文章编号:1001 - 9081(2009)08 - 2281 - 04

基于 LSSVM 的 MIMO 系统快速在线辨识方法

周欣然[1,2],滕召胜[1],赵新闻[3]
(1. 湖南大学 电气与信息工程学院,长沙 410082; 2. 中南大学 信息科学与工程学院,长沙 410075;
3. 中南大学 物理科学与技术学院,长沙 410075)
(zhou_xr@ mail. csu. edu. cn)

摘 要:针对时变非线性多输入多输出(MIMO)系统在线辨识较困难的问题,提出一种基于最小二乘支持向量机(LSSVM)的快速在线辨识方法。介绍了现有 LSSVM 增量式和在线式学习算法,并为它引入了一些加速实现策略,得到 LSSVM 快速在线式学习算法。将 MIMO 系统分解为多个多输入单输出(MISO)子系统,对每一个 MISO 利用一个 LSSVM 在线建模;这些 LSSVM 执行快速在线式学习算法。数字仿真显示该方法建模速度快,模型预测精度高。

关键词:在线系统辨识;时变非线性系统;多输入多输出系统;最小二乘支持向量机
中图分类号: TP273;TP18 **文献标志码**:A

Fast online system identification for MIMO using LSSVM

ZHOU Xin-ran[1,2], TENG Zhao-sheng[1], ZHAO Xin-wen[3]
(1. *College of Electrical and Information Engineering, Hunan University, Changsha Hunan* 410082, *China*;
2. *School of Information Science and Engineering, Central South University, Changsha Hunan* 410075, *China*;
3. *School of Physical Science and Technology, Central South University, Changsha Hunan* 410083, *China*)

Abstract: To tackle the difficulty in identifying time-varying nonlinear Multi-Input Multi-Output (MIMO) system online, a fast online system identification approach based on Least Squares Support Vector Machine (LSSVM) was proposed. The existing LSSVM incremental and online learning algorithms were introduced, and some speeding up implementing tactics were designed and adopted in the algorithm; consequently, a fast online LSSVM learning algorithm was obtained. MIMO system was decomposed into multiple Multi-Input Single-Output (MISO) subsystems, and each MISO was modeled online via a LSSVM. The numerical simulation shows the modeling method is faster and the obtained models provide accurate prediction.

Key words: online system identification; time-varying nonlinear system; Multi-Input Multi-Output (MIMO) system; Least Squares Support Vector Machine (LSSVM)

0 引言

对多输入多输出(Multi-Input Multi-Output, MIMO)动态系统建模往往是设计控制系统的前提,利用被控对象工作机制来建模一般较困难,因此常用所测的对象输入输出数据来建模。近年来对 MIMO 系统建模研究较活跃,如文献[1]提出了基于系统频率响应的建模方法,文献[2-3]提出了基于神经网络的建模方法,文献[4]提出了基于模糊模型的建模方法,文献[5]提出了基于 ε-SVR 的建模方法。这些方法对时不变 MIMO 系统能取得较好的建模效果,若系统是时变的,则所得的模型不能及时反应系统的动态特性,因此,它们都只能用于离线建模。

最小二乘支持向量机[6-7]用等式约束代替支持向量机[8]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[9]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,因而适合训练 LSSVM 对时变非线性 MIMO 系统在线建模。

本文先介绍 LSSVM 回归及其现有的增量式和在线式学习算法,然后分析该算法的特征,引入一些快速实现策略以减少运算量,得出快速在线型 LSSVM 学习算法。再将 MIMO 分解为多个 MISO 子系统,对每一个 MISO 利用一个 LSSVM 在线建模,各 LSSVM 执行快速在线型学习算法得出即时 LSSVM 模型用于实时预测。

1 LSSVM 回归及其学习算法

1.1 LSSVM 回归

对于一组输入样本序列 $\{\boldsymbol{x}_i, y_i\}$, $i = 1,\cdots,N$, $\boldsymbol{x}_i \in \mathbf{R}^d$, $y \in \mathbf{R}$, LSSVM 利用非线性映射 $\psi(\cdot): X \to F$,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:

$$y(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\psi(\boldsymbol{x}) + b \tag{1}$$

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:

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

$$\text{s.t. } y_i = \boldsymbol{w}^{\mathrm{T}}\psi(\boldsymbol{x}_i) + b + e_i;\ i = 1,2,\cdots,N \tag{2}$$

收稿日期:2009 - 02 - 13;**修回日期**:2009 - 03 - 31。
基金项目:国家自然科学基金资助项目(60872128);国家技术创新基金资助项目(07C26214301740)。
作者简介:周欣然(1976 -),男,湖南衡山人,讲师,博士研究生,主要研究方向:智能检测、智能信息处理; 滕召胜(1963 -),男,湖南辰溪人,教授,博士生导师,主要研究方向:智能检测、测控系统、自动化装置、多传感器数据融合; 赵新闻(1968 -),男,湖南衡南人,副教授,主要研究方向:功能材料。

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：

$$L(\boldsymbol{w},b,\boldsymbol{e},\boldsymbol{a}) = J(\boldsymbol{w},\boldsymbol{e}) - \sum_{i=1}^{N} a_i(\boldsymbol{w}\psi(\boldsymbol{x}_i) + b + e_i - y_i) \tag{3}$$

根据 KKT 条件有：

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_{i=1}^{N} a_i\psi(\boldsymbol{x}_i) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} a_i = 0 \\ \dfrac{\partial L}{\partial e_i} = 0 \rightarrow a_i = Ce_i \\ \dfrac{\partial L}{\partial a_i} = 0 \rightarrow \boldsymbol{w}\psi(\boldsymbol{x}_i) + b + e_i - y_i = 0 \end{cases} \tag{4}$$

消去方程组(4)中 e_i，$\boldsymbol{w}$ 后得到线性方程组：

$$\begin{bmatrix} 0 & \boldsymbol{e}1^{\mathrm{T}} \\ \boldsymbol{e}1 & \boldsymbol{Q} + \boldsymbol{C}^{-1}\boldsymbol{I} \end{bmatrix}\begin{bmatrix} b \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix} \tag{5}$$

其中，$\boldsymbol{y} = (y_1,y_2,\cdots,y_N)^{\mathrm{T}}$，$\boldsymbol{e}1 = (1,1,\cdots,1)^{\mathrm{T}}$，$\boldsymbol{a} = (a_1,a_2,\cdots,a_N)^{\mathrm{T}}$，$Q_{ij} = (\psi(\boldsymbol{x}_i)\cdot\psi(\boldsymbol{x}_j)) = k(\boldsymbol{x}_i,\boldsymbol{x}_j)$，$i,j = 1,2,\cdots,N$。$k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 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：

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{\beta^2}\right) \tag{6}$$

$\beta > 0$，称为核参数；$\boldsymbol{Q}$ 称为核矩阵。

从式(5)解出 b、$\boldsymbol{a}$，回归函数(1)化为：

$$y(\boldsymbol{x}) = \sum_{i=1}^{N} a_i k(\boldsymbol{x}_i,\boldsymbol{x}) + b \tag{7}$$

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，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[10]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[9]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；本文进一步去掉该算法中不必要的和重复的计算，得出快速在线型 LSSVM 学习算法

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 $\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{i=l}$，随着新样本的到来，样本编号 l 逐步增大。记：

$\boldsymbol{y}(l) = (y_1,y_2,\cdots,y_l)^{\mathrm{T}}$，$\boldsymbol{a}(l) = (a_1,a_2,\cdots,a_l)^{\mathrm{T}}$

$Q(l)_{ij} = (\psi(\boldsymbol{x}_i)\cdot\psi(\boldsymbol{x}_j)) = k(\boldsymbol{x}_i,\boldsymbol{x}_j)$；$i,j = 1,2,\cdots,l$

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

显然，它们的维数或值随 l 变化；则方程(5)表示为：

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

方程(8)的解为：

$$b(l) = \frac{\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{H}(l)^{-1}\boldsymbol{y}(l)}{\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{H}(l)^{-1}\boldsymbol{e}1} \tag{9a}$$

$$\boldsymbol{a}(l) = \boldsymbol{H}(l)^{-1}\left(\boldsymbol{y}(l) - \frac{\boldsymbol{e}1\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{H}(l)^{-1}\boldsymbol{y}(l)}{\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{H}(l)^{-1}\boldsymbol{e}1}\right) = \boldsymbol{H}(l)^{-1}(\boldsymbol{y}(l) - \boldsymbol{e}1b(l)) \tag{9b}$$

得出的回归模型为：

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

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

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

其中，$\boldsymbol{V}(l+1) = [k(\boldsymbol{x}_{l+1},\boldsymbol{x}_1),\cdots,k(\boldsymbol{x}_{l+1},\boldsymbol{x}_l)]^{\mathrm{T}}$，$v(l+1) = k(\boldsymbol{x}_{l+1},\boldsymbol{x}_{l+1}) + 1/C$。利用分块矩阵求逆公式[9,11]得出：

$$\boldsymbol{H}(l+1)^{-1} = \begin{bmatrix} \boldsymbol{H}(l) & \boldsymbol{V}(l+1) \\ \boldsymbol{V}(l+1)^{\mathrm{T}} & v(l+1) \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{H}(l)^{-1} + \boldsymbol{H}(l)^{-1}\boldsymbol{V}(l+1)\rho^{-1}\boldsymbol{V}(l+1)^{\mathrm{T}}\boldsymbol{H}(l)^{-1} & -\boldsymbol{H}(l)^{-1}\boldsymbol{V}(l+1)\rho^{-1} \\ -\rho^{-1}\boldsymbol{V}(l+1)^{\mathrm{T}}\boldsymbol{H}(l)^{-1} & \rho^{-1} \end{bmatrix} \tag{12}$$

其中 $\rho = v(l+1) - \boldsymbol{V}(l+1)^{\mathrm{T}}\boldsymbol{H}(l)^{-1}\boldsymbol{V}(l+1)$。可见不必重新计算 $\boldsymbol{H}(l+1)^{-1}$，而是利用 $\boldsymbol{H}(l)^{-1}$ 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初始 L 个样本后，LSSVM 要进行在线式学习，即 $l \geqslant L$ 后，总是利用最新的 L 个样本 $\{(x_i,y_i)\}_{i=l-L+1}^{i=l}$ 训练 LSSVM。此后记：

$\boldsymbol{y}(t) = (y_{l-L+1},y_{l-L+2},\cdots,y_l)^{\mathrm{T}}$，$\boldsymbol{a}(t) = (a_{l-L+1},a_{l-L+2},\cdots,a_l)^{\mathrm{T}}$，$Q(t)_{ij} = k(\boldsymbol{x}_{l-L+i},\boldsymbol{x}_{l-L+j})$，$i,j = 1,2,\cdots,L$，得出的回归模型为：

$$y(\boldsymbol{x},l) = \sum_{i=l-L+1}^{l} a_i(t)k(\boldsymbol{x}_i,\boldsymbol{x}) + b(l);\ l \geqslant L \tag{13}$$

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

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

$\bar{\boldsymbol{H}}(l) = \bar{\boldsymbol{Q}}(l) + \boldsymbol{C}^{-1}\boldsymbol{I}$，则 $\boldsymbol{H}(l)$ 可写成另一种分块形式：

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

其中，$\bar{\boldsymbol{V}}(l) = [k(\boldsymbol{x}_{l-L+1},\boldsymbol{x}_{l-L+2}),\cdots,k(\boldsymbol{x}_{l-L+1},x_l)]$，$\bar{v}(l-L+1) = k(\boldsymbol{x}_{l-L+1},\boldsymbol{x}_{l-L+1}) + 1/C$，利用分块矩阵求逆公式[9,11]得出：

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

其中 $\bar{\rho} = \bar{v}(l-L+1) - \bar{\boldsymbol{V}}(l)\bar{\boldsymbol{H}}(l)^{-1}\bar{\boldsymbol{V}}(l)^{\mathrm{T}}$。又将已用式(12)递推求出的 $\boldsymbol{H}(l)^{-1}$ 写成另一种分块形式：

$$\bar{\boldsymbol{H}}(l)^{-1} = \begin{bmatrix} \eta & \boldsymbol{P}_{1\times(L-1)} \\ \boldsymbol{P}^{\mathrm{T}} & \boldsymbol{D}_{(L-1)\times(L-1)} \end{bmatrix}$$

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

$$\bar{\boldsymbol{H}}(l)^{-1} = \boldsymbol{D} - \boldsymbol{P}^{\mathrm{T}}\eta^{-1}\boldsymbol{P} \tag{17}$$

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

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

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

1.2.3　LSSVM 学习算法的加速实现策略

当编程实现上面学习算法时不宜直接套用公式,为了避免不必要的和重复的计算,本文对上面学习算法引入如下加速执行策略。

1) 因 $\boldsymbol{H}(l)$ 是对称矩阵,故 $\boldsymbol{H}(l)^{-1}$ 也为对称矩阵,故每一步只要计算和存储它的上(或下)三角部分即可。对于式(12)而言,记 $\boldsymbol{G}(l+1)=\boldsymbol{H}(l)^{-1}\boldsymbol{V}(l+1)$,是 $\min(l,L-1)$ 维的列向量,则:

$$\boldsymbol{G}(l+1)_i=\sum_{j=1}^{i}\boldsymbol{H}(l)_{ji}^{-1}\boldsymbol{V}(l+1)_j+\sum_{j=i+1}^{l}\boldsymbol{H}(l)_{ij}^{-1}\boldsymbol{V}(l+1)_j \tag{19}$$

$$\rho=v(l+1)-\boldsymbol{V}(l+1)^{\mathrm{T}}\boldsymbol{G}(l+1) \tag{20}$$

$$\boldsymbol{H}(l+1)_{i,end}^{-1}=-\boldsymbol{G}(l+1)_i\rho^{-1};$$

$$1\leqslant i\leqslant\min(l,L-1),\ end=\min(l+1,L) \tag{21}$$

$\boldsymbol{H}(l+1)^{-1}$ 的前 $\min(l,L-1)$ 列的上三角部分元素:

$$\boldsymbol{H}(l+1)_{ij}^{-1}=\boldsymbol{H}(l)_{ij}^{-1}-\boldsymbol{G}(l+1)_i\boldsymbol{H}(l+1)_{j,l+1}^{-1};$$

$$1\leqslant i\leqslant j\leqslant\min(l,L-1) \tag{22}$$

2) $\bar{\boldsymbol{H}}(l)$,$\bar{\boldsymbol{H}}(l)^{-1}$ 也是对称矩阵,在 $\boldsymbol{H}(l)^{-1}$ 已发挥作用(构成了即时 LSSVM 并预测)后,就可用于计算 $\bar{\boldsymbol{H}}(l)^{-1}$。计算式(17)时,为节省存储空间,$\bar{\boldsymbol{H}}(l)^{-1}$ 与 $\boldsymbol{H}(l)^{-1}$ 存于同一二维数组中,为避免过多的赋值运算,将式(16)中 $\boldsymbol{H}(l)^{-1}$ 的第一行 $\boldsymbol{H}(l)_{1,2}^{-1}$ 后所有元素缓存于向量 $\boldsymbol{HA}(l)$,为避免重复的乘法运算,还将 $\eta^{-1}\boldsymbol{HA}(l)$ 缓存于 $\boldsymbol{HB}(l)$,则 $\bar{\boldsymbol{H}}(l)^{-1}$ 上三角部分元素:

$$\bar{\boldsymbol{H}}(l)_{ij}^{-1}=\boldsymbol{H}(l)_{i+1,j+1}^{-1}-\boldsymbol{HA}(l)_i\boldsymbol{HB}(l)_j;$$

$$1\leqslant i\leqslant\min(l-1,L-1),\ i\leqslant j\leqslant\min(l-1,L-1) \tag{23}$$

3) 当利用 $\boldsymbol{x}_{l+1}$ 根据式(10)预测 y_{l+1} 时有:$\hat{y}_{l+1}=y(\boldsymbol{x}_{l+1},l)=\sum_{i=1}^{l}a_i(l)k(\boldsymbol{x}_i,\boldsymbol{x}_{l+1})+b(l)=\boldsymbol{a}(l)\boldsymbol{V}(l+1)+b(l)$。可见此时已经为下一时间步算出了式(11)中 $\boldsymbol{V}(l+1)$。类似地,式(18)中 $\boldsymbol{V}(l+1)$ 在上一步利用式(13)进行预测时也已算出,只要将后 $L-1$ 个数据都向前挪动一位即可。可见预测时算出的 $\boldsymbol{V}(l+1)$ 可用于下一时间步的增量学习。

4) 对于式(9)而言,因 $\boldsymbol{H}(l)^{-1}$ 只有上三角部分元素有效,出现 $\boldsymbol{H}(l)^{-1}$ 与列(行)向量右(左)乘时类似(19)方法计算;因 $\boldsymbol{e}1$、$\boldsymbol{e}1^{\mathrm{T}}$ 的元素全为 1,它们与向量相乘则直接将向量元素求和即可;式(9a)中分子分母相同部分 $\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{H}(l)^{-1}$ 计算一次缓存后可供第二次使用。

引入上面加速策略后,算法执行速度将明显提高(见后仿真实验),额外付出的空间代价为少数几个缓存向量。

2　基于 *LSSVM* 的 *MIMO* 系统快速在线辨识方法

对于可辨识的 n 个输入 k 个输出的 MIMO 系统,其 t 时刻第 j 个输出的离散形式为:

$$y_j(t)=f_{jt}(y'_{1j}(t),\cdots,y'_{rj}(t),\cdots,y'_{kj}(t),\boldsymbol{u}'_{1j}(t),\cdots,\boldsymbol{u}'_{ij}(t),\cdots,\boldsymbol{u}'_{nj}(t)) \tag{24}$$

其中:$y'_{rj}(t)=(y_r(t-yd_{rj}),y_r(t-yd_{rj}-1),\cdots,y_r(t-yo_{rj}))$,$\boldsymbol{u}'_{ij}(t)=(u_i(t-ud_{ij}),u_i(t-ud_{ij}-1),\cdots,u_i(t-uo_{ij}))$。$yd_{rj}$、$ud_{ij}$ 为纯延迟步数,yo_{rj}、uo_{ij} 为反应阶数,它们是系统结构参数;显然,$yd_{rj}\leqslant yo_{rj}$,否则 y_r 对 y_j 不起作用;$ud_{ij}\leqslant uo_{ij}$,否则 u_i 对 y_j 不起作用。

将上面 MIMO 系统分解为 k 个 MISO 子系统,则式(24)就表示第 j 个 MISO 的动态特征。每一个 MISO 用一个 LSSVM 辨识,则用 k 个 $LSSVM_j(j=1,\cdots,k)$ 就可将 k 个 MISO 系统辨识出来。对于式(24)而言,记:$\boldsymbol{x}_{jt}=(y'_{1j}(t),\cdots,y'_{rj}(t),\cdots,y'_{kj}(t),\boldsymbol{u}'_{1j}(t),\cdots,\boldsymbol{u}'_{ij}(t),\cdots,\boldsymbol{u}'_{nj}(t))$,$y_{jt}=y_j(t)$,则 $(\boldsymbol{x}_{jt},y_{jt})$ 就是 $LSSVM_j$ 的一个学习样本。各 $LSSVM_j$ 的超参数值 β_j^2、C_j、滑动时间窗长度 $SWLY_j$ 可以不同,但所测的 MIMO 输入输出向量 $\boldsymbol{u}(t)=(u_1(t),\cdots,u_i(t),\cdots u_n(t))$,$\boldsymbol{y}(t)=(y_1(t),\cdots,u_r(t),\cdots u_k(t))$ 分别存于各自的缓冲队列中。系统结构参数 yd_{rj}、ud_{ij}、yo_{rj}、uo_{ij} 的估计值分别用 ydl_{rj}、udl_{ij}、yol_{rj}、uol_{ij} 表示。记:$MYUL=\max(\max(yol_{rj}),\max(uol_{ij}))$,$MSWL=\max(SWLY_j)$,$MINUD=\min(udl_{ij})$,$r,j=1,\cdots,k$,$i=1,\cdots,n$,$y(t)$ 队列长度为 $MYUL+MSWL$,$u(t)$ 队列长度为 $MYUL+MSWL-MINUD+1$。

现将基于 LSSVM 的 MIMO 系统在线建模预测算法归纳如下:

1) 设定各 $LSSVM_j$ 有关参数:β_j^2、C_j,滑动时间窗长度 $SWLY_j$,模型阶次参数 ydl_{rj}、yol_{rj}、udl_{ij}、uol_{ij}。

用于 $LSSVM_j$ 建模的样本个数 lb_j 置 0(初始阶段 lb_j 小于 $SWLY_j$,后来等于 $SWLY_j$)。

2) 测取 MIMO 系统 $MYUL$ 个时刻输入输出向量 $\boldsymbol{u}(\cdot)$、$\boldsymbol{y}(\cdot)$ 存于缓冲队列。

3) 各LSSVM进入在线建模预测阶段,每一步它们执行增量式或在线式学习算法得出模型并用来预测:

3.1) 测取新的 $\boldsymbol{y}(\cdot)$ 存入缓冲队列尾;

3.2) 对各 $LSSVM_j$ 增加一个样本学习:$lb_j=lb_j+1$,若 $LSSVM_j$ 的样本数 $lb_j=1$,则 $\boldsymbol{H}_j(1)^{-1}=1/(k(\boldsymbol{x}_{jt},\boldsymbol{x}_{jt})+1/C_j)$,否则用式(19 ~ 22)计算 $\boldsymbol{H}_j(l)^{-1}$;时刻 t 与样本编号 l 不必对应;

3.3) 给定新的 $\boldsymbol{u}(\cdot)$ 存入缓冲队列尾;

3.4) 对各 $LSSVM_j$ 如下处理:学完两个样本($lb_j\geqslant 2$)后就利用式(9)算出 b、$\boldsymbol{a}$ 建立模型(10)或(13),且用它们来计算下一时刻的预测值 $\hat{y}_j(\cdot)$;

3.5) 对各 $LSSVM_j$ 如下处理:若 lb_j 等于 $SWLY_j$ 则利用式(23)消除一个老样本,并且 $lb_j=lb_j-1$;

3.6) 若在进入 3.5 时出现某个 lb_j 等于 $MSWL$,则将 MIMO 输入输出向量 $\boldsymbol{u}(\cdot)$、$\boldsymbol{y}(\cdot)$ 缓冲区中所有数据前挪一个位置,抛弃位于前面的最早时刻数据。

4) 转向 3.1)。

3　仿真实验及分析

3.1　时变非线性 *MIMO* 系统在线辨识

假设某时变非线性 MIMO 系统可用如下差分方程:

$$y_1(t)=y_1(t-1)/(1+y_2^2(t-1))+u_1(t-1)+u_2(t-1)\times\sin(2\pi t/50)+e_1(t)$$

$$y_2(t)=y_1(t-1)y_2(t-1)/(1+y_2^2(t-1))+u_2(t-1)-u_1(t-1)\times\mathrm{mod}(t,50)/50+e_2(t)$$

其中 $e_1(t)$ 和 $e_2(t)$ 皆为(-0.05, 0.05)中均匀分布白色噪声序列,mod()是整数求模函数。

输入信号 $u_1(t)$ 在仿真时段内为周期为 50 的方波,其第一个周期如下:

$$u_1(t) = \begin{cases} 2, & 0 \leqslant t \leqslant 25 \\ 0, & 25 < t < 50 \end{cases}$$

输入信号 $u_2(t)$ 是叠加三角函数波,形式为:

$$u_2(t) = 2\sin(2\pi t/25) + 2\sin(2\pi t/50) + 2\sin(2\pi t/100)$$

设定各 $LSSVM_j(j = 1,2)$ 有关参数:$\beta_1^2 = 3, \beta_2^2 = 4, C_1 = C_2 = 500, SWLY_j = 50, lb_j = 0$。模型阶次参数:

$$\boldsymbol{ydl} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \boldsymbol{yol} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\boldsymbol{udl} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \boldsymbol{uol} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

仿真步数 $SM = MYUL + 200$。显然此处有 $MYUL = 2, n = k = 2$。

下面考察得出模型的预测效果。图 1 是在线辨识模型对 $y_1(t)$、$y_2(t)$ 的一步预测结果。

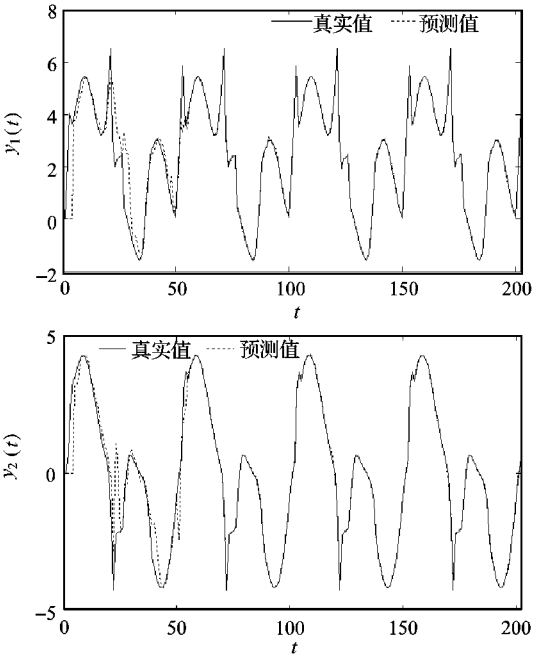

图 1 模型预测曲线与过程输出曲线

定义模型预测误差 $e_j(t) = y_j(t) - \hat{y}_j(t), (j = 1,2)$,图 2 是预测误差 $e_j(t)$ 的变化曲线。

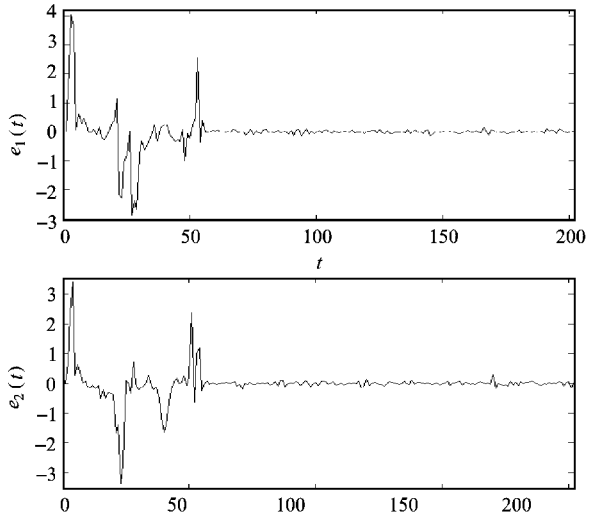

图 2 模型预测误差曲线

从图 1 和 2 可看出约在时刻 50 后各 LSSVM 模型预测精度较高。

3.2 实验结果对比分析

作者利用文献[9]的在线式算法和文献[12]的在线 MSVR 建模方法对上面两输出系统建模预测,其中用文献[9]方法时每个输出用一个 LSSVM 建模(下面简称直接法)。文献[12]算法的超参数 C 要根据滑动时间窗长度 $SWLY$ 和 δ[12] 反推确定。现将三种建模预测方法历经 SM 步仿真的运行时间和预测误差列于表 1,其中预测误差 $error$ 定义为:

$$error = \sum_{j=1}^{k} \frac{\sum_{t=MYUL+SWLY_{j+1}}^{SM} | y_j(t) - \hat{y}_j(t) |}{SM - MYUL - SWLY_j}$$

表 1 算法运行时间和预测误差比较

滑动时间窗长度	运行时间/s			预测误差		
	本文方法	直接法	在线 MSVR	本文方法	直接法	在线 MSVR
25	0.9113	1.7024	0.8524	1.3115	1.3115	1.8451
50	1.6624	4.6066	1.7454	0.0911	0.0911	0.8654
75	2.6538	8.0716	2.8712	0.0756	0.0756	0.5521
100	3.5251	12.2176	3.3245	0.0680	0.0680	0.7534
125	3.9957	16.7341	3.8721	0.0727	0.0727	0.8274
150	4.6467	20.6697	4.4651	0.0822	0.0822	0.7547

算法运行机器环境为:奔腾 CPU 1.6 GHz + 750 MB 内存 + Windows Professional 2000 + Matlab 7.01。此处各 $LSSVM$ 取等长的滑动时间窗长度 $SWLY$。为公平比较,在 $SWLY$ 取相同值的一轮对比仿真实验中,每次在 MIMO 系统两输出上叠加相同的噪声序列 $e1(t)$、$e2(t)$。从表 1 可见,本文引入加速策略的方法速度比直接法快,而且滑动时间窗越长,速度提高越明显;本文方法速度与在线 MSVR 方法相差不大。本文方法精度与直接法一样高,因为它只是去掉文献[9]在线算法中的大量重复或冗余的计算;本文方法精度比在线 MSVR 要高很多,因为 MSVR 方法用最小化即时风险(非结构风险)的方法训练 MSVR,而且删除老样本采用的是近似方法,该方法波动性大[12],靠牺牲精度提高速度。从上可见,本文方法综合性能优于其他两种方法。

有关参数的说明:设定各 $LSSVM_j$ 有关参数 β_j^2、C_j 靠试探确定;在能使预测精度满意的情况下,滑动时间窗长度 $SWLY_j$ 越短越好,这一方面可减少运算量,另一方面可摆脱老数据的影响;模型结构参数不能自动辨识,yol_{ij}、uol_{ij} 越小越好,ydl_{ij}、udl_{ij} 越大越好,这可减少 $LSSVM_j$ 嵌入维数,从而减少运算量,这些参数也靠试探确定。

4 结语

本文在分析现有 LSSVM 增量式和在线式学习算法的基础上,去掉其中不必要的和重复的计算,减少运算量,提高计算速度,得出快速在线式学习算法,并将执行这种快速在线式学习算法的 LSSVM 用于 MIMO 系统的在线建模,建模速度快,得出的模型预测精度高。

参考文献:

[1] MOHD-MOKHTAR R, WANG LIUPING. System identification of MIMO magnetic bearing via continuous time and frequency response data[C]// Proceedings of IEEE International Conference on Mechatronics. Taibei: IEEE, 2005: 191 - 196.

[2] 黄德先,金以慧. 基于小波神经网络的通用多变量非线性系统辨识算法和应用[J]. 控制理论与应用, 2001, 18(Suppl): 63 - 68.

(下转第 2314 页)

接收者等安全属性是否合理；

2)基于消息中的认证信息元素，分以下几种情况处理：若为X.509证书，检查证书及其相关签名的有效性，将请求进一步传递给访问控制模块；若为用户/口令、SAML票据[6]，直接将解析结果传递给后续应用；

3)若请求消息经过签名、加密，采用XML解密技术解密，其中密钥管理遵循XKMS规范；

4)经上述处理的请求消息传递给后续处理过程。

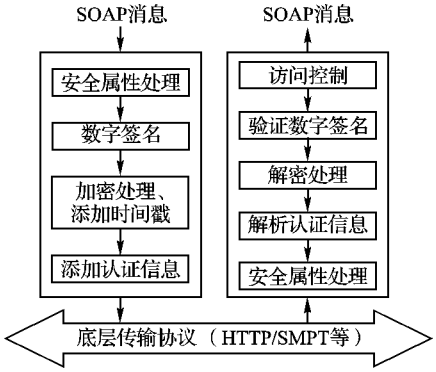

图3 SOAP消息安全性框架

3 应用前景

图4描述了s-AMM在手机移动(端用户)支付电子文献检索的应用场景，其中文献提供商(如维普)拥有电子文献检索服务，移动运营商(如中国移动)拥有面向手机用户的小额支付服务，银联能够为企业之间的资金往来提供可靠的交易服务。

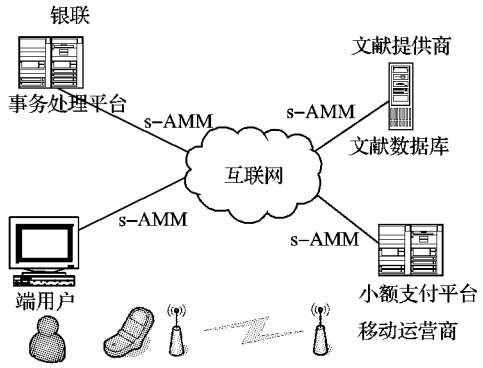

图4 s-AMM在移动支付电子文献检索中的应用

上述移动电子案例的网络环境相对复杂。文献提供商可通过Internet连接到处访问；银联事务处理平台支持远程终端访问，对安全和可靠性要求很高；移动运营商的服务器分布在全国各地，支持无线网络连接；端用户可以是手机或PDA。显然，这个案例对独立、异步、可靠性和灵活数据传输有很高要求，传统消息中间件虽有具有一定的异步传输和可靠特性，但没有采用面向服务架构因而不适于解决这类跨企业应用集成。面向服务的异步消息中间件s-AMM采用面向服务架构，为移动电子商务中不同的平台间进行业务沟通数据的传递提供松耦合、异步和安全可靠保证，为最终基于面向服务架构实现跨企业应用集成提供支持。

由于s-AMM体系结构可扩展性，其公共服务和私有服务可结合具体的网络环境要求定制，s-AMM的这种灵活性是传统消息中间件不具备的。

随着s-AMM的不断完善，面向服务的异步消息中间件在大规模复杂网络环境必将具有更广阔的应用前景。

4 结语

本文设计了一种面向服务的异步消息中间件s-AMM，并深入研究了消息传输算法，服务重组和安全服务等关键技术。面向服务的异步消息中间件有很多关键技术，限于篇幅仅选择上述几个具有特色的关键技术予以说明，以突出s-AMM的可扩展，松耦合，异步传输和安全可靠等特点。

下一步研究工作主要围绕以下两方面展开：1)面向服务的异步消息中间件的体系结构是具有可扩展性，但目前涵盖公共服务和私有服务仍属最小核心服务集合，需要进一步引入容错和实时等服务，提升对企业应用集成的支撑能力。2)随着s-AMM在大规模网络环境中应用的推广，在保证性能总体稳定的基础上，需要大力优化性能。

参考文献：

[1] COLLEEN E, DAVE C, DOUG D. Web services reliability ver2003 [EB/OL]. [2008-08-01]. http://otn.oracle.com/tech/webservices/htdocs/spec/WS-ReliabilityV1.0.pdf.

[2] RUSLAN B, ADAM B, DON B. Web services reliable messaging protocol (WS-Reliable Messaging) [EB/OL]. [2008-09-01]. ftp://www6.software.ibm.com/software/developer/library/ws-reliablemessaging200403.pdf.

[3] 苗春雨，史美林，姜进磊. MOM-S：基于Web服务的消息中间件系统[J]. 通信学报，2006，27(11)：96-105.

[4] WARWICK F, PHILLIP H, BARBARA F, *et al*. XML key management specification [EB/OL]. [2001-03-30]. http://www.w3.org/TR/xkms/XML Key Management Specification (XMMS).htm.

[5] EASTLAKE D, REAGLE J, SOLO D. XML-signature syntax and processing: W3C recommendation[EB/OL]. [2008-09-01]. http://www.w3.org/TR/2002/REC-xmldsig-core-20020212.

[6] ROB P. Security and privacy considerations for the OASIS security assertion markup language (SAML) v1.1 [EB/OL]. [2008-09-01]. http://www.oasis-open.org/committees/documents.php?wg_abbrev=security. 2003-09-12.

[7] SU U. Java message service specification [EB/OL]. [2008-09-01]. http://java.sun.com/products/jms/2006.

(上接第2284页)

[3] 舒华，舒怀林. 基于PID神经网络的多变量非线性动态系统辨识[J]. 计算机工程与应用，2006，42(12)：47-49.

[4] 朱文彪，孙增圻. 一种MIMO复杂过程的模糊建模新方法[J]. 系统工程与电子技术，2005，27(1)：97-99.

[5] 蔡艳宁，胡昌华. 辨识非线性MIMO系统的多输出ε-SVR模型研究[J]. 控制与决策，2008，23(7)：813-816，822.

[6] SUYKENS J A K. Least squares support vector machine classifiers [J]. Neural Process Letter, 1999, 9(3): 293-299.

[7] SUYKENS J A K. Nonlinear modeling and support vector machines [C]// IEEE Instrumentation and Measurement Technology Conference. Budapest, Hungary: IEEE, 2001: 287-294.

[8] VAPNIK V. Statistical learning theory[M]. New York: Wiley, 1998.

[9] 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.

[10] KEERTHI S S, SHEVADE S K. SMO algorithm for least squares SVM formulations[J]. Neural Computation, 2003, 15(2): 487-507.

[11] 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.

[12] 胡根生，邓飞其. 在线多输出支持向量回归及在投资决策中的应用[J]. 华南理工大学学报，2006，34(6)：64-68.