第 28 卷 第 4 期　　湖南科技大学学报(自然科学版)　　Vol. 28 No. 4
2013 年　12 月　　Journal of Hunan University of Science & Technology(Natural Science Edition)　　Dec.　2013

基于 Cholesky 分解的 LSSVM 在线学习算法

蒋星军[1,2],周欣然[3],唐钊轶[2]

(1. 北京工业大学 计算机学院,北京 100022;2. 湖南广播电视大学 计算机系,湖南 长沙 410004;
3. 中南大学 信息科学与工程学院,湖南 长沙 410075)

摘　要:针对最小二乘支持向量机(LSSVM)用于在线建模时存在的计算复杂性问题,提出一种 LSSVM 在线学习算法. 首先引入了基于 Cholesky 分解求 LSSVM 的方法,接着根据在线建模期间核函数矩阵的更新特点,将分块矩阵 Cholesky 分解用于 LSSVM 的在线求解,使三角因子矩阵在线更新从而得出一种新的 LSSVM 在线学习算法. 该算法能充分利用历史训练结果,减少计算量. 仿真实验显示了这种在线学习算法的有效性.

关键词:最小二乘支持向量机; 在线学习; Cholesky 分解; 滚动时间窗; 系统在线辨识

中图分类号:TP183 ,TP273　　**文献标识码**:A　　**文章编号**:1672 - 9102(2013)04 - 0074 - 04

最小二乘支持向量机[1-2]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凸规划问题求解运算量减少许多. 为将 LSSVM 用于系统,尤其是时变系统或工作点移动系统在线建模,往往取滑动时间窗中样本构成 LSSVM 的学习样本集,每一时刻采集一个新样本,同时抛弃最老的那个样本. 文献[4]利用分块矩阵求逆公式设计了 LSSVM 在线学习算法,每一步学习利用前一次学习的结果,与每一步直接求矩阵逆的方法相比大大减少了计算时间;文献[5]将在线 LSSVM 成功地用于预测控制中. 为了进一步提高 LSSVM 的在线建模速度,文献[6]将 LSSVM 结构风险变形,消去其偏置得出无偏置 LSSVM(NB - LSSVM),再利用分块矩阵 Cholesky 分解设计了 NB - LSSVM 在线学习算法,但 NB - LSSVM 增加了一个超参数 λ.

LSSVM 的建模效果与其超参数的值有关,当前的超参数选择方法,如人工蜂群算法[7]、基于小世界优化算法[8],只适用于时不变系统建模的 LSSVM 中的参数离线近似寻优;若系统时变,则 LSSVM 的参数也将变化,才能得出相对于各时刻(段)系统特性而言优化的 LSSVM;但目前尚无超参数在线寻优方法;因此,用 LSSVM 为时变系统在线建模时,往往凭经验设定 LSSVM 的超参数. NB - LSSVM 引入了一个新的超参数 λ[6],因此设定 LSSVM 超参数又多了一份主观性,增加了难度,λ 设置不当则降低无偏置 LSSVM 的预测精度.

本文将分块矩阵 Cholesky 分解用于 LSSVM 的在线求解,使分解出的因子矩阵在线更新,从而得出一种新的在线学习算法,该方法与文献[4]LSSVM 在线学习方法相比速度要快;与文献[6]动态无偏置 LSSVM 方法相比,本文方法每一步要多解 2 个系数矩阵为三角阵的方程组,因此速度略有下降,但它不增加超参数;因此,该方法具有快速易用的特点.

1　最小二乘支持向量机

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

收稿日期:2013 - 05 - 15
基金项目:国家自然科学基金(10971060);湖南省科学技术厅项目(2011FJ6033)
通信作者:蒋星军(1964 -),男,湖南长沙人,硕士,高级工程师,主要从事智能控制与信息管理系统的教学与研究. E - mail:445214558@qq.com

$X \to F$,将输入数据映射到一个高维特征空间 F,使输入空间 X 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

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

根据结构风险最小化原则,求回归问题变为约束优化问题:

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

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

式中,$C(>0)$ 是规范化因子. 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:

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

根据 KKT 条件有:

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

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

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

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

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/\beta^2). \qquad (6)$$

式中,$\beta > 0$,称为核参数;$\boldsymbol{Q}$ 称为核矩阵. 记 $\boldsymbol{H} = \boldsymbol{Q} + C^{-1}\boldsymbol{I}$,首先解方程[9]:

$$\boldsymbol{H\rho} = \boldsymbol{e}1. \qquad (7a)$$

$$\boldsymbol{H\eta} = \boldsymbol{y}. \qquad (7b)$$

得出[9]:

$$b = \frac{\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{\eta}}{\boldsymbol{e}1^{\mathrm{T}}\boldsymbol{\rho}} = \frac{\sum_{i=1}^{N}\eta_i}{\sum_{i=1}^{N}\rho_i}. \qquad (8a)$$

$$\boldsymbol{a} = \boldsymbol{\eta} - \boldsymbol{\rho}b. \qquad (8b)$$

回归函数(1)化为

$$y(\boldsymbol{x}) = \sum_{i=1}^{N} a_i k(\boldsymbol{x}_i,\boldsymbol{x}) + b. \qquad (9)$$

当样本集不大时,不必采用文献[9]的共轭梯度法求解 LSSVM. 本文采用 Cholesky 分解法求解方程(7a),(7b). 因 $\boldsymbol{Q}$ 是半正定阵,故 $\boldsymbol{H}$ 是对称正定阵,可将 $\boldsymbol{H}$ 进行 Cholesky 分解,即 $\boldsymbol{H}$ 可唯一地分解为 $\boldsymbol{H} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}$,其中 $\boldsymbol{U}$ 为上三角矩阵,式(7b)可分 2 步求解:从 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{p} = \boldsymbol{y}$ 解出 $\boldsymbol{p}$,再从 $\boldsymbol{U\eta} = \boldsymbol{p}$ 解出 $\boldsymbol{\eta}$,具体由下式计算:

$$P_i = (y_i - \sum_{k=1}^{i-1} u_{ki}p_k)/u_{ii},\ i = 1,\cdots,N;$$

$$\eta_i = (P_i - \sum_{k=i+1}^{N} u_{ik}\eta_k)/u_{ii},\ i = N,\cdots,1. \qquad (10)$$

式(7a)同样求解.

2 最小二乘支持向量机的在线学习算法

本文将上面基于 Cholesky 分解求解 LSSVM 的方法改造成在线形式,每当增加一个样本或减少一个样本时,$\boldsymbol{U}$ 沿用文献[6]的规则递增或缩减地计算,只是本文 $\boldsymbol{U}$ 含义不同而已. 因每步要解(7a),(7b)2 个方程组,所以本文每步比文献[6]多解 2 个系数矩阵为三角阵的方程组. 现将增加样本或消减样本时 $\boldsymbol{U}$ 的递推计算过程描述如下.

2.1 增加样本

设在 t 时刻学习了 l 个样本,则

$$\boldsymbol{H}(t) = \boldsymbol{Q}(t) + C^{-1}\boldsymbol{I} = \begin{bmatrix} k(\boldsymbol{x}_{t-l+1},\boldsymbol{x}_{t-l+1}) + 1/C & \cdots & k(\boldsymbol{x}_{t-l+1},\boldsymbol{x}_t) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_t,\boldsymbol{x}_{t-l+1}) & \cdots & k(\boldsymbol{x}_t,\boldsymbol{x}_t) + 1/C \end{bmatrix}_{l\times l}. \qquad (11)$$

由 t 时刻的学习结果有 $\boldsymbol{H}(t) = \boldsymbol{U}^{\mathrm{T}}(t)\boldsymbol{U}(t)$,可见 $\boldsymbol{U}(t)$ 已算出. 在 $t+1$ 时刻增加了一个新样本,有

$$\boldsymbol{H}(t+1) = \begin{bmatrix} \boldsymbol{H}(t) & \boldsymbol{V}(t+1) \\ \boldsymbol{V}(t+1)^{\mathrm{T}} & v(t+1) \end{bmatrix}_{(l+1)\times(l+1)}. \qquad (12)$$

式中,$\boldsymbol{V}(t+1) = [k(\boldsymbol{x}_{t-l+1},\ \boldsymbol{x}_{t+1}),\cdots,\ k(\boldsymbol{x}_t,\boldsymbol{x}_{t+1})]$,$v(t+1) = k(\boldsymbol{x}_{t+1},\boldsymbol{x}_{t+1}) + 1/C$,$\boldsymbol{H}(t+1)$ 的 Cholesky 分解因子

$$\boldsymbol{U}(t+1) = \begin{bmatrix} \boldsymbol{U}(t) & \boldsymbol{W}(t+1) \\ 0^{\mathrm{T}} & w(t+1) \end{bmatrix}. \qquad (13)$$

式中,$\boldsymbol{W}(t+1)$ 和 $w(t+1)$ 分别是 l 维列向量和实数,利用 $\boldsymbol{H}(t+1) = \boldsymbol{U}^{\mathrm{T}}(t+1)\boldsymbol{U}(t+1)$ 分别算出

$$W_i(t+1) = (V_i(t+1) - \sum_{k=1}^{i-1} U_{ki}(t)W_k(t+1))/U_{ii}(t),\ i = 1,\cdots,l. \qquad (14)$$

$$w(t+1) = (v(t+1) - \sum_{k=1}^{l} W_k^2(t+1))^{1/2}. \qquad (15)$$

由上可知,计算 $\boldsymbol{H}(t+1)$ 的 Cholesky 分解因子 $\boldsymbol{U}(t+1)$ 可以充分利用 $\boldsymbol{U}(t)$,而不必重新分解.

2.2 消减样本

在 $t+1$ 时刻后将 $\boldsymbol{H}(t+1)$ 重新分块如下:

$$\boldsymbol{H}(t+1)=\begin{bmatrix}\bar{v}(t-l+1) & \bar{\boldsymbol{V}}^{\mathrm{T}}(t+1)\\ \bar{\boldsymbol{V}}(t+1) & \bar{\boldsymbol{H}}(t+1)\end{bmatrix}_{(l+1)\times(l+1)}. \quad (16)$$

式中，$\bar{\boldsymbol{H}}(t+1)$ 是去掉最早样本所形成式(7)的系数矩阵，又设 $\bar{\boldsymbol{H}}(t+1)$ 的分解为 $\bar{\boldsymbol{H}}(t+1)=\boldsymbol{R}^{\mathrm{T}}(t+1)\boldsymbol{R}(t+1)$，求 $\boldsymbol{R}(t+1)$ 是消减样本过程的关键. 另一方面，将 $\boldsymbol{U}(t+1)$ 也按相同的划分重新分块如下：

$$\boldsymbol{U}(t+1)=\begin{bmatrix}\bar{w}(t-l+1) & \bar{\boldsymbol{W}}^{\mathrm{T}}(t+1)\\ 0 & \bar{\boldsymbol{U}}(t+1)\end{bmatrix}. \quad (17)$$

利用 $\boldsymbol{H}(t+1)=\boldsymbol{U}^{\mathrm{T}}(t+1)\boldsymbol{U}(t+1)$ 得

$$\bar{\boldsymbol{H}}(t+1)=\bar{\boldsymbol{U}}^{\mathrm{T}}(t+1)\bar{\boldsymbol{U}}(t+1)+\bar{\boldsymbol{W}}(t+1)\bar{\boldsymbol{W}}^{\mathrm{T}}(t+1). \quad (18)$$

故有

$$\boldsymbol{R}^{\mathrm{T}}(t+1)\boldsymbol{R}(t+1)=\bar{\boldsymbol{U}}^{\mathrm{T}}(t+1)\bar{\boldsymbol{U}}(t+1)+\bar{\boldsymbol{W}}(t+1)\bar{\boldsymbol{W}}^{\mathrm{T}}(t+1). \quad (19)$$

式中，$\bar{\boldsymbol{U}}(t+1)\in\boldsymbol{R}^{l\times l}$ 和 $\bar{\boldsymbol{W}}^{\mathrm{T}}(t+1)\in\boldsymbol{R}^{l}$ 均已知，利用 Cholesky 分解的 update 算法[10,11,6]求 $\boldsymbol{R}(t+1)$ 方法为

$$\boldsymbol{J}(l),\cdots,\boldsymbol{J}(i),\cdots,\boldsymbol{J}(1)\begin{bmatrix}\bar{\boldsymbol{W}}^{\mathrm{T}}(t+1)\\ \bar{\boldsymbol{U}}(t+1)\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}^{T}\\ \boldsymbol{R}(t+1)\end{bmatrix}. \quad (20)$$

式中，$\boldsymbol{J}(i)$ 是 $l+1$ 维的 Givens 矩阵，其元素确定方法如下（设其下标 m,n 从 0 开始）：

$$J_{m,n}(i)=\begin{cases}1, & m=n\neq 0,i;\\ s_i, & m=n=0,i;\\ -c_i, & m=0,n=i;\\ c_i, & m=i,n=0;\\ 0, & \text{其它}.\end{cases} \quad (21)$$

式中，s_i,c_i 参照文献[11,6]确定，目的是为了使 $\boldsymbol{J}(i)$ 与其右乘矩阵的积矩阵首行第 i 个元素为0. 该算法求 $\boldsymbol{R}(t+1)$ 的时间复杂度为 $O(l^2)$[12,10]，而对 $\bar{\boldsymbol{H}}(t+1)$ 重新进行 Cholesky 分解的时间复杂度为 $O(l^3)$.

2.3 完整的在线学习算法

将 LSSVM 在线学习算法归纳如下：

1）选定 LSSVM 的核函数、超参数，滑动时间窗长度 N，被建模对象阶次参数估计值及 LSSVM 的输入向量 $\boldsymbol{x}$ 结构.

2）按增加样本计算公式学习初始的 N 个样本：若当前样本只有 1 个，则 $\boldsymbol{U}(t)_{1\times1}=\sqrt{\boldsymbol{H}(t)_{1\times1}}=\sqrt{k(\boldsymbol{x}_t,\boldsymbol{x}_t)+1/C}$，否则利用式(13)增加样本更新算法求 $\boldsymbol{U}(t)_{l\times l}$，$l\leqslant N$. 其间获得 2 个样本，即 $l\geqslant 2$ 后就可用式(10)，式(8)算出 $\boldsymbol{a}$、b，建立模型(9)，用 $\boldsymbol{x}_{t+1}$ 预测 $\hat{y}_{t+1}$.

3）在线学习滑动时间窗中的 N 个样本并预测：

① 抛弃最老的样本，即利用式(17)，式(20)从 $\boldsymbol{U}(t)_{N\times N}$ 求 $\boldsymbol{R}(t)_{(N-1)\times(N-1)}$；

② 测出 y_{t+1}，增加新样本 $(\boldsymbol{x}_{t+1},y_{t+1})$，将 $\boldsymbol{R}(t)_{(N-1)\times(N-1)}$ 看作 $\boldsymbol{U}(t)_{(N-1)\times(N-1)}$，利用式(13)求 $\boldsymbol{U}(t+1)_{N\times N}$；

③ 用式(10)，式(8)算出 $\boldsymbol{a}$，b，建立模型(9)用于预测下一时刻的输出.

4）转向 3）进行重复的学习和预测.

3 仿真实验

下面用执行本文在线学习算法的 LSSVM 对时变系统在线建模，考察在线 LSSVM 的预测精度和计算时间. 设待建模的时变非线性过程在仿真期间可用如下差分方程表示：

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

输入信号 $u(t)$ 是叠加三角函数波，形式为

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

设定 LSSVM 的超参数 $\beta^2=4$；$C=500$；$N=50$；仿真步数 $SM=654$；取 $y(1)=y(2)=y(3)=0$. 本文在线 LSSVM 的预测结果如图 1 所示. 利用预测误差均值 $mpe=\dfrac{1}{SM-nl-N}\sum_{T=nl+N+1}^{SM}|y_t-\hat{y}_t|$ 来反映建模预测方法效果，其中 $nl=3$. $mpe=0.073\,431\,064\,803\,04$，而采用文献[4]的在线 *LSSVM* 时 $mpe=0.073\,431\,064\,803\,27$，可见 2 种方法计算精度相当.

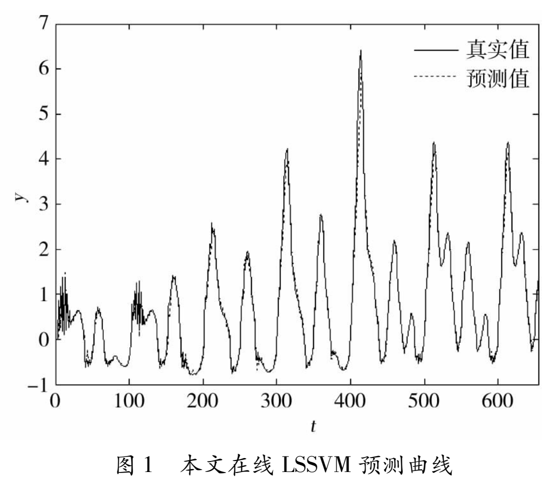

图 1 本文在线 LSSVM 预测曲线

Fig. 1 Prediction results of the proposed online LSSVM

下面分别将本文在线 LSSVM、文献[4]在线 LSSVM 和文献[6]动态无偏 LSSVM 在时间窗长度 N 取不同值时的仿真运行时间 T 绘制于图 2.

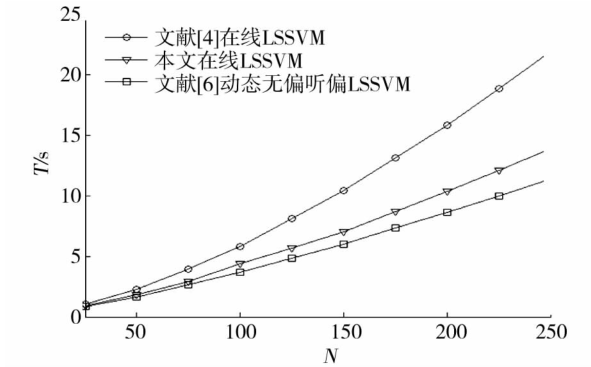

图 2　各算法运行时间曲线

Fig. 2 Running time of those algorithms

由图 2 可见,本文方法比文献[4] LSSVM 在线学习方法要快,比文献[6]动态无偏 LSSVM 方法略慢,但本文方法不增加超参数.

4　结论

本文首先介绍了 LSSVM,引入基于 Cholesky 分解求解 LSSVM 的方法,接着根据模型动态变化过程中核函数矩阵的特点,将分块矩阵 Cholesky 分解用于 LSSVM 的在线求解,使分解的因子矩阵在线更新从而得出一种新的 LSSVM 在线学习算法. 与基于分块矩阵求逆的 LSSVM 在线训练算法相比,本文方法具有较快的运算速度;与基于分块矩阵 Cholesky 分解的无偏置 LSSVM 在线训练算法相比,本文方法速度略有下降,但它不增加超参数,设置超参数相对容易. 这为 LSSVM 应用于系统在线辨识和时间序列预测等场合时提供了一种新的在线学习算法.

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]//Proceedings of IEEE Instrumentation and Measurement Technology Conference. Budapest:Hungary, 2001.

[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.
Zhang H R, Wang X D. Incremental and online learning algorithm for regression least squares support vector machine [J]. China Journal of Computers,2006,29(3):400 – 406.

[5] Li L J, Su H Y, Chu J. Generalized predictive control with online least squares support vector machines[J]. Acta Automatica Sinica , 2007,33(11):1182 – 1188.

[6] 蔡艳宁,胡昌华. 一种基于 Cholesky 分解的动态无偏 LS2SVM 学习算[J]. 控制与决策,2008,23(12):1363 – 1367.
Cai Y N, Hu C H. Dynamic non – bias LS – SVM learning algorithm based on Cholesky factorization[J]. Control and Decision,2008,23 (12):1363 – 1367.

[7] Mohd H S, Mohd W M, Hussain S, et al. An application of artificial bee colony algorithm with least squares support vector machine for real and reactive power tracing in deregulated power system[J]. International Journal of Electrical Power and Energy Systems, 2012,37(1):67 – 77.

[8] Mao W T, Yan G R, Dong L L, et al. Model selection for least squares support vector regressions based on small – world strategy [J]. Expert Systems with Applications,2011, 38(4):3227 – 3237.

[9] Chu W, Ong C J, Keerthi S S. An improved conjugate gradient scheme to the solution of least squares SVM[J]. IEEE Transactions on Neural Networks, 2005,16(2):498 – 501.

[10] Matthias S. Low rank updates for the cholesky decomposition[R]. Tuebingen: Max Planck Society, 2005.

[11] Gill P E, Golub G H, Murray W, et al. Methods for modifying matrix factorizations[J]. Mathematics of Computation,1974,126 (28):505 – 535.

[12] Golub G H, Loan C F V. Matrix computations [M]. Baltimore: The Johns Hopkins University Press, 1996.

An online learning algorithm for LSSVM based on Cholesky factorization

JIANG Xing – jun[1,2] ,ZHOU Xin – ran[3] ,TANG Zhao – yi[2]

(1. School of Computer, Beijing University of Technology, Beijing 100022, China;

2. Department of Computer, Hunan Radio & TV University, Changsha 410004, China;

3. School of Information Science and Engineering, Central South University, Changsha 410075, China)

Abstract: Aiming at the computational complexity of least squares support vector machine(LSSVM)'s online modeling, an online learning algorithm for LSSVM was proposed. First, the solution of LSSVM through the Cholesky factorization was introduced, then the Cholesky factorization of partitioned matrix was applied to the online solution of LSSVM according to the updating character of kernel function matrix during online modelling, and triangle factor matrix was renewed online, consequently, a novel online learning algorithm for LSSVM was obtained. The improved learning algorithm can make full use of the historical training results and reduce the computation amount. The numerical simulation results the validity of the online learning algorithm for LSSVM.

Key words: least square support vector machine; online learning; Cholesky factorization; sliding time window; system online identification