{-1} = 1/(k$

$(\mathbf{x}_i, \mathbf{x}_i) + 1/C)$, 否则用式(12)计算 $\mathbf{H}(l)^{-1}$; 并且学完两个样本 ($l \geq 2$) 后就利用式(9)算出 b 、 a 建立模型(10), 且用它们来计算预测值, 并用三个预测值的均值作为过程输出预测值。学习阶段末所有 LSSVM 都利用式(17)消除一个老样本;

(2) LSSVM _{i} ($i=1,2,3$) 在比较阶段的每步依次执行: 测量获取过程实际输出值; 计算并累计各 LSSVM _{i} 的前次预测误差绝对值存于 ES_i ; 利用式(12)和式(9)建立模型(13)且用它们来计算预测值, 用三个 LSSVM 预测值的均值作为过程输出预测值; 利用式(17)消除一个老样本。

步骤 3 在启动阶段末或每个工作周期末选出 ES_i 最小的 LSSVM 作为后续工作周期的工作 LSSVM; 清空另两个 LSSVM 的数据, 即将它们的样本集大小 l 重新置 0; 按启发式规则重新设定它们的参数 σ , 用其作为后续工作周期的比较 LSSVM。

步骤 4 执行工作周期中的操作

(1) 在学习阶段, 工作 LSSVM 执行在线式学习算法, 并且用建立的模型(13)作预测模型; 两比较 LSSVM 执行增量式学习算法, 学习阶段末它们都利用式(17)消除一个老样本;

(2) 在比较阶段, 工作 LSSVM 仍然执行在线式学习算法, 并且用作预测模型; 比较 LSSVM 也执行在线式学习算法, 并计算预测值; 分别累计三个 LSSVM _{i} 在此阶段中的预测误差 ES_i 。

步骤 5 转向步骤 3。

3 仿真实验及分析

为检测本文算法的效果, 下面利用它对三个过程模型在线建模预测。设定算法有关参数: $WL = 50$, $CL = 10$, $\sigma_{\min} = 0.5$, $\sigma_{\max} = 20$, $\sigma_1 = 3$, $\sigma_2 = 9$, $\sigma_3 = 18$, $C = 500$, $n' = 3$, $m' = 2$, $ud' = 1$, 仿真步数 $SM = 654$ 。

仿真过程模型 M_1 在仿真时段内是恒定过程, 其离散形式如下

$$y(t) = \frac{y(t-1)y(t-2)u(t-1)[y(t-3)-1] + u(t-2)}{1 + y^2(t-1) + y^2(t-2)}$$

仿真模型 M_2 是持续时变过程, 可表示为如下形式

$$y(t) = \frac{y(t-1)y(t-2)u(t-1)[y(t-3)-1] + u(t-2)}{1 + y^2(t-1) + y^2(t-2)} +$$

$$2\sin(2\pi t/50)$$

仿真模型 M_3 是分段时变过程, 可表示为如下形式

$$y(t) = \begin{cases} \frac{y(t-1)y(t-2)u(t-1)[y(t-3)-1] + u(t-2)}{1 + y^2(t-1) + y^2(t-2)}, & t \leq 150 \\ \frac{y(t-1)y(t-2)u(t-1)[y(t-3)-1] + u(t-2)}{1 + y^2(t-1) + y^2(t-2)} + \sin(2\pi(t-150)/50) + (t-150)/300, & 150 < t \leq 450 \\ \frac{y(t-1)y(t-2)u(t-1)[y(t-3)-1] + u(t-2)}{1 + y^2(t-1) + y^2(t-2)} + 1, & t < 450 \end{cases}$$

仿真过程的输入信号 $u_1(t)$ 在仿真时段内是周期为 50 的方波,其第一个周期如下

$$u(t) = \begin{cases} 2, & 0 \leq t \leq 25 \\ 0, & 25 < t < 50 \end{cases}$$

输入信号 $u_2(t)$ 是叠加三角函数波,形式为

$$u(t) = \sin(2\pi t/25) + \sin(2\pi t/50) + \sin(2\pi t/100)$$

输入信号 $u_3(t)$ 是由方波、三角函数波、锯齿波构成的分段信号,形式为

$$u(t) = \begin{cases} 2, & 50k \leq t \leq 50k + 25, k = (0, 1, 2, 3, 4), t \leq 200 \\ 0, & 50k + 25 < t < 50k + 50, k = (0, 1, 2, 3), t \leq 200 \\ \sin(2\pi t/25) + \sin(2\pi t/50) + \sin(2\pi t/100), & 200 < t \leq 400 \\ 2(t - 50k)/25, & 50k \leq t \leq 50k + 25, k = (8, \dots, 13), 400 < t \leq 654 \\ 0, & 50k + 25 < t < 50k + 50, k = (8, \dots, 12), 400 < t \leq 654 \end{cases}$$

用 LSSVM 建模的传统方法中,其核参数凭主观设定后,可用模型每步的预测误差之和 $pe = \sum_{t=1}^{SM} |y(t) - \hat{y}(t)|$ 衡量模型效果,我们不能凭某一次预测误差和来衡量这种方法的优劣,而要用其统计特性度量。本文将 σ 取 $[\sigma_{\min}, \sigma_{\max}]$ 中的一系列值 $(0.5, 1, \dots, 19, 20)$ 得出一系列 LSS-

VM,分别计算它们的 $pe_j (j=1, 2, \dots, 21)$,然后计算它们预测误差和的均值,并用它度量传统方法的性能。

现用传统方法(多次)和本文方法对上面三种模型(取 $y(1) = y(2) = y(3) = 0$)在三种输入信号作用下进行在线辨识和单步预测,将传统方法 pe_j 的统计数据与本文方法的建模预测误差和如表 1 所示。

表 1 传统方法预测误差和统计数据与本文方法的预测误差和

	u_1		u_2		u_3	
	传统方法 pe_j	本文方法 pe	传统方法 pe_j	本文方法 pe	传统方法 pe_j	本文方法 pe
M_1	最小值	9.497 0	最小值	23.274 8	最小值	16.870 3
	最小值	17.931 0	最小值	55.360 1	最小值	29.330 8
	均值	13.815 0	均值	30.528 2	均值	20.900 4
M_2	最小值	50.802 0	最小值	82.509 2	最小值	78.786 9
	最小值	86.302 7	最小值	355.192 6	最小值	192.864 1
	均值	62.279 6	均值	110.228 8	均值	90.436 0
M_3	最小值	101.057 5	最小值	44.546 5	最小值	49.451 8
	最小值	277.633 0	最小值	184.931 1	最小值	98.989 6
	均值	127.290 2	均值	57.739	均值	57.677 2

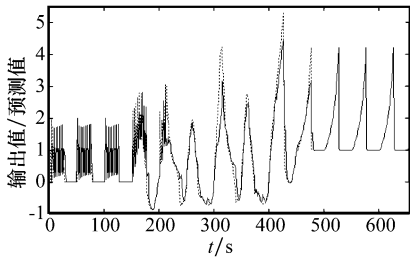
从表 1 的九种情况可看出,从统计角度而言,本文方法建模的预测误差比传统方法的均值要小,可能比传统方法的最优值还好(如 u_1 作用于 M_2);本文方法预测结果可以避免出现传统方法的最坏情况。

下面进一步单独考察 u_3 作用于 M_3 的情况。图 2(a)和图 2(b)分别是传统方法 $\sigma=0.5, \sigma=4$ 的预测曲线与过程输出曲线图,两预测曲线对应的预测误差和分别是 21 次试验中的最大者 98.986 8 和最小者 49.451 8;图 2(c)则是本文方法预测曲线与过程输出曲线图,预测曲线对应的预测误差和是 50.553 1。容易看出,图 2(c)的预测曲线与 2(b)比较相似,接近真实输出;图 2(c)在拐点处预测效果比图 2(a)要好。

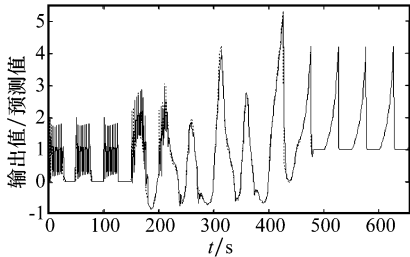
下面是 u_3 作用于 M_3 的情况时,本文方法调整参数 σ 的过程:在启动阶段和各工作周期中元组 $(\sigma_1, \sigma_2, \sigma_3)$ 值依次为 $(3, 9, 18)$ 、 $(3.000 0, 9.000 0, 1.750 0)$ 、 $(3.000 0, 1.125 0, 1.750 0)$ 、 $(3.000 0, 15.750 0, 1.750 0)$ 、 $(3.000 0, 9.375 0, 2.375 0)$ 、 $(3.000 0, 9.375 0, 17.343 8)$ 、 $(19.335 9,$

$9.375 0, 17.343 8)$ 、 $(19.335 9, 19.834 0, 17.343 8)$ 、 $(19.335 9, 19.834 0, 19.958 5)$ 、 $(19.335 9, 19.834 0, 9.918 0)$ 、 $(19.335 9, 5.209 0, 9.918 0)$,元组中用下划线标出的表示当前时段工作 LS_SVM 的 σ 值。

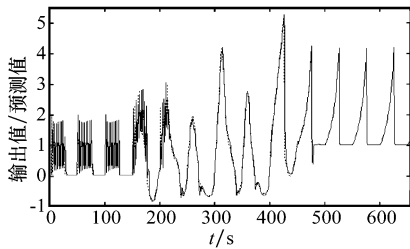
传统方法的不足有:LSSVM 的较佳参数 $\sigma(=\beta)$ 无法预知,凭主观设定;过程工作后,其传递函数和输入可能变化,而核参数不能相应变化;即使利用初始阶段样本离线求出近似最优的参数值,但随着过程动态特性的变化,这些参数值会变得不是近似最优的。本文方法的优点是,只要设定 LSSVM 的 β 的寻优区间 $[\sigma_{\min}, \sigma_{\max}]$;初始阶段结束或每个工作周期结束,都能对工作 LSSVM 的参数在上面区间内进行调整,尽管很难找到每一时段的最佳值,但有一定的自适应调节能力。与单个 LSSVM 建模预测方法^[4]相比,这种方法所需的存储空间增加两倍;引入文献[10]的加速策略后,这种方法计算量增加不到一倍。本文方法的一个不足是一次核参数调节要历经一个工作周期,可见调节有滞后性;如果系统的动态特性变化太快,本文方法就不能及时调整。



(a) 传统方法 $\sigma=0.5$ 的预测曲线与过程输出曲线



(b) 传统方法 $\sigma=4$ 的预测曲线与过程输出曲线



(c) 本文方法预测曲线与过程输出曲线

——: 预测值; - - - - -: 输出值

图2 模型预测曲线与过程输出曲线

4 结束语

针对利用最小二乘支持向量机在线辨识时变非线性过程时,设定其核参数的困难性和设定的核参数缺乏自适应性,本文充分利用 LSSVM 的增量式和在线式学习算法,设计了一种适合时变非线性过程的辨识方法。在不同的时

段,LSSVM 的参数 σ 根据设定的启发式规则在设定区间自适应地调整,在线训练出较好的模型,从而取得较好的预测效果。

如何在线调整 σ 的寻优区间使它能包含或靠近后续时段最佳核参数值是值得做的后续工作;另外,LSSVM 的惩罚因子 C 与 σ 的含义、作用不同,但对建模效果也会有影响,设计一种让 C 与 σ 都能在线自适应变化的建模方法也是值得做的后续工作。

参考文献:

- [1] Suykens J A K. Least squares support vector machine classifiers [J]. *Neural Process Letter*, 1999, 9(3):293 - 299.
- [2] Suykens J A K. Nonlinear modeling and support vector machines[C]// *IEEE Instrumentation and Measurement Technology Conference*, 2001:287 - 294.
- [3] Vapnik V. *Statistical learning theory*[M]. New York:Wiley, 1998.
- [4] 张浩然,汪晓东. 回归最小二乘支持向量机的增量和在线式学习算法[J]. *计算机学报*, 2006, 29(3):400 - 406.
- [5] 董春曦,饶鲜,杨绍全,等. 支持向量机参数选择方法研究[J]. *系统工程与电子技术*, 2004, 26(8):1117 - 1120. (Dong Chunxi, Rao Xian, Yang Shaoquan, et al. Method for selecting the parameters of support vector machines[J]. *Systems Engineering and Electronics*, 2004, 26(8):1117 - 1120.)
- [6] Guo X C, Yang J H, Wu C G, et al. A novel LS-SVMs hyperparameter selection based on particle swarm optimization [J]. *Neurocomputing*, 2008, 71(16):3211 - 3215.
- [7] 刘昌平,范明钰,王光卫,等. 基于梯度算法的支持向量机参数优化方法[J]. *控制与决策*, 2008, 23(11):1291 - 1295,1300.
- [8] Keerthi S S, Shevade S K. SMO algorithm for least squares SVM formulations[J]. *Neural Computation*, 2003, 15(2):487 - 507.
- [9] 毛汉清. 可逆矩阵的分块求逆方法研究[J]. *上海铁道学院学报*, 1994, 15(3):110 - 117.
- [10] 周欣然,滕召胜,赵新闻. 基于 LSSVM 的 MIMO 系统快速在线辨识方法[J]. *计算机应用*, 2009, 29(8):2281 - 2284,2314.