

基于LS-SVM方法求高阶线性ODE近似解

周水生, 王保军, 安亚利

ZHOU Shuisheng, WANG Baojun, AN Yali

西安电子科技大学 数学与统计学院, 西安 710071

School of Mathematics and Statistics, Xidian University, Xi'an 710071, China

ZHOU Shuisheng, WANG Baojun, AN Yali. High order linear ODE approximate solution based on LS-SVM method. Computer Engineering and Applications, 2018, 54(23):51-56.

Abstract: For linear ordinary differential equations, the analytical solution is convenient for qualitative analysis and practical application. However, most of the differential equations are not available in many cases. The regression method is applied to obtain approximate analytical solution, and the Least Square Support Vector Machine (LS-SVM) method is demonstrated the efficiency of the proposed method over existing methods. But this method requires not only to take the higher derivative of the kernel but also to solve a large linear equation system. Concerning higher order Ordinary Differential Equations (ODE), the available approach is the reduction of the problem to a system of first-order differential equations and then the system is solved by LS-SVM regression method. These model parameters are adjusted to minimize an error function, finally, the high precision approximate solution (continuous and differentiable) is obtained by solving three system of linear equations. Experimental results verify the effectiveness of the proposed method.

Key words: approximate analytical solution; kernel function; least square support vector machine; ordinary differential equations; system of ordinary differential equations

摘 要: 对于线性常微分方程, 解析解方便定性分析和实际应用, 然而大多数微分方程没有解析解。回归的方法被应用获取近似解析解, 其中最小二乘支持向量机 (LS-SVM) 是目前为止最好的方法。但是该方法不仅需要对核函数求高阶导数而且需要求解一个大的线性方程组。为此, 把高阶线性常微分方程转化为一阶线性常微分方程组, 构建含有一阶导数形式的 LS-SVM 回归模型。该模型利用最小化误差函数去获得合适的参数, 最终通过求解三个小的线性方程组获得高精度的近似解 (连续、可微)。实验结果验证了该方法的有效性。

关键词: 近似解; 核函数; 最小二乘支持向量机; 常微分方程; 常微分方程组

文献标志码: A **中图分类号:** O241.8 **doi:** 10.3778/j.issn.1002-8331.1712-0326

1 引言

在工程研究和科学计算中, 大多数微分方程很难得到解析解。为了简化计算又能满足一定的实际需求, 数值解的方法被应用。常用的数值方法有^[1]: 欧拉法, 龙格-库塔法, 有限差分法, 打靶法以及配置法等。龙格-库塔法实质是 Taylor 展式的变形, 函数越光滑精度越高, 常用的 ode45 就是具有 4 阶精度的龙格-库塔法。虽然数值解得到广泛应用, 但解的形式离散, 需要经过额外的插值过程获得整个区域的解, 同时为了获取更高精度的

数值解, 则需要不断减小步长, 这些都增加了计算量。

近年, 随着计算机技术的发展, 一些新的智能算法被应用近似求解微分方程。这些方法基于不同的回归模型并利用优化方法求解该模型。它们克服了传统方法的缺陷, 获得封闭连续可微的近似解析解。如基于遗传算法求解常微分方程^[2], 神经网络方法求解微分方程^[3-6], 无监督核最小平方算法求解常微分方程^[7]。尽管这些方法有很好的效果, 但也有一些缺点, 如神经网络的方法无法确定隐藏单元的数量, 并且容易陷入局部最

基金项目: 国家自然科学基金 (No.61772020)。

作者简介: 周水生 (1972—), 男, 博士, 教授, 研究领域为最优化理论、算法及应用, 智能信息处理、机器学习、支持向量机等; 王保军 (1985—), 男, 硕士, 研究领域为最优化理论, 机器学习、支持向量机等, E-mail: wangbaojun0916@126.com。

收稿日期: 2017-12-22 **修回日期:** 2018-03-27 **文章编号:** 1002-8331(2018)23-0051-06

CNKI 网络出版: 2018-07-25, <http://kns.cnki.net/kcms/detail/11.2127.TP.20180724.1711.004.html>

小值而无法得到令人满意的结果。

支持向量(SVM)^[8]是一种机器学习算法基于VC维理论和结构风险最小化,由Vapnik在1995年提出。该算法适用于解决小样本的分类和回归问题,具有很强的泛化能力。然而,对于大样本数据,它不得不求解一个复杂的二次规划(QP)问题而花费大量时间。Suykens^[9]提出LS-SVM把不等式约束转化为等式约束,最终求解一个线性方程组,避免了求解复杂的QP问题,同时通过减少稀疏性而加快计算速度。文献[10]应用LS-SVM模型近似求解常微分方程,该方法将常微分方程问题转换为含有导数的目标优化问题,再构建LS-SVM回归模型。由于该方法对线性常微分方程有良好的性能,从而推广到求解线性时变广义系统^[11],参数估计延迟微分方程^[12],以及偏微分方程^[13]。然而,该方法对于非线性微分方程,需要与其他方法相结合^[10],否则需要改进模型^[14-15],不易求解。另一方面,对于高阶线性常微分方程,该方法需要对核函数求高阶导数^[10],从而对核函数提出了更高要求。

因此,将高阶线性常微分方程转化为一阶常微分方程组,构建含有一阶导数的LS-SVM模型,从而避免对核函数求高阶导数。为了方便比较和应用,称此模型为L-LS-SVM。在求解两点边值问题时,利用线性叠加原理^[16],将边值问题转化为两个初值问题,再利用该方法求解。

2 LS-SVM回归模型

2.1 LS-SVM回归算法

对于给定的训练集 $\{t_k, y_k\}, k=1, 2, \dots, N$, 其中 $t_k \in \mathbb{R}$, $y_k \in \mathbb{R}$ 分别为输入和输出数据。LS-SVM回归^[9]的目的就是获得估计函数 $\hat{y}(t) = \mathbf{w}^T \phi(t) + b$, 优化模型如下:

$$\min_{\mathbf{w}, b, e_k \in \Gamma} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{k=1}^N e_k^2 \quad (1)$$

$$\Gamma = \{(\mathbf{w}, b, e_k) | y_k = \mathbf{w}^T \phi(t_k) + b + e_k\}$$

这里 $\gamma \in \mathbb{R}^+$, $b \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^h$, $\phi(t_i): \mathbb{R} \rightarrow \mathbb{R}^h$ 把低维空间映射到 h 维特征空间。寻找适合的参数 \mathbf{w} 和 b 去近似训练数据集。

首先构造如下拉格朗日函数:

$$L(\mathbf{w}, b, e_k, \alpha_k) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{k=1}^N e_k^2 - \sum_{k=1}^N \alpha_k (\mathbf{w}^T \phi(t_k) + b + e_k - y_k)$$

其中 α_k 是拉格朗日乘子, 对变量 $\mathbf{w}, b, \alpha_k, e_k$, $k=1, 2, \dots, N$ 求偏导获得KKT优化条件, 利用该条件消去变量 \mathbf{w} 和 e_k , 整理后得到如下线性方程组:

$$\begin{bmatrix} \Psi + I_{N-1}/\gamma & I_N \\ 1_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \quad (2)$$

其中 $\Psi = K(x_i, y_j) = \phi(x_i)^T \phi(y_j)$, $I_N = [1, \dots, 1]^T \in \mathbb{R}^N$, $\mathbf{a} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ 并且 $\mathbf{y} = [y_1, y_2, \dots, y_N]$ 。最终近似解为:

$$\hat{y}(t) = \sum_{k=1}^N \alpha_k K(t_k, t) + b \quad (3)$$

这里 α_k 和 b 通过式(2)得到, $K(t_k, t)$ 是核函数。

应用上述方法求解微分方程时,需要对核函数求导数^[17]。根据Mercer核理论,并以高斯核函数 $K(x, y) = e^{-(x-y)^2/\sigma^2}$ 为例给出其偏导数:

$$\nabla_0 K(x, y) = \phi(x)^T \phi(y) = K(x, y)$$

$$\nabla_1^0 K(x, y) = \frac{\partial(\phi(x)^T \phi(y))}{\partial y} = \frac{2(x-y)}{\sigma} K(x, y)$$

$$\nabla_1^0 K(x, y) = \frac{\partial(\phi(x)^T \phi(y))}{\partial x} = -\frac{2(x-y)}{\sigma} K(x, y)$$

$$\nabla_1^1 K(x, y) = \frac{\partial(\phi(x)^T \phi(y))}{\partial y} = \left[\frac{2}{\sigma} - \left(\frac{2(x-y)}{\sigma^2} \right)^2 \right] K(x, y)$$

为了方便,并在后面章节中使用,对上述偏导做如下标记:

$$[\Psi_n^m]_{i,j} = \nabla_n^m K(p, q) \Big|_{p=t_i, q=t_j} = \frac{\partial^{m+n} K(p, q)}{\partial p^n \partial q^m} \Big|_{p=t_i, q=t_j} \quad (4)$$

$$[\Psi_n^0]_{i,j} = \nabla_n^0 K(p, q) \Big|_{p=t_i, q=t_j} = K(t_i, t_j) \quad (5)$$

$$[\Psi_n^m]_{1,2:N} = [\Psi_n^m]_{1,2}, \dots, [\Psi_n^m]_{1,N}, \tilde{\Psi}_n^m = [\Psi_n^m]_{2:N, 2:N} \quad (6)$$

这里 $[\Psi_n^m]_{i,j}$ 表示核矩阵 Ψ_n^m 的第 (i, j) 个元素。

2.2 应用LS-SVM模型求解常微分方程

给出如下 m 阶时变系数线性常微分方程:

$$y^{(m)}(t) - \sum_{k=1}^m f_k(t) y^{(m-k)}(t) = r(t), a \leq t \leq c \quad (7)$$

$$y(a) = p_1, y^{(l-1)}(a) = p_l, l=2, 3, \dots, m$$

不同于一般的LS-SVM回归模型,这里没有目标值,该模型无噪音,近似解可以通过下面的优化问题获得:

$$\min_{\mathbf{w}, b, e_i \in \gamma} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} \gamma \sum_{i=2}^N e_i^2 \quad (8)$$

$$\gamma = \left\{ (\mathbf{w}, b, e_i) \left\{ \begin{aligned} & \mathbf{w}^T \phi^{(m)}(t_i) = \mathbf{w}^T \left[\sum_{k=1}^m f_k(t_i) \phi_i^{(m-k)}(t_i) \right] + \\ & f_m(t_i) b + r(t_i) + e_i, i=2, 3, \dots, N \\ & \mathbf{w}^T \phi(t_1) + b = p_1 \\ & \mathbf{w}^T \phi^{(i-1)}(t_1) = p_i, i=2, 3, \dots, m \end{aligned} \right. \right\}$$

这里 $\gamma \in \mathbb{R}^+$, $b \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^h$, $\phi(t_i): \mathbb{R} \rightarrow \mathbb{R}^h$ 把低维空间映射到 h 维特征空间, $\phi^{(m)}(t_i): \mathbb{R} \rightarrow \mathbb{R}^h$ 是对映射函数求高阶导数。构建拉格朗日函数并求KKT条件,根据KKT条件消去 \mathbf{w}, e_i 获得式(8)的对偶解,详细过程可以参考文献[10]。上述方法,需要对核函数求高阶导数,这对核函数提出了更高要求。为了避免求高阶导数,给出了下面的方法。

3 边值问题求解

定义线性算子 T :

$$Ty = y'' - a_1(t)y'(t) - a_2(t)y(t)$$

二阶线性边值问题:

$$Ty = r_2(t), a \leq t \leq c, y(a) = p, y(c) = q \quad (9)$$

由于线性微分方程具有叠加性,它的解可以由一个非齐次的特解和一个齐次的基本解组合而来。应用解析法的思想,边值问题(9)转化为两个初值问题:

$$Tu = r_2(t), a \leq t \leq c, u(a) = p, u'(a) = 0 \quad (10)$$

$$Tv = 0, a \leq t \leq c, v(a) = 0, v'(a) = 1 \quad (11)$$

这两个初值问题的近似解为 $\bar{u}(t)$ 和 $\bar{v}(t)$, 若 $v(c) \neq 0$, 通过线性叠加得到原问题近似解:

$$\bar{y}(t) = \bar{u}(t) + \frac{q - u(c)}{v(c)} \bar{v}(t) \quad (12)$$

为了解 $\bar{u}(t)$ 和 $\bar{v}(t)$, 上面的初值问题(10)和(11)转化为如下方程组:

$$\begin{aligned} u'_1 &= u_2, u'_2 = a_{22}(t)u_2 + a_{21}(t)u_1 - r_2(t), \\ u_1(a) &= p, u_2(a) = 0 \end{aligned} \quad (13)$$

$$v'_1 = v_2, v'_2 = a_{22}(t)v_2 + a_{21}(t)v_1, v_1(a) = 0, v_2(a) = 1 \quad (14)$$

以式(13)为例,近似解 $\bar{u}_k(t) = w_k \phi(t) + b_k, k = 1, 2$ 的求解过程如下。

构造LS-SVM优化模型:

$$\min_{w_k, b_k, e_k} \frac{1}{2} \sum_{k=1}^2 w_k^T w_k + \frac{\gamma}{2} \sum_{k=1}^2 e_k^T e_k \quad (15)$$

$$\begin{aligned} \text{s.t.} \quad \begin{bmatrix} w_1^T \phi'(t_i) \\ w_2^T \phi'(t_i) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ a_{21}(t_i) & a_{22}(t_i) \end{bmatrix} \begin{bmatrix} w_1^T \phi(t_i) + b_1 \\ w_2^T \phi(t_i) + b_2 \end{bmatrix} + \\ &\begin{bmatrix} 0 \\ r_2(t_i) \end{bmatrix} + \begin{bmatrix} e_{1i} \\ e_{2i} \end{bmatrix} \\ \begin{bmatrix} w_1^T \phi(t_1) + b_1 \\ w_1^T \phi(t_1) + b_1 \end{bmatrix} &= \begin{bmatrix} p_1 \\ 0 \end{bmatrix}, i = 2, 3, \dots, N \end{aligned}$$

拉格朗日函数如下:

$$\begin{aligned} L(w_k, b_k, e_k, \xi_i, \alpha_{ki}, \beta_k) &= \frac{1}{2} \sum_{k=1}^2 w_k^T w_k + \\ &\frac{\gamma}{2} \sum_{k=1}^2 e_k^T e_k - \sum_{i=2}^N \alpha_{1i} (w_1^T \phi'(t_i) - w_2^T \phi(t_i) - b_2 - e_{1i}) - \\ &\sum_{i=2}^N \alpha_{2i} \left(w_2^T \phi'(t_i) - \sum_{l=1}^2 a_{2l}(t_i) (w_l^T \phi(t_i) + b_l) - r_2(t_i) - e_{2i} \right) - \\ &\beta_1 (w_1^T \phi(t_1) - b_1 - p) - \beta_2 (w_2^T \phi(t_1) - b_2) \end{aligned}$$

其中 β_k, α_{ki} 是拉格朗日乘子,对变量求导得到KKT条件,之后消去变量 $w, e_k, k = 1, 2$, 整理后得到线性方程组,详细过程可以参考下一章。利用核函数标记(4)~(6),将线性方程组写成矩阵形式:

$$\begin{bmatrix} K & H & -G_A \\ H^T & I_{2 \times 2} & I_{2 \times 2} \\ -G_A^T & I_{2 \times 2}^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ b \end{bmatrix} = \begin{bmatrix} r \\ p \\ 0 \end{bmatrix} \quad (16)$$

$$\alpha = [\alpha_1, \alpha_2]^T \in \mathbb{R}^{2(N-1)}, \alpha_k = [\alpha_{k2}, \dots, \alpha_{kN}]^T, k = 1, 2$$

$$r = [0, r_2]^T \in \mathbb{R}^{m(N-1)}, r_2 = [r_2(t_2), \dots, r_2(t_N)]^T$$

$$\beta = [\beta_1, \beta_2]^T, b = [b_1, b_2]^T, p = [p_1, 0]^T$$

$$K = \begin{bmatrix} K_s & K_u + K_{12} \\ K_l + K_{21} & K_{22} \end{bmatrix} \in \mathbb{R}^{2(N-1) \times 2(N-1)}$$

$$K_s = \Psi_1^T + \Psi_0^T + I_{N-1}/\gamma, K_u = -\Psi_1^T, K_l = -\Psi_0^T$$

$$K_{12} = -\Psi_0^T D_{21} + \Psi_0^T D_{21}, K_{21} = -D_{21} \Psi_1^T + D_{22} \Psi_0^T$$

$$K_{22} = \Psi_1^T - \Psi_0^T D_{22} + \sum_{k=1}^2 D_{2k} \Psi_0^T D_{2k} - D_{22} \Psi_1^T + I_{N-1}/\gamma$$

$$H = \begin{bmatrix} h_{p_s} & h_{p_u} \\ h_{p_{21}} & h_{p_{22}} \end{bmatrix} \in \mathbb{R}^{2(N-1) \times 2}, G_A = \begin{bmatrix} 0 & I_{(N-1) \times 1} \\ G_{A_{21}} & G_{A_{22}} \end{bmatrix} \in \mathbb{R}^{2(N-1) \times 2}$$

$$h_{p_s} = [\Psi_0^T]_{1,2:N}^T, h_{p_u} = -[\Psi_0^T]_{1,2:N}^T, h_{p_{22}} = [\Psi_0^T]_{1,2:N}^T - D_{22} [\Psi_0^T]_{1,2:N}^T$$

$$h_{p_{21}} = -D_{21} [\Psi_0^T]_{1,2:N}^T, G_{A_{2l}} = [a_{2l}(t_2), \dots, a_{2l}(t_N)]^T \in \mathbb{R}^{N-1}, l = 1, 2$$

$D_{ij} = \text{diag}(a_{ij})$ 是以 $a_{ij} = [a_{ij}(t_2), \dots, a_{ij}(t_N)]^T \in \mathbb{R}^{(N-1) \times 1}, i, j = 1, 2$ 为元素的对角阵。

为了避免大规模求解线性方程(16),给出如下变形:

$$\begin{cases} K\alpha + H\beta - G_A b = f \\ H^T \alpha + \beta + b = p \\ -G_A^T \alpha + \beta = 0 \end{cases} \rightarrow \begin{cases} P\alpha = f + G_A p \\ b = p - H^T \alpha - G_A^T \alpha \\ \beta = G_A^T \alpha \end{cases} \quad (17)$$

这里 $P = K + HG_A^T + G_A H^T + G_A G_A^T$, 线性方程(16)分解成三个小的线性方程(17)并求解,得到问题(15)的近似解如下:

$$\begin{aligned} \bar{u}_1(t) = \bar{u}(t) &= \sum_{i=2}^N \alpha_{1i} \Psi_1^0(t_i, t) - \sum_{i=2}^N \alpha_{2i} a_{21}(t_i) \Psi_0^0(t_i, t) + \\ &\beta_1 \Psi_0^0(t_1, t) + b_1 \end{aligned}$$

同理可以求出 $\bar{v}(t)$, 利用线性叠加性得到问题(9)的近似解(12)。

4 高阶初值常微分方程近似解

对于高阶线性常微分方程,将其转化为一阶的常微分方程组,构造LS-SVM模型求解,为了方便,称此模型L-LS-SVM。具体过程如下:

给出类似于式(7)的 m 阶线性时变系数常微分方程:

$$y^{(m)}(t) - \sum_{k=1}^m a_{mk}(t) y^{(m-k)}(t) = r(t)$$

$$a \leq t \leq c, y(a) = p_1, y^{(l-1)}(a) = p_l, l = 2, 3, \dots, m \quad (18)$$

为了避免对核函数求高阶导数,将上述线性常微分方程转化为如下微分方程组:

$$\begin{cases} \frac{dy_1}{dt} = y_2, \dots, \frac{dy_{m-1}}{dt} = y_m, \frac{dy_m}{dt} = \sum_{l=1}^m a_{ml}(t) y_l + r_m(t) \\ y_1(a) = p_1, y_2(a) = p_2, \dots, y_m(a) = p_m, a \leq t \leq c \end{cases} \quad (19)$$

对于问题(19),近似解 $\bar{y}_k(t) = w_k \phi(t) + b_k, k = 1, 2, \dots, m$ 可以通过下面的LS-SVM模型获得:

$$\min_{w_k, b_k, e_k} \frac{1}{2} \sum_{k=1}^m w_k^T w_k + \frac{\gamma}{2} \sum_{k=1}^m e_k^T e_k \quad (20)$$

$$\begin{bmatrix} w_1^T \phi(t_i) \\ \vdots \\ w_{m-1}^T \phi(t_i) \\ w_m^T \phi(t_i) \end{bmatrix} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \\ a_{m1}(t_i) & a_{m2}(t_i) & \cdots & a_{mm}(t_i) \end{bmatrix} \begin{bmatrix} w_1^T \phi(t_i) + b_1 \\ \vdots \\ w_{m-1}^T \phi(t_i) + b_{m-1} \\ w_m^T \phi(t_i) + b_m \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ r_m(t_i) \end{bmatrix} + \begin{bmatrix} e_{1i} \\ \vdots \\ e_{(m-1)i} \\ e_{mi} \end{bmatrix}$$

$$\begin{bmatrix} w_1^T \phi(t_1) + b_1 \\ \vdots \\ w_m^T \phi(t_1) + b_m \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix}, i = 2, 3, \dots, N$$

构造拉格朗日函数:

$$L(w_k, b_k, e_k, \xi_i, \alpha_{ki}, \beta_k) = \frac{1}{2} \sum_{k=1}^m w_k^T w_k + \frac{\gamma}{2} \sum_{k=1}^m e_k^T e_k - \sum_{k=1}^{m-1} \sum_{i=2}^N \alpha_{ki} (w_k^T \phi(t_i) - w_{k+1}^T \phi(t_i) - b_{k+1} - e_{ki}) - \sum_{i=2}^N \alpha_{mi} \left(w_m^T \phi(t_i) - \sum_{l=1}^m a_{ml}(t_i) (w_l^T \phi(t_i) + b_l) - r_m(t_i) - e_{mi} \right) - \sum_{k=1}^m \beta_k (w_k^T \phi(t_1) - b_k - p_k)$$

其中 β_k, α_{ki} 是拉格朗日乘子。

对拉格朗日函数求导数,得到KKT条件如下:

$$\begin{aligned} \frac{\partial L}{\partial w_k} = 0 &\rightarrow w_k = \sum_{i=2}^N \alpha_{ki} \phi'(t_i) - \sum_{i=2}^N \alpha_{(k-1)i} \phi(t_i) - \sum_{i=2}^N \alpha_{mi} a_{mk}(t_i) \phi(t_i) + \beta_k \phi(t_1), \alpha_{0i} = 0 \\ \frac{\partial L}{\partial b_k} = 0 &\rightarrow \beta_k = \sum_{i=2}^N \alpha_{(k-1)i} + \sum_{i=2}^N \alpha_{mi} a_{mk}(t_i), \\ k = 1, 2, \dots, m, \alpha_{0i} &= 0 \\ \frac{\partial L}{\partial e_{ki}} = 0 &\rightarrow e_{ki} = -\frac{\alpha_{ki}}{\gamma}, i = 2, 3, \dots, N, k = 1, 2, \dots, m \\ \frac{\partial L}{\partial \alpha_{ki}} = 0 &\rightarrow w_k^T \phi'(t_i) - w_{k+1}^T \phi(t_i) - b_{k+1} - e_{ki} = 0, \\ i = 2, 3, \dots, N, k = 1, 2, \dots, m-1 \\ \frac{\partial L}{\partial \alpha_{mi}} = 0 &\rightarrow 0 = w_m^T \phi'(t_i) - \sum_{k=1}^m a_{mk}(t_i) (w_k^T \phi(t_i) + b_k) - r_m(t_i) - e_{mi}, i = 2, 3, \dots, N \\ \frac{\partial L}{\partial \beta_k} = 0 &\rightarrow p_k = w_k^T \phi(t_1) + b_k, k = 1, 2, \dots, m \end{aligned}$$

之后消去变量 $w_k, e_k, k = 1, 2, \dots, m$, 利用核函数标记(4)~(6), 整理后得到如下矩阵:

$$\begin{bmatrix} K & H & -G_A \\ H^T & I_{m \times m} & I_{m \times m} \\ -G_A^T & I_{m \times m}^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ b \end{bmatrix} = \begin{bmatrix} r \\ p \\ 0 \end{bmatrix} \quad (21)$$

$$\alpha = [\alpha_1, \alpha_2, \dots, \alpha_m]^T \in \mathbb{R}^{m(N-1)}, \alpha_k = [\alpha_{k2}, \dots, \alpha_{kN}]^T \in \mathbb{R}^{N-1}$$

$$k = 1, 2, \dots, m, r = [r_1, r_2, \dots, r_m]^T \in \mathbb{R}^{m(N-1)}$$

$$r_k = [r_k(t_2), \dots, r_k(t_N)]^T \in \mathbb{R}^{N-1}, k = 1, 2, \dots, m$$

$$\beta = [\beta_1, \beta_2, \dots, \beta_m]^T, b = [b_1, b_2, \dots, b_m]^T,$$

$$p = [p_1, p_2, \dots, p_m]^T$$

$$K = \begin{bmatrix} K_s & K_u & 0 & \cdots & 0 & K_{1m} \\ K_l & K_s & K_u & \cdots & 0 & K_{2m} \\ 0 & K_l & K_s & \cdots & 0 & K_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & K_s & K_{(m-1)m} + K_u \\ K_{m1} & K_{m2} & K_{m3} & \cdots & K_{m(m-1)} + K_l & K_{mm} \end{bmatrix}$$

$$K \in \mathbb{R}^{m(N-1) \times m(N-1)}, K_s = \Psi_1^1 + \Psi_0^0 + I_{m-1}/\gamma, K_u = -\Psi_1^0$$

$$K_l = -\Psi_1^0, K_{km} = -\Psi_1^0 D_{mk} + \Psi_0^0 D_{m(k+1)}, k = 1, 2, \dots, m-1$$

$$K_{mk} = -D_{mk} \Psi_1^0 + D_{m(k+1)} \Psi_0^0, k = 1, 2, \dots, m-1$$

$$K_{mm} = \Psi_1^1 - \Psi_1^0 D_{mm} + \sum_{k=1}^m D_{mk} \Psi_0^0 D_{mk} - D_{mm} \Psi_1^0 + I_{N-1}/\gamma$$

$$H = \begin{bmatrix} h_{p_s} & h_{p_u} & \cdots & 0 & 0 \\ 0 & h_{p_s} & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & h_{p_s} & h_{p_u} \\ h_{p_{m1}} & h_{p_{m2}} & \cdots & h_{p_{m(m-1)}} & h_{p_{mm}} \end{bmatrix} \in \mathbb{R}^{m(N-1) \times m}$$

$$h_{p_s} = [\Psi_1^1]_{1,2:N}^T, h_{p_u} = -[\Psi_0^0]_{1,2:N}^T$$

$$h_{p_{mk}} = -D_{mk} [\Psi_0^0]_{1,2:N}^T, k = 1, 2, \dots, m-1$$

$$h_{p_{mm}} = [\Psi_1^1]_{1,2:N}^T - D_{mm} [\Psi_0^0]_{1,2:N}^T$$

$$G_A = \begin{bmatrix} 0 & I_{(N-1) \times 1} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & I_{(N-1) \times 1} \\ G_{A_{m1}} & G_{A_{m2}} & \cdots & G_{A_{mm}} \end{bmatrix} \in \mathbb{R}^{m(N-1) \times m}$$

$$G_{A_{ml}} = [a_{ml}(t_2), \dots, a_{ml}(t_N)]^T \in \mathbb{R}^{N-1}, l = 1, 2, \dots, m$$

$D_{ij} = \text{diag}(a_{ij})$ 是以 $a_{ij} = [a_{ij}(t_2), \dots, a_{ij}(t_N)]^T \in \mathbb{R}^{(N-1) \times 1}$ 为元素的矩阵。为了避免大规模求解线性系统(21),给出如下变形:

$$\begin{cases} K\alpha + H\beta - G_A b = f \\ H^T \alpha + I_{m \times m} \beta + I_{m \times m} b = p \rightarrow \\ -G_A^T \alpha + I_{m \times m}^T \beta = 0 \end{cases} \quad (22)$$

$$\begin{cases} Q\alpha = f + G_A b \\ I_{m \times m} b = p - H^T \alpha - G_A^T \alpha \\ I_{m \times m}^T \beta = G_A^T \alpha \end{cases}$$

这里 $Q = K + HG_A^T + G_A H^T + G_A G_A^T$ 为实对称矩阵。把线性方程(21)分解成三个小的线性方程(22)并求解,得到问题(20)的近似解如下:

$$\bar{y}_1 = \bar{y}(t) = \sum_{i=2}^N \alpha_{1i} \Psi_1^0(t_i, t) - \sum_{i=2}^N \alpha_{mi} a_{m1}(t_i) \Psi_0^0(t_i, t) + \beta_1 \Psi_0^0(t_1, t) + b_1 \tag{23}$$

$$\bar{y}_k = \frac{d^{k-1} \bar{y}(t)}{dt^{k-1}} = \sum_{i=2}^N \alpha_{ki} \Psi_1^0(t_i, t) - \sum_{i=2}^N \alpha_{(k-1)i} \Psi_0^0(t_i, t) - \sum_{i=2}^N \alpha_{mi} a_{mk}(t_i) \Psi_0^0(t_i, t) + \beta_k \Psi_0^0(t_1, t) + b_k, \tag{24}$$

$k = 2, 3, \cdots, m$

5 数值实验

LS-SVM算法的性能依赖于核间隔参数 σ 和正则化参数 γ 。对于线性常微分方程求解,由于没有目标值,LS-SVM算法是无噪音。当 γ 取一个很大值时,损失误差 e 将足够小,从而获得更高精度的解,一般取 γ 值大于 10^7 。为了方便比较,无特别说明,实验中取200个测试点,参数 $\gamma = 10^{10}$, $\sigma = 2$ 。用于测试精度的观察点来自精确值。近似解的精度通过均值平方误差 $MSE = \frac{1}{M} \sum_{i=1}^M (y(t_i) - \bar{y}(t_i))^2$ 、最大绝对误差 $\|y(t_i) - \bar{y}(t_i)\|_{\infty}$,

$i = 1, 2, \cdots, M$ 、标准差 $STD_{test} = \sqrt{\frac{1}{M} \sum_{i=1}^M (y(t_i) - \bar{y}(t_i))^2}$ 来测量,这里 M 为测试点的个数。

通过一个边值问题和两个高阶初值问题的常微分方程来验证L-LS-SVM方法的有效性,并和文献[10]做了比较。MATLAB 2014a用于实现代码,所有计算都在Intel-core i7-4790 CPU和8.00 GB RAM的Windows 7系统上进行。

5.1 边值问题常微分方程求解

例1 考虑边值问题线性常微分方程:

$$y'' = -ty' + 4y + 12t^2 - 3t, 0 \leq t \leq 1, y(0) = 0, y(1) = 2$$

该边值问题的精确解为 $y(t) = t^4 + t$, 利用叠加原理,把上述边值问题转化为两个初值问题的微分方程:

$$u'' = -tu' + 4u + 12t^2 - 3t, 0 \leq t \leq 1, u(0) = 0, u'(1) = 0$$

$$v'' = -tv' + 4v, 0 \leq t \leq 1, v(0) = 0, v'(1) = 1$$

并再次转化为下面两个微分方程组:

$$\begin{cases} u_1' = u_2 \\ u_2' = -tu_2 + 4u_1 + 12t^2 - 3t \\ u_1(0) = 0, u_2(0) = 0, 1 \leq t \leq 2 \end{cases} \quad \begin{cases} v_1' = v_2 \\ v_2' = -tv_2 + 4v_1 \\ v_1(0) = 0, v_2(0) = 1, 0 \leq t \leq 1 \end{cases}$$

利用LS-SVM算法分别求出这两个线性常微分方程组的近似解 $\bar{u}_1(t)$ 和 $\bar{v}_1(t)$, 最终利用线性叠加性得到原问题的近似解 $\bar{y}(t) = \bar{u}_1(t) + \frac{2 - u_1(1)}{\bar{v}_1(1)} \bar{v}_1(t)$ 。

图1是例1的数值实验,在区间[0, 1]内取10个等距的训练点,图1(a)为近似解和精确解实验对比曲线,图1(b)为近似解和精确解的偏差值 $E(t) = y(t) - \bar{y}(t)$ 。表1给出了训练点多少对近似解精度的影响。

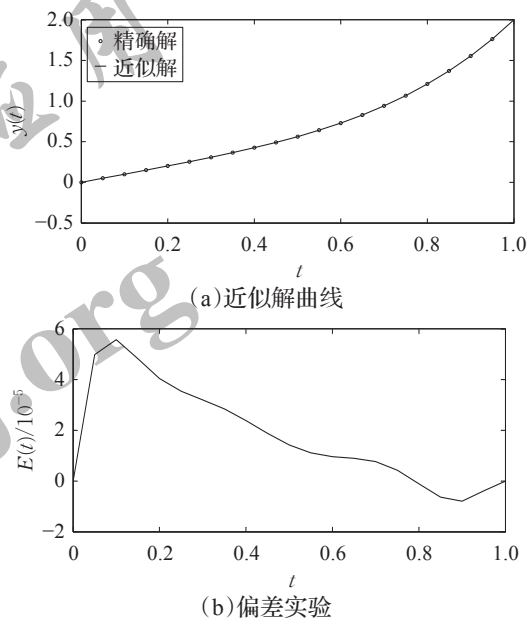


图1 例1的数值实验

5.2 高阶线性常微分方程

两个例子被给出验证L-LS-SVM方法的有效性,并和文献[10]中的方法做比较,说明两种方法得到近似解精度相当,具体结果见下面的实验和数据。

例2 考虑二阶时变系数的常微分方程:

$$y'' - \frac{1}{t}y' = \frac{3}{t^3}, t \in [1, 2], y(1) = 2, y'(1) = 1$$

转化为如下方程组: $y_1' = y_2, y_2' = \frac{1}{t}y_2 + \frac{3}{t^3}, y(1) = 2, y'(1) = 1$,

该方程的解析解为: $y(t) = \frac{1}{t} + t^2$ 。

图2是例2的数值实验,在区间[1, 2]内取10个等距的训练点,图2(a)为近似解和精确解在区间内外对比曲线,图2(b)为近似解和精确解在区间内的偏差 $E(t) =$

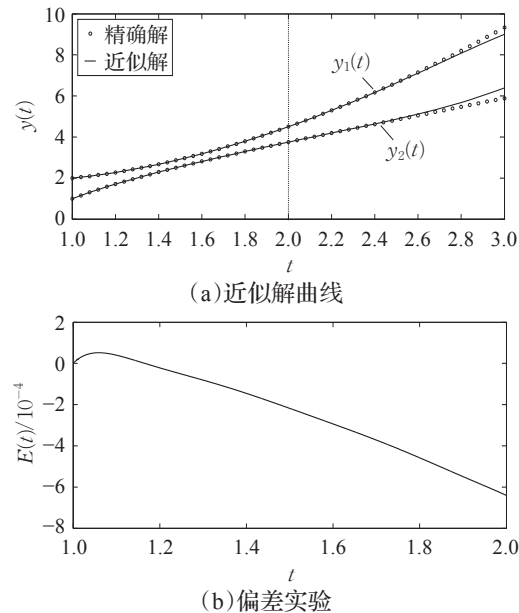


图2 例2的数值实验

$y(t) - \bar{y}(t)$ 。图3当 $\gamma = 10^{10}$ 时给出了核间隔参数 σ 对近似解精度的影响曲线。表1给出了训练点多少对近似解精度的影响。文献[10]中的方法是目前求近似解最好的方法,因此一个详细的比较在表2中给出。

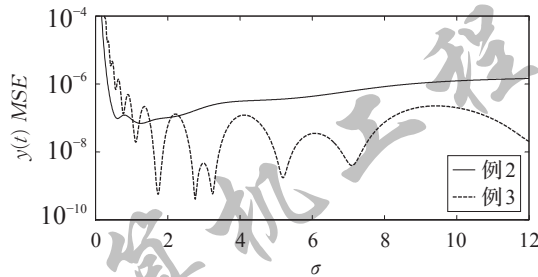


图3 初值问题的核函数间隔参数 σ 的敏感性实验

表1 训练点多少对 $y(t)$ 的 MSE 影响

N	10	20	160	320
例1	7.13×10^{-10}	9.09×10^{-12}	1.51×10^{-13}	8.52×10^{-14}
例2	1.01×10^{-7}	1.62×10^{-9}	1.14×10^{-12}	2.18×10^{-13}
例3	8.14×10^{-9}	8.20×10^{-11}	1.14×10^{-13}	6.41×10^{-14}

表2 两种方法误差比较

	方法	$\ y - \bar{y}\ _{\infty}$	MSE	STD
例2	L-LS-SVM	6.40×10^{-4}	1.01×10^{-7}	2.10×10^{-4}
	文献[10]	6.29×10^{-4}	1.10×10^{-7}	1.86×10^{-4}
例3	L-LS-SVM	3.04×10^{-4}	5.63×10^{-8}	5.57×10^{-5}
	文献[10]	3.60×10^{-4}	6.60×10^{-8}	1.30×10^{-5}

例3 考虑三阶时变系数的常微分方程:

$$y''' + ty'' + (t+1)y' + (t-1)y = 24t + 12t^3 + 4(t+1)t^3 + (t-1)t^4$$

$$y(0) = 0, y'(0) = 0, y''(0) = 0, t \in [0, 3]$$

把上面的常微分方程转化为微分方程组:

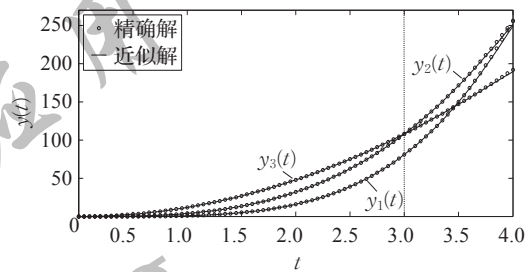
$$\begin{cases} y_1' = y_2, y_2' = y_3, y_3' = -(t-1)y_1 - (t+1)y_2 - ty_3 + \\ (t+1)y_3 + 24t + 12t^3 + 4(t+1)t^3 + (t-1)t^4 \\ y_1(0) = 0, y_2(0) = 0, y_3(0) = 0 \end{cases}$$

原方程的精确解析解为 $y(t) = t^4$ 。图3当 $\gamma = 10^{10}$ 时给出了核间隔参数 σ 对近似解精度的影响曲线。

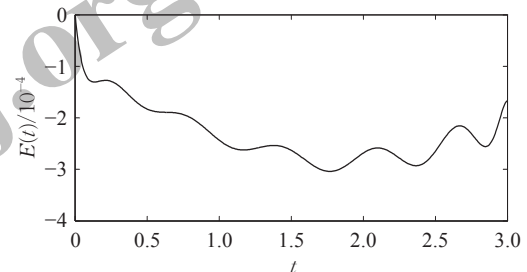
图4在区间[0, 3]内取30个等距的训练点,图4(a)为近似解和精确解在区间内外对比曲线,图4(b)为近似解和精确解在区间内的偏差 $E(t) = y(t) - \bar{y}(t)$,具体实验数据以及与文献[10]的比较在表2中给出。

6 结束语

对于高阶线性常微分方程,应用LS-SVM方法求解时,需要对核函数求高阶导数,为此将高阶线性常微分方程转化为一阶线性微分方程组,构建LS-SVM模型去求解该线性微分方程组,从而避免了对核函数求高阶导数。对于边值问题,利用解析的方法将它转化为两个初值问题的微分方程组求解。实验结果证明了L-LS-SVM方法的有效性,并和文献[10]中的方法做比较,说明两



(a) 近似解曲线



(b) 偏差实验

图4 例3的数值实验

种方法得到近似解精度相当。将来该方法可以推广求解任意阶线性常微分方程组。

参考文献:

- [1] Butcher J C. Numerical methods for ordinary differential equations in the 20th century[M]. [S.l.]: Elsevier Science Publishers, 2000.
- [2] Diver D A. Applications of genetic algorithms to the solution of ordinary differential equations[J]. Journal of Physics A General Physics, 1993, 26(14): 3503.
- [3] Ramuhalli P, Udupa L, Udupa S S. Finite-element neural networks for solving differential equations[J]. IEEE Transactions on Neural Networks, 2005, 16(6): 1381-1392.
- [4] Tsoulos I G, Gavrilis D, Glavas E. Solving differential equations with constructed neural networks[J]. Neurocomputing, 2009, 72(10/12): 2385-2391.
- [5] Mcfall K S, Mahan J R. Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions[J]. IEEE Transactions on Neural Networks, 2009, 20(8): 1221-1233.
- [6] Yazdi H S, Pourreza R. Unsupervised adaptive neural-fuzzy inference system for solving differential equations[J]. Applied Soft Computing, 2010, 10(1): 267-275.
- [7] Sadoghi Yazdi H, Pakdaman M, Modaghegh H. Unsupervised kernel least mean square algorithm for solving ordinary differential equations[J]. Neurocomputing, 2011, 74(12/13): 2062-2071.
- [8] Vapnik V N. The nature of statistical learning theory[M]. New York: Springer-Verlag, 1995.
- [9] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural Processing Letters, 1999, 9(3): 293-300.

(下转第73页)