IJAMML 6:2 (2017) 87-101 June 2017
ISSN: 2394-2258
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/ijamml_7100121844

NUMERICAL SOLUTION BASED ON LS-SVR FOR HIGH ORDER LINEAR ODES WITH VARIABLE COEFFICIENTS

Lichao Xia[a], Kaihui Li[b], Wenjing Xia[b] and Jianqiang Gao[c]

[a]Developing Zone Experimental Primary School, Liaocheng, Shandong, 252000, P. R. China

[b]School of Mathematics Sciences, Liaocheng University, Shandong, 252059, P. R. China

[c]College of Computer and Information, Hohai University, Nanjing, 211100, P. R. China

Abstract

In this paper, an efficient, novel and simple method is proposed to approximately solve linear ordinary differential equations (ODEs), emphasis is put on high-order linear ODEs with variable coefficients. What's more, by using the least squares support vector regression (LS-SVR) method, numerical solution based on LS-SVR for high order linear ODEs with variable coefficients can be easily solve. The following illustrative examples are given to demonstrate the effectiveness and high accuracy of we proposed method.

Keywords: SVR, variable coefficients, numerical solution, LS-SVR, ODE.

1. Introduction

High order linear ODEs with variable coefficients can be found in the mathematical formulation of physical phenomena in a wide variety of

*Corresponding author.
E-mail address: xiawenjing001@126.com (Lichao Xia).

2010 Mathematics Subject Classification: 68W40, 68Q25.
Received June 15, 2017; Revised July 03, 2017

applications especially in science and engineering. Depending upon the form of the boundary conditions to be satisfied by the solution, problems involving ODEs can be divided into two main categories, namely, initial value problems (IVPs) and boundary value problems (BVPs). Exact solutions for these problems are not generally available. Hence, besides numerical approaches, some approximate methods are usually as powerful tools for solving initial-value or boundary-value problems associated with ordinary differential equations (ODEs) [1].

Kernel least squares support vector machines (LS-SVMs) are a potent methodology for solving pattern recognition and function estimation problems, which based on Vapnik and Chervonenkis structural risk minimization principle [2]. They show better widespread ability compared to other machine learning methods on a wide variety of real-world problems, such as image segmentation [3], optimal control [4], time series prediction [5], pattern classification [5, 6], matrix learning [7] and so on. In this way, one maps data into a high dimensional feature space H by introducing a kernel function and solves a linear regression problem in H, which leads to solving quadratic programming problems. The function estimation by regression is of great importance in many fields of research such as bioinformatics, relevant literature such as DNA microarray gene expression data [8], control theory, relevant literature such as non-parametric regression and density estimation under control of modality [9], economics, such as Statistical Analysis of Multiple Regression with Crisp and Applications in Analyzing Economic Data of China [10], information science and signal processing. The main challenge in developing a useful regression model is to capture accurately the underlying functional relationship between the given inputs and their output values. Once the resulting model is obtained it can be used as a tool for analysis, simulation and prediction.

As far as the numerical solution of high order linear ODEs with variable coefficients is concerned, there has many kinds of other method, such as the Adomian decomposition method, Taylor expansion approach,

Volterra integral equation [11], an improved starting step of the G-B-S-method [12], two-parameter families of predictor-corrector methods [13] and so on. In this paper, in order to improve the speed of solve such problems, we utilize LS-SVR to solve above problem, it use the idea in LS [14] and SVR [15], it should be pointed out that our LS-SVR, not only improve the speed of calculate, but also insure the accuracy.

This paper is organized as follows. Section 2 briefly introduce the kernel LS-SVR and some concepts. Section 3 proposes numerical solution based on LS-SVR. Numerical examples and experimental results are described in Section 4. Finally, concluding remarks are given in Section 5.

2. Kernel LS-SVR and Some Concepts

Consider an m-order variable coefficient linear ordinary differential equations:

$$\sum_{l=0}^{m} f_l(t)y^{(l)}(t) = r(t), \quad t \in [a, c], \tag{1}$$

where $f_0(t), \cdots, f_m(t), r(t)$ in Equation (1) are variable coefficients and $y^{(l)}(t)$ is the l-order derivative of $y(t)$. The corresponding initial value problems is

$$\sum_{l=0}^{m} f_l(t)y^{(l)}(t) = r(t), \quad t \in [a, c],$$

$$y^{(l)}(a) = p_l, \quad l = 0, \cdots, m-1. \tag{2}$$

Set $L(y(t)) = \sum_{l=0}^{m} f_l(t)y^{(l)}(t)$, we call $L(y)$ as m-order linear differential operator, we main consider how to utilize LS-SVR get the approximate solution of Equation (2), so we suppose Equation (2) have the solution looks like $\bar{y}(t) = w^T\varphi(t) + b$. Let $k : R^n \times R^n \to R$ be a Gaussian radial

basis function (RBF) kernel function with the reproducing kernel Hilbert space (RKHS) H and the nonlinear feature mapping $\phi : R^n \to H$, apparently, $k(u, v) = \varphi(u)^T \varphi(v)$, $\forall u, v \in R$. Next, we will utilize LS-SVR to learn: $(\omega, b) \in H \times R$. Discrete time $[a, c]$ interval: $a = t_1 < t_2 < \cdots < t_N = c$, we seem $\{t_i\}_{i=1}^N$ as input data, so we suppose $w = \sum_{i=1}^N \beta_i \varphi(t_i)$, if $\overline{y}(t)$ is the exact solution of Equation (2), so we have $L(\overline{y}(t_i)) = r(t_i)$, $i = 1, \cdots, N$, $\overline{y}^{(l)}(a) = p_l$, $l = 0, \cdots, m-1$, generally speaking, $L(\overline{y}(t_i))$ may not equal to $r(t_i)$, $i = 1, \cdots, N$, $\overline{y}^{(l)}(a)$ may not equal to p_l, $l = 0, \cdots, m-1$, so we hope $|L(\overline{y}(t_i)) - r(t_i)|$, $i = 1, \cdots, N$ and $|y^{(l)}(a) - p_l|$, $l = 0, \cdots, m-1$, as small as possible. So we consider the following optimization problem:

$$\min_{\beta, b} \frac{1}{2} \sum_{i=1}^N (L(\overline{y}(t_i)) - r(t_i))^2,$$

or we can consider

$$\min_{\beta, b} \frac{1}{2} \sum_{i=1}^N (L(\overline{y}(t_i)) - r(t_i))^2$$

$$s.t.\ \overline{y}^{(l)}(a) = p_l,\ l = 0, \cdots, m-1.$$

In order to explain the problem more clearly, let

$$\nabla_n^m k(u, v) = \frac{\partial^{n+m} k(u, v)}{\partial u^n \partial v^m}, \quad \forall u, v \in R,\ n, m = 1, 2, \cdots,$$

then $\nabla_n^m k(u, v) = (\phi^{(n)}(u))^T \phi^{(m)}(v)$, $\forall u, v \in R$, $n, m = 1, 2, \cdots$, for special

$$\nabla_0^0 k(u, v) = \varphi(u)^T \varphi(v) = k(u, v), \quad \forall u, v \in R,$$

$$\nabla_0^1 k(u, v) = \varphi(u)^T \varphi'(v), \quad \forall u, v \in R,$$

$$\nabla_1^0 k(u, v) = \varphi'(u)^T \varphi(v), \ \forall u, v \in R,$$

and let

$$\beta = (\beta_1, \cdots, \beta_N)^T \in R^N,$$

$$K = [k(t_i, t_j)] \in R^{N\times N},$$

$$K_n^m = [\nabla_n^m k(t_i, t_j)] = [\nabla_n^m k(u, v)|_{u=t_i, v=t_j}] \in R^{N\times N},$$

thus

$$K_0^0 = K \in R^{N\times N},$$

$$K_0^1 = [\nabla_0^1 k(t_i, t_j)] = [\varphi(t_i)^T \varphi'(t_j)] \in R^{N\times N},$$

$$K_1^0 = [\nabla_1^0 k(t_i, t_j)] = [\varphi'(t_i)^T \varphi(t_j)] \in R^{N\times N}.$$

If we use $(K_n^m)_i$ indicate the column of matrix K_n^m, so we can get

$$\overline{y}(t_i) = w^T \varphi(t_i) + b = \sum_{j=1}^N \beta_j \varphi^T(t_j)\varphi(t_i) + b = \sum_{j=1}^N \beta_j k(t_j, t_t) = (K_0^0)_i^T \beta + b,$$

$$\overline{y}^{(l)}(t_i) = w^T \varphi^{(l)}(t_i) = \sum_{j=1}^N \beta_j \varphi^T(t_j)\varphi^{(l)}(t_i) = \sum_{j=1}^N \beta_j \nabla_0^l k(t_j, t_t) = (K_0^l)_i^T \beta,$$

$i = 1, \cdots, N, \ l = 1, \cdots, m,$

and then

$$L(\overline{y}(t_i)) = \sum_{l=0}^m f_l(t_i)\overline{y}^{(l)}(t_i) = (\sum_{l=0}^m f_l(t_i)(K_0^l)^T \beta + f_0(t_i)b, \ i = 1, \cdots, N.$$

3. Numerical Solution Based on LS-SVR

In this section, we mainly study how to find numerical solutions of the Equation (2) by using LS-SVR. Specifically, we assume that the numerical solution of the Equation (2) has the form $\overline{y}(t) = w^T \varphi(t) + b$ and

wish that the unknown $w \in H$ and $b \in R$ can be learned by kernel LS-SVR. For this purpose, we first discrete the domain $[a, b]$ of the Equation (2) into a set of collocation points $a = t_1 < t_2 < \cdots < t_N = c$ with the same step size $h_0 = \frac{c-a}{N-1}$, such that $\mu = t_n \in (t_1, t_N)$. Let

$$L(y(t)) = \sum_{l=0}^{m} f_l(t) y^{(l)}(t).$$

We hope the smaller the better of the values $|L(\bar{y}(t_i)) - r(t_i)|$, $i = 1, \cdots, N$, and $|y^{(l)}(a) - p_l|$, $l = 0, \cdots, m-1$. For this end, we view $\{(x_j, g(x_j))\}_{j=1}^{l}$ as the sample set and consider the following optimization modellings. According to the discussion in Section 2, we know $w = \sum_{i=1}^{l} \beta_i \varphi(t_i)$.

In order to insure the values $|L(\bar{y}(x_i)) - r(x_i)|$, $i = 1, \cdots, l$ and $|y^{(l)}(a) - p_l|$, $l = 0, \cdots, m-1$, being as small as possible, we consider the following optimization problem:

$$\begin{aligned} &\min_{\beta} \frac{1}{2} \beta^T K \beta + \frac{C}{2} \sum (L(\hat{y}(t_i)) - g(t_i))^2, \\ &s.t. (K_0^l)_1^T \beta = p_l, \; l = 1, \cdots, m-1, \\ &p_0 = (k)_1^T \beta + b, \end{aligned} \tag{3}$$

where $C > 0$ is a pre-specified value. Because $p_0 = k_1^T \beta + b$, we can know $b = p_0 - K_1^T \beta$, and then

$$L(\bar{y}(x_i)) - r(t_i) = (\sum_{l=0}^{m} f_l(t_i)(K_0^l)_i - f_0(t_i)(K)_1)^T \beta + f_0(t_i) p_0 - r(t_i).$$

Let

$$m_i = \sum_{l=0}^{m} f_l(t_i)(K_0^l)_i - f_0(t_i)(K)_1 \in R^N, \; i = 1, \cdots, N,$$

$$M = [m_1, m_2, \cdots, m_N] \in R^{N\times N},$$

$$p = (p_1, p_2, \ldots p_m - 1)^T \in R^{m-1},$$

$$A = [(K_0^1)_1, (K_0^2)_1, \cdots, (K_0^m - 1)_1] \in R^{N\times(m-1)},$$

$$q_i = f_0(t_i)p_0 - r(t_i),\ i = 1, \cdots, N,$$

$$q = [q_1, q_2, \cdots, q_N] \in R^N.$$

So, we can get

$$\sum_{i=1}^{N}(L(\bar{y}(x_i)) - r(t_i))^2 = \sum_{i=1}^{N}(m_i^T\beta + q_i)^2 = (M^T\beta + q)^T(M^T\beta + q).$$

Then the Equation (3) can be expressed with matrix:

$$\min_{\beta} \frac{1}{2}\beta^T K\beta + \frac{C}{2}(M^T\beta + q)^T(M^T\beta + q)$$

$$s.t.\ A^T\beta = p. \tag{4}$$

We can write down the Lagrange function of Equation (4):

$$L(\beta, \alpha) = \frac{1}{2}\beta^T K\beta + \frac{C}{2}(M^T\beta + q)^T(M^T\beta + q) + \alpha^T(A^T\beta - p),\ \forall\alpha \in R^{m-1}$$

$$= \frac{1}{2}\beta^T(K + CMM^T)\beta + (CMq + A\alpha)^T\beta - p^T\alpha + \frac{C}{2}q^Tq.$$

Let $\frac{\partial L}{\partial\beta} = (K + CMM^T)\beta + CMq + A\alpha = 0$, then $\beta = -(K + CMM^T)^{-1}(CMq + A\alpha)$ and the Lagrange function can be written as

$$L(\alpha) = -\frac{1}{2}(CMq + A\alpha)^T(K + CMM^T)^{-1}(CMq + A\alpha) - p^T\alpha + \frac{C}{2}q^Tq.$$

Put $D = (K + CMM^T)^{-1} \in R^{N\times N}$ and $h = CA^TDMq + p \in R^{m-1}$, then

$$L(\alpha) = \frac{1}{2}\alpha^T A^T DA\alpha - h^T\alpha - \frac{1}{2}C^2 q^T M^T DMq + \frac{C}{2}q^T q.$$

Since $-\frac{1}{2}C^2 q^T M^T DMq + \frac{C}{2}q^T q$ is not related to α, the Wolfe dual form of the problem (4) is

$$\min_{\alpha} g(\alpha) = \frac{1}{2}\alpha^T A^T DA\alpha + h^T\alpha.$$

Let $\frac{dg(\alpha)}{d\alpha} = 0$, then we can deduce that $A^T DA\alpha + h = 0$ and then $\alpha = (A^T DA)^{-1}h$. Consequently,

$$\beta = D(CMq + A\alpha) = D(CMq - A(A^T DA)^{-1}h) = DCMq + DA(A^T DA)^{-1}h,$$

$$b = p_0 - (K)_1^T \beta.$$

4. Numerical Examples

In order to demonstrate the effectiveness of the proposed method, in this section, we consider variable coefficient linear ordinary differential equations, that have analytic solutions. All computations are implemented in Matlab 2014b [9] on a PC with 2.5GHz CPU and 4GB memory. The RBF kernel function is chosen in experiment and we set the parameter C and *sigma*. In this paper, we choose four integral equations and their analytic solutions and listed them as follows equation (a)-(d). All numerical results are listed in Tables 1-4, and we describe the appreciate solution and exact solution vividly through Figure 1, where red line in Figure 1 (a)-(d) represent numerical solution while green line represent exact solution.

(a):

$$y''(x) - (1 - \cos x)y'(x) = e^x \cos x, \quad x \in [0, 1],$$

$$y(0) = 0,\ y'(0) = 1,$$

$$y(x) = e^x.$$

Table 1. Comparison for (a)

x	Exact solution	Numerical solution	Absolute error
0	1.0000	1.0000	0.0000
0.1	1.1052	1.1059	0.0007
0.2	1.2214	1.2242	0.0028
0.3	1.3499	1.3557	0.0059
0.4	1.4918	1.5016	0.0097
0.5	1.6487	1.6628	0.0141
0.6	1.8221	1.8407	0.0186
0.7	2.0138	2.0365	0.0227
0.8	2.2255	2.2511	0.0256
0.9	2.4596	2.4858	0.0262
1	2.7183	2.7413	0.0230

(b):

$$y''(x) - \tan(x)y'(x) - \frac{1}{\cos^2 x}\, y(x) = 0, \quad x \in [0, 1],$$

$$y(0) = 0,\ y'(0) = 1,$$

$$y(x) = \tan(x).$$

Table 2. Comparison for (b)

x	Exact solution	Numerical solution	Absolute error
0	0.0000	0.0000	0.0000
0.1	0.0998	0.0987	0.0011
0.2	0.1987	0.1950	0.0037
0.3	0.2955	0.2886	0.0069
0.4	0.3894	0.3797	0.0097
0.5	0.4794	0.4682	0.0112
0.6	0.5646	0.5541	0.0106
0.7	0.6442	0.6374	0.0069
0.8	0.7174	0.7180	0.0006
0.9	0.7833	0.7959	0.0126
1	0.8415	0.8712	0.02698

(c):

$$y''(x) - \sin(x)y'(x) + \cos(x)y(x) = -\sin(x), \quad x \in [0, 1],$$

$$y(0) = 0,\ y'(0) = 1,$$

$$y(x) = \sin(x).$$

Table 3. Comparison for (c)

x	Exact solution	Numerical solution	Absolute error
0	0.0000	0.0000	0.0000
0.1	0.0998	0.0990	0.0009
0.2	0.1987	0.1957	0.0030
0.3	0.2955	0.2898	0.0057
0.4	0.3894	0.3810	0.0084
0.5	0.4794	0.4690	0.0104
0.6	0.5646	0.5535	0.0111
0.7	0.6442	0.6342	0.0100
0.8	0.7174	0.7108	0.0065
0.9	0.7833	0.7832	0.0002
1	0.8415	0.8510	0.0095

(d):

$$y'' + \frac{1}{x}y' + y = 5 + 9x + x^2 + x^3, \quad x \in (0, 1],$$

$$y(0) = 0,\ y'(0) = 0,$$

$$y(x) = x^2 + x^3.$$

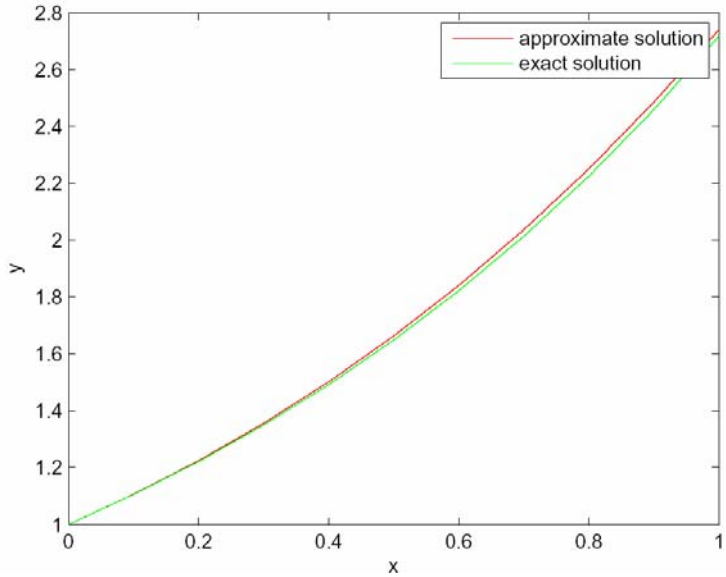
Table 4. Comparison for (d)

x	Exact solution	Numerical solution	Absolute error
0	0.0000	0.0000	0.0000
0.1	0.0110	0.0182	0.0072
0.2	0.0480	0.0729	0.0249
0.3	0.1170	0.1637	0.0467
0.4	0.2240	0.2905	0.0665
0.5	0.3750	0.4529	0.0779
0.6	0.5760	0.6503	0.0491
0.7	0.8330	0.8821	0.0743
0.8	1.1520	1.1477	0.0043
0.9	1.5390	1.4462	0.0928
1	2.0000	1.7766	0.2234

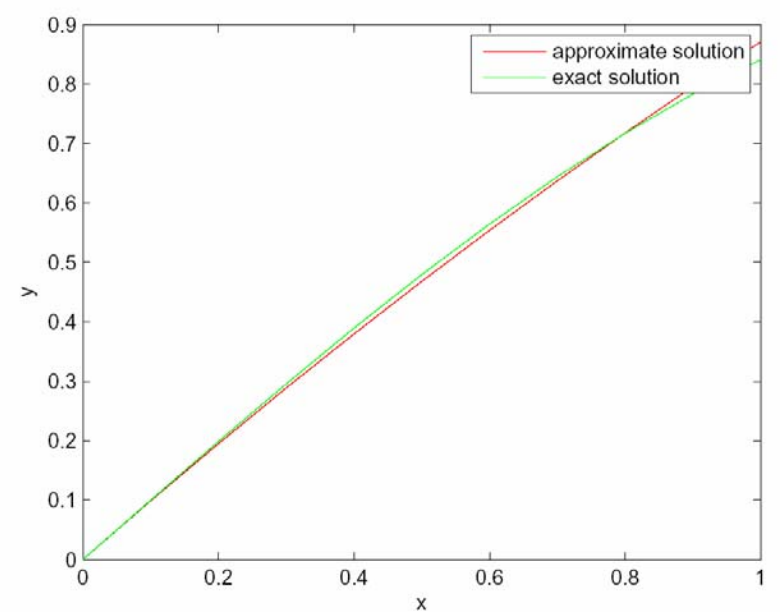
5. Conclusion

As we mentioned above and our excellent result of experiment, it can be seen that the largest absolute errors are not beyond 10^{-4} magnitude order. What is more, from the Figure 1, we can know that the red line and the green line curve almost coincide completely in Figure 1(a), Figure 1(b), and Figure 1(c), it meaning that the exact solution and numerical solution curve almost coincide completely, which show that the proposed algorithm could reach a quite agreeable accuracy. Therefore, we can see that we have proposed an approximal method for the solution of variable coefficient linear ordinary differential equations based on LS-SVR algorithm. This new algorithm can get the solution of Equation (1) by

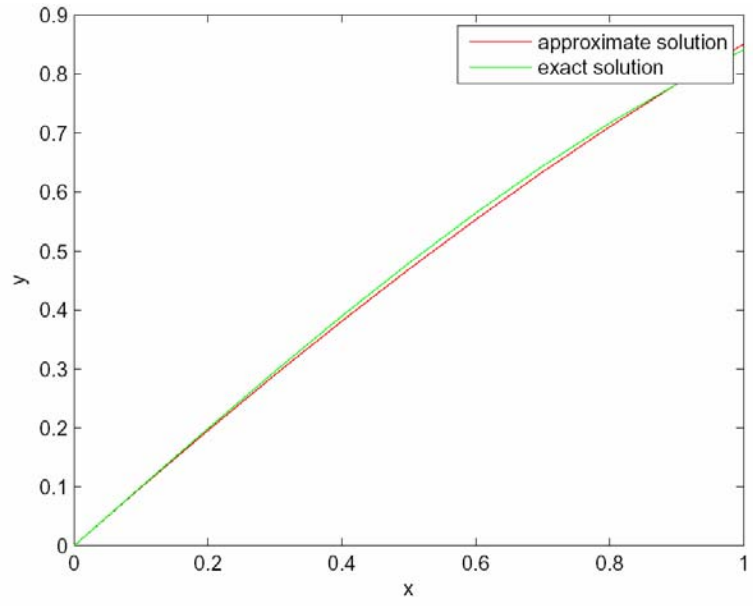
solving a convex quadratic programming, which has a powerful regression ability. Then verified the solving methods that we have given by examples. Experimental results show that our LS-SVR method really reached a high accuracy.



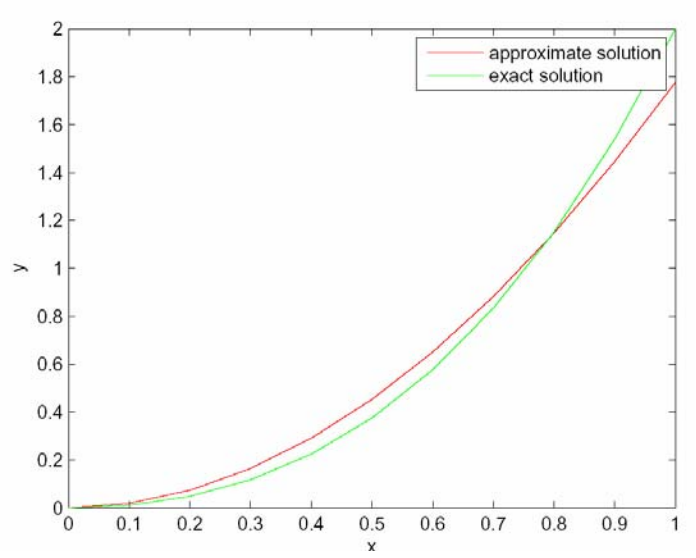
(a) Comparison for equation (a).



(b) Comparison for equation (b).



(c) Comparison for equation (c).



(d) Comparison for equation (d).

Figure 1. Diagrams of comparisons for four examples.

References

[1] M. Braun, Differential Equations and their Applications, Fourth Edition, Springer-Verlag, New York, 1993.

[2] V. N. Vapnik, The Nature of Statistical Learning Theory (2nd Edition), Springer, New York, 2000.

[3] S. Chen and M. Wang, Seeking multi-thresholds directly from support vectors for image segmentation, Neurocomputing 67(1-4) (2005), 335-344.

[4] J. A. K. Suykens, J. Vandewalle and B. De Moor, Optimal control by least squares support vector machines, Neural Networks 14(1) (2001), 23-25.

[5] K. W. Lau and Q. H. Wu, Local prediction of non-linear time series using support vector regression, Pattern Recognition 41(5) (2008), 1539-1547.

[6] M. Arun Kumar and M. Gopal, Least squares twin support vector machines for pattern classification, 36 (2009), 7535-7543.

[7] W. J. Xia and L. Y. Fan, Least squares support matrix machines based on bilevel programming, International Journal of Applied Mathematics and Machine Learning 4(1) (2016), 1-18.

[8] Xian Wang, Ao Li, Zhaohui Jiang and Huanqing Feng, Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme in BMC Bioinformatics 22 (2006),1-10.

[9] P. L. Davies and A. Kovac, Relevant literature such as non-parametric regression and density estimation under control of modality, (2000), 259-264.

[10] Jin-Guan Lin and Qing-Yun Zhuang, Chao Huang in Computational Economics Fuzzy Statistical Analysis of Multiple Regression with Crisp and Fuzzy Covariates and Applications in Analyzing Economic Data of China 39 (2012), 29-49.

[11] ECHI Nadhem, Approximate solution of linear differential equations, 58 (2013), 1502-1509.

[12] M. Kiehl and C. Zenger, An improved starting step of the G-B-S-method for the solution of ordinary differential equations, 41 (1989), 131-136.

[13] Per Grove Thomsen and Zahari Zlatev, Two-parameter families of predictor-corrector methods for the solution of ordinary differential equations, 19 (1979), 503-517.

[14] M. Arun Kumar and M. Gopal, Least squares twin support machines for pattern classification, 36 (2009), 7535-7543.

[15] Jin-Tsong Jeng, Chen-Chia Chuang and Shun-Feng Su, Support vector interval regression networks for interval regression analysis, 138 (2003), 283-300.

■