Expert Systems with Applications 37 (2010) 8784–8792

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

On Lagrangian support vector regression

S. Balasundaram [*], Kapil

School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India

ARTICLE INFO

Keywords:
Lagrangian support vector machines
Support vector regression
Time series

ABSTRACT

Prediction by regression is an important method of solution for forecasting. In this paper an iterative Lagrangian support vector machine algorithm for regression problems has been proposed. The method has the advantage that its solution is obtained by taking the inverse of a matrix of order equals to the number of input samples at the beginning of the iteration rather than solving a quadratic optimization problem. The algorithm converges from any starting point and does not need any optimization packages. Numerical experiments have been performed on Bodyfat and a number of important time series datasets of interest. The results obtained are in close agreement with the exact solution of the problems considered clearly demonstrates the effectiveness of the proposed method.

1. Introduction

Support vector machine (SVM) methods based on statistical learning theory (Vapnik, 2000) have been successfully applied to many problems of practical importance (Guyon, Weston, Barnhill, & Vapnik, 2002; Osuna, Freund, & Girosi, 1997) due to its high generalization performance over other learning methods. It is well known that the standard SVM formulation (Burges, 1998; Cristianini & Shawe-Taylor, 2000) leads to the solution of a quadratic programming problem with linear inequality constraints and that the problem will have an unique solution. With the combined advantages of generalization performance and unique solution SVM becomes an attractive method on problems of interest.

The goal of regression problem is in determining the underlying mathematical relationship between the given input observations and their output values. Regression models have been successfully applied in many important fields of study such as economics, engineering and bioinformatics. By the introduction of ε-insensitive error loss function proposed by Vapnik (2000) SVM methods have been successfully applied to regression problems (Mukherjee, Osuna, & Girosi, 1997; Muller, Smola, Ratsch, Schölkopf, & Kohlmorgen, 1999; Tay & Cao, 2001).

Considering the 2-norm error loss function instead of the usual 1-norm and maximizing the margin with respect to both the orientation and the relative location to the origin of the bounding planes, Mangasarian and Musicant (2001a, 2001b) studied "equivalent" SVM formulations for classification problems. This formulation leads to solving a positive-definite dual problem having only the non-negative constraints of the dual variables. Further their work on the study of formulating the problem of machine learning and data mining as an unconstrained minimization problem whose objective function is strongly convex and obtaining its solution using finite Newton method (see Fung & Mangasarian, 2003; Mangasarian, 2002). Since the objective function is not twice differentiable in this formulation by applying a smoothing technique a new SVM formulation called smooth SVM (SSVM) has been proposed in Lee and Mangasarian (2001). For the study on an extension of SSVM to ε-insensitive error loss based support vector regression (SVR) problems (see Lee, Hsieh, & Huang, 2005). Finally for the extension of the Active set SVM (ASVM) (Mangasarian & Musicant, 2001b) method proposed for classification problems to SVR problems we refer the reader to Musicant and Feinberg (2004).

Motivated by the study of Lagrangian SVM (Mangasarian & Musicant, 2001a) for classification problems we propose in this paper Lagrangian ε-insensitive SVR formulation. The main advantage of our approach in comparison with the standard SVR formulation is that the solution of the problem is obtained by taking the inverse of a matrix at the beginning of the iteration rather than solving a quadratic programming problem. In order to verify the effectiveness of the proposed method a number of problems of practical importance are considered. It is observed that the results obtained are in close agreement with the exact solution of the problems considered.

Throughout in this work all the vectors are assumed as column vectors. For any two vectors x, y in the n-dimensional real space R^n the inner product of the vectors will be denoted by $x^t y$ where x^t is the transpose of the vector x. When x is orthogonal to y we write $x \perp y$. The 2-norm of a vector x and a matrix Q will be denoted by $\|x\|$ and $\|Q\|$ respectively. For any vector $x \in R^n$, x_+ is a vector in R^n obtained by setting all the negative components of x to zero. For matrices $M \in R^{m\times n}$ and $N \in R^{n\times \ell}$, the kernel matrix K of size

[*] Corresponding author. Tel.: +91 11 26704724; fax: +91 11 26741586.
E-mail addresses: balajnu@hotmail.com, balajnu@gmail.com (S. Balasundaram).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.028

$m \times \ell$ is denoted by $K = K(M,N)$. The identity matrix of appropriate size is denoted by I and the column vector of ones of dimension m by e.

The paper is organized as follows. In Section 2 the linear and nonlinear SVR formulations for the standard 1-norm and 2-norm are introduced. By considering the Karush–Kuhn–Tucker (KKT) conditions the Lagrangian SVR algorithm is formulated in Section 3 and its convergence follows from the result of Mangasarian and Musicant (2001a). Numerical experiments have been performed on Bodyfat, Mackey–Glass, IBM, Google, Citigroup datasets and their results are compared with the exact solutions in Section 4. Finally we conclude our work in Section 5.

2. Support vector regression formulation

In this section, we briefly describe the standard 1-norm and 2-norm SVR formulations. For the given set of input samples $\{(x_i,y_i)\}_{i=1,2,\ldots,m}$ where $x_i \in R^n$ and $y_i \in R$, the linear SVR problem is the method in approximating the output by a function $f(\cdot)$ of the form:

$$f(x) = x^t w + b, \tag{1}$$

where $w \in R^n$ and $b \in R$ are determined as the solution of the following constrained, quadratic optimization problem with parameters $\nu > 0$ and $\varepsilon > 0$, written in matrix form as:

$$\min_{(w,b,\xi,\xi^*)\in R^{n+1+m+m}} \frac{1}{2} w^t w + \nu(e^t \xi + e^t \xi^*)$$

subject to

$$\begin{aligned} &y - Aw - be \leqslant \varepsilon e + \xi, \\ &Aw + be - y \leqslant \varepsilon e + \xi^* \end{aligned}$$

and

$$\xi_i, \xi_i^* \geqslant 0 \quad \text{for } i = 1,2,\ldots,m, \tag{2}$$

where $\xi = (\xi_1,\ldots,\xi_m)^t, \xi^* = (\xi_1^*,\ldots,\xi_m^*)^t$ are vectors of slack variables, $y = (y_1,\ldots,y_m)^t$ is the vector of observed values and $A \in R^{m\times n}$ be the matrix whose ith row, denoted by A_i, is defined to be the vector x_i^t. By introducing Lagrange multipliers u_1, $u_2 \in R^m$ the dual of the above problem (2) can be formulated as:

$$\min_{u_1,u_2\in R^m} \frac{1}{2}(u_1 - u_2)^t AA^t (u_1 - u_2) - y^t(u_1 - u_2) + \varepsilon e^t(u_1 + u_2)$$

subject to

$$e^t(u_1 - u_2) = 0 \text{ and } 0 \leqslant u_1, u_2 \leqslant \nu. \tag{3}$$

Further, for any example $x \in R^n$ its prediction is given by the following function

$$f(x) = x^t A^t (u_1 - u_2) + b.$$

For the nonlinear case, the input data is mapped into a higher dimensional feature space using a kernel function $k(\cdot,\cdot)$ and here in the feature space a linear SVR estimation is obtained. In this case, the support vector regression method will lead to the solution of the following Lagrangian problem:

$$\min_{(u_1,u_2)\in R^{m+m}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (u_{1i} - u_{2i}) k(x_i,x_j)(u_{1j} - u_{2j}) - \sum_{i=1}^{m} y_i(u_{1i} - u_{2i}) + \varepsilon \sum_{i=1}^{m} (u_{1i} + u_{2i})$$

subject to

$$e^t(u_1 - u_2) = 0 \text{ and } 0 \leqslant u_1, u_2 \leqslant \nu, \tag{4}$$

where $k(\cdot,\cdot)$ is the kernel function and $u_1 = (u_{11},\ldots,u_{1m})^t$, $u_2 = (u_{21},\ldots,u_{2m})^t$ in R^m are the Lagrange multipliers. For example, for the Gaussian kernel

$$k(x_i,x_j) = \exp(-\mu\|x_i - x_j\|^2) \text{ for } i,j = 1,\ldots,m$$

and μ is a parameter.

Using the solution of the problem (4), the regression estimation function $f(\cdot)$ for the nonlinear case is obtained to be: For any input $x \in R^n$,

$$f(x) = \sum_{i=1}^{m} (u_{1i} - u_{2i}) k(x,x_i) + b.$$

Following the approach of Mangasarian and Musicant (2001a, 2001b), Musicant and Feinberg (2004), by considering the square of the 2-norm of the slack variables ξ, ξ^* instead of 1-norm and adding the term $\left(\frac{b^2}{2}\right)$ in the definition of the objective function of (2) consider the linear SVR formulation with ε-insensitive error loss function (Vapnik, 2000) as a constrained minimization problem of the following form (Musicant & Feinberg, 2004):

$$\min_{(w,b,\xi,\xi^*)\in R^{n+1+m+m}} \frac{1}{2}(w^t w + b^2) + \frac{\nu}{2} \sum_{i=1}^{m} (\xi_i^2 + \xi_i^{*2})$$

subject to

$$\begin{aligned} &(y_i - A_i w - b) \leqslant (\varepsilon + \xi_i), \\ &(A_i w + b - y_i) \leqslant (\varepsilon + \xi_i^*) \quad i = 1,2,\ldots,m, \end{aligned} \tag{5}$$

where ξ_i, ξ_i^* are slack variables and ε, ν are input parameters. Since none of the components of the vector $\xi = (\xi_1,\ldots,\xi_m)^t$ or $\xi^* = (\xi_1^*,\ldots,\xi_m^*)^t$ will be negative at optimality (Musicant & Feinberg, 2004) their non-negativity constraints have been dropped in the formulation (5). Since the linear regression estimation function takes the form (1) its approximation to the vector $y \in R^m$ of observed values will become

$$y \approx Aw + be,$$

where w and b be the solution of (5).

By introducing the Lagrange multipliers $u_1 = (u_{11},\ldots,u_{1m})^t$ and $u_2 = (u_{21},\ldots,u_{2m})^t$ in R^m the Lagrangian function L can be obtained to be:

$$\begin{aligned} L(w,b,\xi,\xi^*,u_1,u_2) = {} & \frac{1}{2}(w^t w + b^2) + \frac{\nu}{2} \sum_{i=1}^{m} (\xi_i^2 + \xi_i^{*2}) \\ & + \sum_{i=1}^{m} u_{1i}(y_i - A_i w - b - \varepsilon - \xi_i) \\ & + \sum_{i=1}^{m} u_{2i}(A_i w + b - y_i - \varepsilon - \xi_i^*). \end{aligned}$$

Using the condition that the partial derivatives of L with respect to the primal variables will be zero at optimality the dual problem can be written as a minimization problem of the following form (Musicant & Feinberg, 2004)

$$\begin{aligned} \min_{0\leqslant u_1,u_2\in R^m} & \frac{1}{2}\left[(u_1 - u_2)^t (AA^t + ee^t)(u_1 - u_2)\right] + \frac{1}{2\nu}(u_1^t u_1 + u_2^t u_2) \\ & - y^t(u_1 - u_2) + \varepsilon e^t(u_1 + u_2), \end{aligned} \tag{6}$$

in which

$$w = A^t(u_1 - u_2) \text{ and } b = e^t(u_1 - u_2) \tag{7}$$

hold. Now from (1) and (7) the linear regression estimation function $f(\cdot)$ is given by

$$f(x) = [x^t \quad 1]\begin{bmatrix} A^t \\ e^t \end{bmatrix}(u_1 - u_2). \tag{8}$$

Define $G = [A \ e]$ an augmented matrix. Then the dual problem (6) can be written in the following form:

$$\min_{0\leqslant u_1,u_2\in R^m} \frac{1}{2}[u_1^t \quad u_2^t]\,Q\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - r^t\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{9}$$

where

$$Q = \begin{bmatrix} \frac{I}{v} + GG^t & -GG^t \\ -GG^t & \frac{I}{v} + GG^t \end{bmatrix} \tag{10}$$

and

$$r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} y - \varepsilon e \\ -y - \varepsilon e \end{bmatrix} \tag{11}$$

are block matrices.

Following the approach of Mangasarian and Musicant (2001a), Musicant and Feinberg (2004), the linear SVR formulation (9) defined in dual variables can be extended to nonlinear SVR model.

For $A \in R^{m\times n}$, consider the kernel matrix K of order m defined by

$$K = K(A, A^t)$$

whose (i,j)th element is given by

$$K(A, A^t)_{ij} = k(x_i, x_j) \in R,$$

where $k(\cdot,\cdot)$ is a nonlinear kernel function. Also for a given vector $x \in R^n$, we define

$$K(x^t, A^t) = (k(x, x_1), \ldots, k(x, x_m)),$$

a row vector in R^m.

Replacing GG^t by the positive semi-definite symmetric kernel matrix $K = K(G, G^t)$, the nonlinear SVR problem in dual variables can be formulated in the form of (9) where the matrix Q will become

$$Q = \begin{bmatrix} \frac{I}{v} + K(G, G^t) & -K(G, G^t) \\ -K(G, G^t) & \frac{I}{v} + K(G, G^t) \end{bmatrix} \tag{12}$$

Following a similar approach to classification kernels (Mangasarian & Musicant, 2001a), for any vector $x \in R^n$ the kernel regression estimation function $f(\cdot)$ is obtained to be of the form

$$f(x) = K([x^t \ 1], G^t)(u_1 - u_2). \tag{13}$$

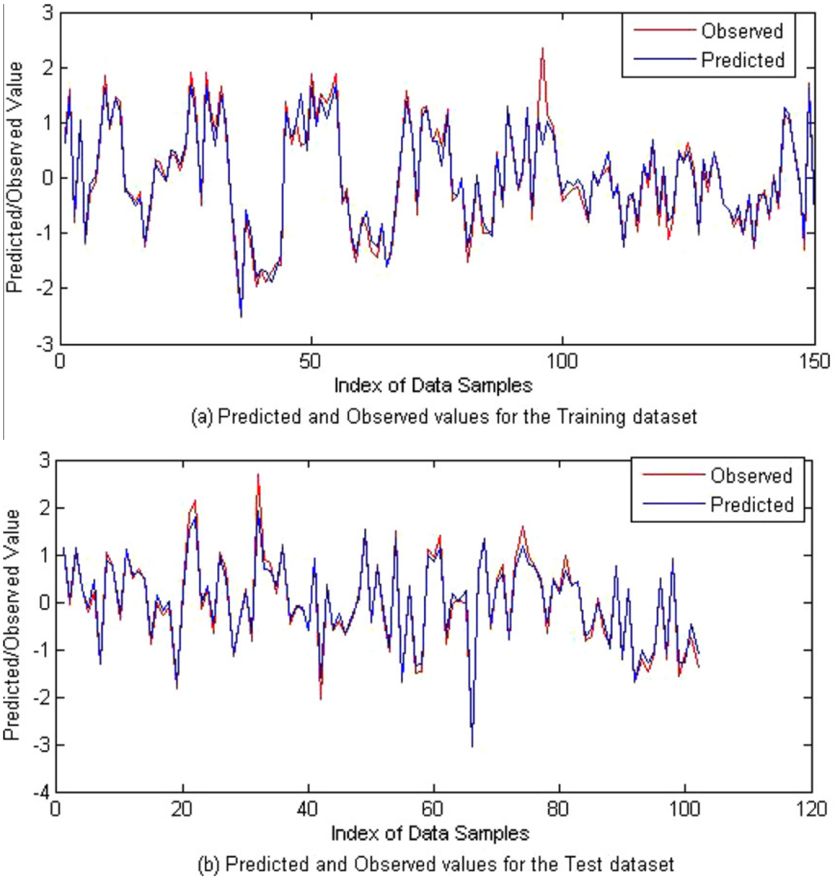
Throughout in this work we assume that the kernel matrix $K(G, G^t)$ is symmetric and positive semi-definite.

3. Lagrangian support vector regression algorithm

It was shown in the previous section that the dual problem for either the linear or nonlinear case can be formulated as

$$\min_{0\leqslant u\in R^{2m}} \frac{1}{2}u^tQu - r^tu, \tag{14}$$

where $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ is a vector in R^{2m}. In this section, following the procedure of Mangasarian and Musicant (2001a), we discuss the method of solution for the Lagrangian SVR formulation (14). The KKT necessary and sufficient optimality conditions for the dual problem (14) will become solving the classical nonlinear complementarity problem (Mangasarian, 1994)



(a) Predicted and Observed values for the Training dataset

(b) Predicted and Observed values for the Test dataset

Fig. 1. Accuracy plots for the Bodyfat dataset with $v = 100$ & $\mu = 2^{-10}$ using the number of training and test samples equal to 150 and 102 respectively.

$$0 \leqslant u \perp (Qu - r) \geqslant 0. \tag{15}$$

However, the optimality condition (15) is satisfied if and only if for any $\alpha > 0$ the relation

$$(Qu - r) = ((Qu - r) - \alpha u)_+$$

holds.

Now for obtaining the solution of the above problem we apply the following simple iterative scheme

$$u^{i+1} = Q^{-1}(r + ((Qu^i - r) - \alpha u^i)_+). \tag{16}$$

Keeping in view of a general discussion of the algorithm and its convergence, we define

$$H = \begin{cases} GG^t & \text{for the linear case} \\ K(G, G^t) & \text{for the nonlinear case} \end{cases}$$

Then, the matrix Q is written as a block matrix of the form

$$Q = \begin{bmatrix} \frac{I}{\nu} + H & -H \\ -H & \frac{I}{\nu} + H \end{bmatrix}. \tag{17}$$

Lemma 1

(i) *$Q \in R^{2m \times 2m}$ is a symmetric and positive-definite matrix;*
(ii) *For any real number $\beta > 0$,*

$$\left(\frac{I}{\nu} + \beta H\right)^{-1} H = H\left(\frac{I}{\nu} + \beta H\right)^{-1}.$$

Proof

(i) Since H is symmetric and positive semi-definite, the matrix

$$\begin{bmatrix} H & -H \\ -H & H \end{bmatrix}$$

is also symmetric and positive semi-definite. Clearly, by definition (17), Q will become symmetric and positive-definite.

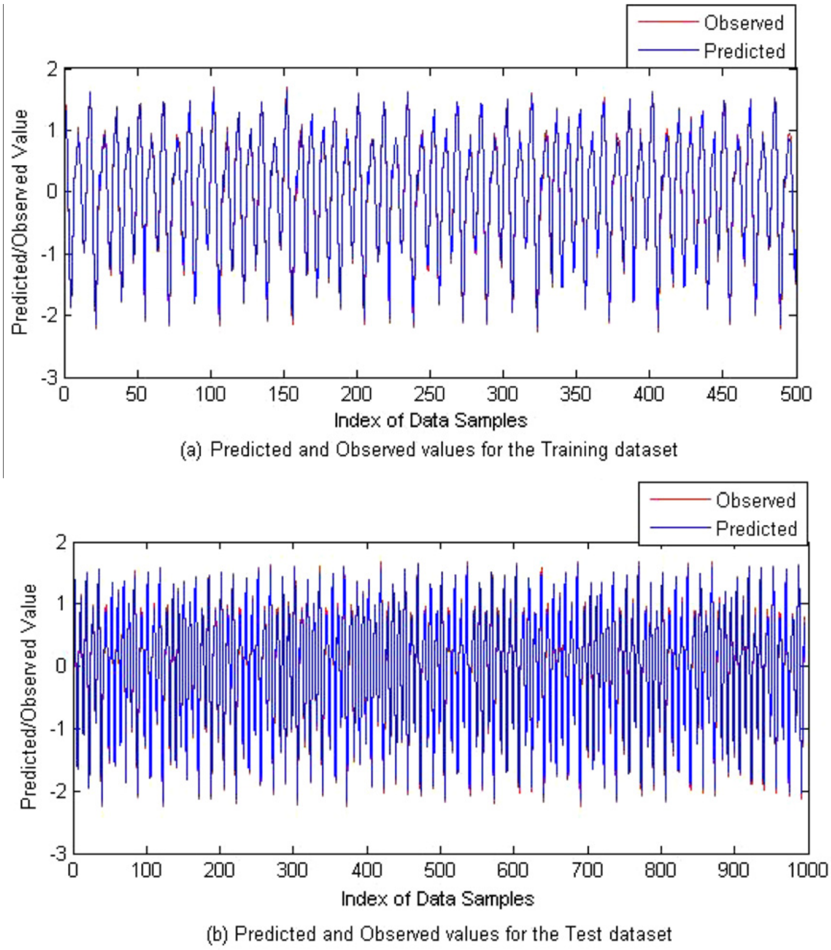
(ii)

$$\begin{aligned} \left(\frac{I}{\nu} + \beta H\right)^{-1} H &= \left(\frac{I}{\nu} + \beta H\right)^{-1} H\left(\frac{I}{\nu} + \beta H\right)\left(\frac{I}{\nu} + \beta H\right)^{-1} \\ &= \left(\frac{I}{\nu} + \beta H\right)^{-1}\left(\frac{I}{\nu} + \beta H\right) H\left(\frac{I}{\nu} + \beta H\right)^{-1} \\ &= H\left(\frac{I}{\nu} + \beta H\right)^{-1}. \qquad \square \end{aligned}$$

Lemma 2 (Press, Teukolsky, Vetterling, and Flannery, 1994). *Suppose an invertible matrix $\bar{A}$ of the following form*

$$\bar{A} = \begin{bmatrix} \overline{P} & \overline{Q} \\ \overline{R} & \overline{S} \end{bmatrix}$$

is given where $\overline{P}, \overline{Q}, \overline{R}$ and $\overline{S}$ are block matrices of size $p \times p$, $p \times s$, $s \times p$ and $s \times s$ respectively. Assume that $\bar{A}^{-1}$ can be partitioned in a similar manner



(a) Predicted and Observed values for the Training dataset

(b) Predicted and Observed values for the Test dataset

Fig. 2. Accuracy plots for the MG_{17} dataset with $\nu = 1000$ & $\mu = 2^{-3}$ using the number of training and test samples equal to 500 and 995 respectively.

$$\overline{A}^{-1} = \begin{bmatrix} \widetilde{P} & \widetilde{Q} \\ \widetilde{R} & \widetilde{S} \end{bmatrix}$$

where $\widetilde{P}, \widetilde{Q}, \widetilde{R}$ *and* $\widetilde{S}$ *are matrices of the same size as* $\overline{P}, \overline{Q}, \overline{R}$ *and* $\overline{S}$ *respectively. Then,*

$$\widetilde{S} = (\overline{S} - \overline{R}\overline{P}^{-1}\overline{Q})^{-1},$$

$$\widetilde{P} = \overline{P}^{-1} + \overline{P}^{-1}\overline{Q}\widetilde{S}\overline{R}\overline{P}^{-1},$$

$$\widetilde{Q} = -\overline{P}^{-1}\overline{Q}\widetilde{S}$$

and

$$\widetilde{R} = -\widetilde{S}\overline{R}\overline{P}^{-1}.$$

In the following theorem, we compute the inverse of the matrix Q using the previous Lemma.

Theorem 1. *For the matrix Q defined by (17), let*

$$Q^{-1} = \begin{bmatrix} \widetilde{P} & \widetilde{Q} \\ \widetilde{R} & \widetilde{S} \end{bmatrix}$$

Then,

$$\widetilde{P} = (I + vH)\left(\frac{I}{v} + 2H\right)^{-1},$$

$$\widetilde{Q} = vH\left(\frac{I}{v} + 2H\right)^{-1},$$

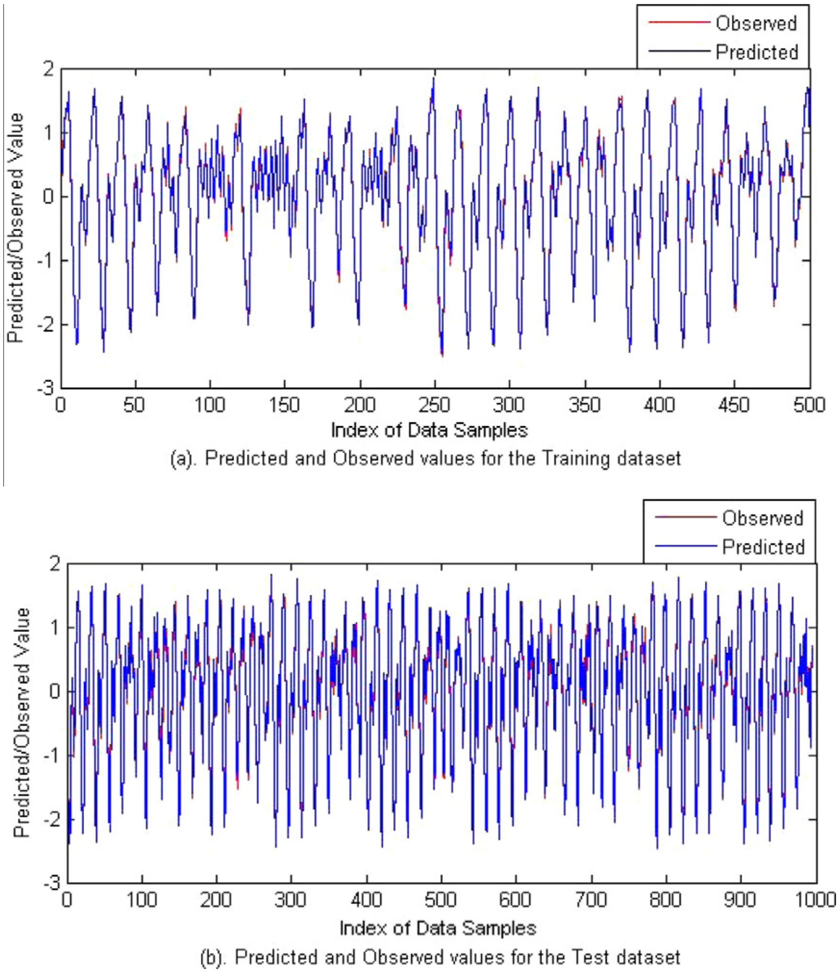
$$\widetilde{R} = v\left(\frac{I}{v} + 2H\right)^{-1} H$$

and

$$\widetilde{S} = v\left(\frac{I}{v} + 2H\right)^{-1}\left(\frac{I}{v} + H\right) = \left(\frac{I}{v} + 2H\right)^{-1}(I + vH).$$

Proof. First we obtain the result for $\widetilde{S}$. Consider

$$\begin{aligned} &\left[\left(\frac{I}{v} + H\right) - H\left(\frac{I}{v} + H\right)^{-1} H\right] \\ &\quad = \left(\frac{I}{v} + H\right) - \left(\frac{I}{v} + H\right)^{-1}(H)(H) \quad \text{(by Lemma 1(ii))} \\ &\quad = \left(\frac{I}{v} + H\right)^{-1}\left[\left(\frac{I}{v} + H\right)\left(\frac{I}{v} + H\right) - (H)(H)\right] \\ &\quad = \left(\frac{1}{v}\right)\left(\frac{I}{v} + H\right)^{-1}\left(\frac{I}{v} + 2H\right). \end{aligned}$$



(a). Predicted and Observed values for the Training dataset

(b). Predicted and Observed values for the Test dataset

Fig. 3. Accuracy plots for the MG_{30} dataset with $v = 1000$ & $\mu = 2^{-2}$ using the number of training and test samples equal to 500 and 995 respectively.

Then, by Lemma 2

$$\widetilde{S} = \left[\left(\frac{1}{v}\right)\left(\frac{I}{v}+H\right)^{-1}\left(\frac{I}{v}+2H\right)\right]^{-1} = v\left(\frac{I}{v}+2H\right)^{-1}\left(\frac{I}{v}+H\right). \qquad (18)$$

Now, using Lemmas 2 and 1(ii) and (18), we get

$$\begin{aligned}
\widetilde{P} &= \left(\frac{I}{v}+H\right)^{-1}\left[I+H\widetilde{S}H\left(\frac{I}{v}+H\right)^{-1}\right] \\
&= \left(\frac{I}{v}+H\right)^{-1}\left[I+H\widetilde{S}\left(\frac{I}{v}+H\right)^{-1}H\right] \\
&= \left(\frac{I}{v}+H\right)^{-1}\left[I+Hv\left(\frac{I}{v}+2H\right)^{-1}H\right] \\
&= \left(\frac{I}{v}+H\right)^{-1}\left[I+vH\left(\frac{I}{v}+2H\right)^{-1}H\right] \\
&= \left(\frac{I}{v}+H\right)^{-1}\left[I+vHH\left(\frac{I}{v}+2H\right)^{-1}\right] \\
&= \left(\frac{I}{v}+H\right)^{-1}\left[\left(\frac{I}{v}+H\right)+(H+vHH)\right]\left(\frac{I}{v}+2H\right)^{-1} \\
&= \left(\frac{I}{v}+H\right)^{-1}\left[\left(\frac{I}{v}+H\right)+v\left(\frac{I}{v}+H\right)H\right]\left(\frac{I}{v}+2H\right)^{-1} \\
&= (I+vH)(\frac{I}{v}+2H)^{-1}.
\end{aligned}$$

Also, we have

$$\widetilde{R} = \widetilde{S}H\left(\frac{I}{v}+H\right)^{-1} = \widetilde{S}\left(\frac{I}{v}+H\right)^{-1}H = v\left(\frac{I}{v}+2H\right)^{-1}H. \quad \text{(by (18))}$$

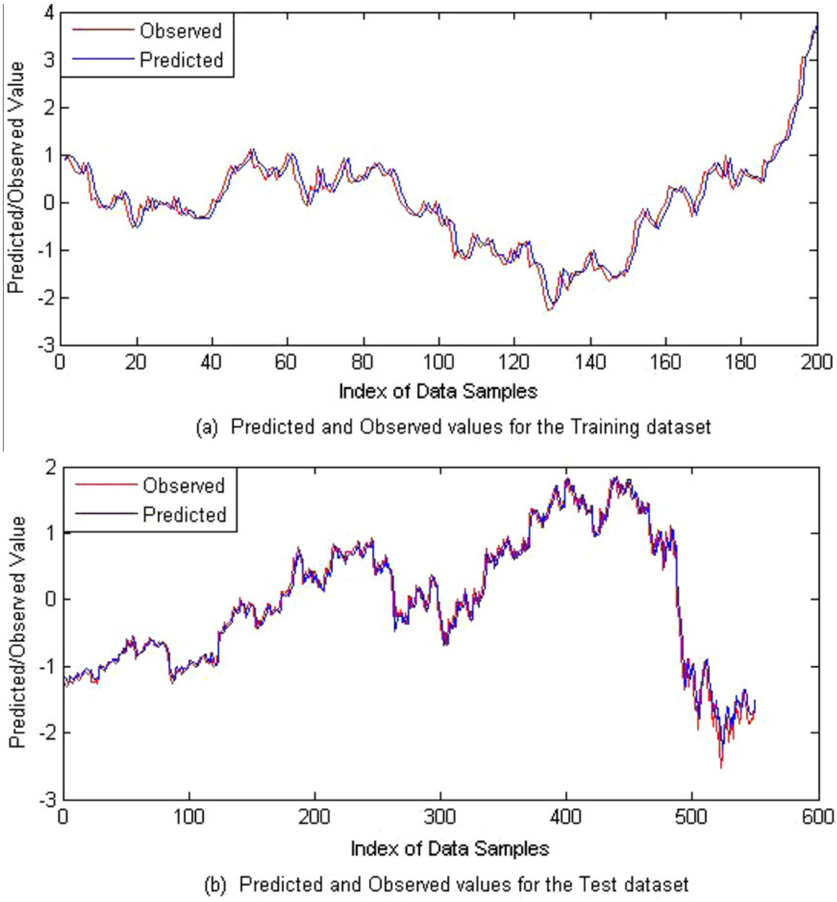
Finally, since $(\frac{I}{v}+2H)(\frac{I}{v}+H) = (\frac{I}{v}+H)(\frac{I}{v}+2H)$ we get

$$\begin{aligned}
\widetilde{Q} &= \left(\frac{I}{v}+H\right)^{-1}H\widetilde{S} = H\left(\frac{I}{v}+H\right)^{-1}\widetilde{S} \\
&= vH\left[\left(\frac{I}{v}+H\right)^{-1}\left(\frac{I}{v}+2H\right)\left(\frac{I}{v}+H\right)\right]^{-1} \quad \text{(by (18))} \\
&= vH\left(\frac{I}{v}+2H\right)^{-1} = v\left(\frac{I}{v}+2H\right)^{-1}H. \qquad \square
\end{aligned}$$

In the following theorem we establish that the iterative method given by (16) can be rewritten in another form with the advantage being that it is sufficient to invert a matrix of order *m* rather than inverting the matrix Q given by (17) of order *2m* at the beginning of the algorithm.

Theorem 2. *The iterative algorithm (16) can be formulated in the following form: For i = 0,1,2,...*

$$\begin{aligned}
u_1^{i+1} &= \widetilde{P}\left(r_1+\left(\left(\frac{I}{v}+H\right)u_1^i-\alpha u_1^i-Hu_2^i-r_1\right)_+\right) \\
&\quad+\widetilde{Q}\left(r_2+\left(\left(\frac{I}{v}+H\right)u_2^i-\alpha u_2^i-Hu_1^i-r_2\right)_+\right)
\end{aligned}$$



(a) Predicted and Observed values for the Training dataset

(b) Predicted and Observed values for the Test dataset

Fig. 4. Accuracy plots for the IBM dataset with $v = 1000$ & $\mu = 2^{-10}$ using the number of training and test samples equal to 200 and 550 respectively.

and

$$u_2^{i+1} = \widetilde{R}\left(r_1 + \left(\left(\frac{I}{\nu}+H\right)u_1^i - \alpha u_1^i - Hu_2^i - r_1\right)_+\right) + \widetilde{S}\left(r_2 + \left(\left(\frac{I}{\nu}+H\right)u_2^i - \alpha u_2^i - Hu_1^i - r_2\right)_+\right)$$

where $\widetilde{P}, \widetilde{Q}, \widetilde{R}$ *and* $\widetilde{S}$ *are matrices defined in* Theorem 1 *and the vector* r *is given by* (11).

Proof. Since

$$((Q-\alpha I)u^i - r)_+ = \begin{bmatrix} ((\frac{I}{\nu}+H)u_1^i - \alpha u_1^i - Hu_2^i - r_1)_+ \\ ((\frac{I}{\nu}+H)u_2^i - \alpha u_2^i - Hu_1^i - r_2)_+ \end{bmatrix}$$

by Theorem 1 the iterative scheme (16) becomes

$$\begin{bmatrix} u_1^{i+1} \\ u_2^{i+1} \end{bmatrix} = \begin{bmatrix} \widetilde{P} & \widetilde{Q} \\ \widetilde{R} & \widetilde{S} \end{bmatrix}\left(\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \begin{bmatrix} \left((\frac{I}{\nu}+H)u_1^i - \alpha u_1^i - Hu_2^i - r_1\right)_+ \\ \left((\frac{I}{\nu}+H)u_2^i - \alpha u_2^i - Hu_1^i - r_2\right)_+ \end{bmatrix}\right)$$

from which the result follows. □

By Theorems 1 and 2 we see that the iterative algorithm requires the inverse of the matrix $(\frac{I}{\nu}+2H)$ of order m. However for the linear case it can be computed using the Sherman–Morrison–Woodbury or SMW identity (Golub & Van Loan, 1996), i.e.

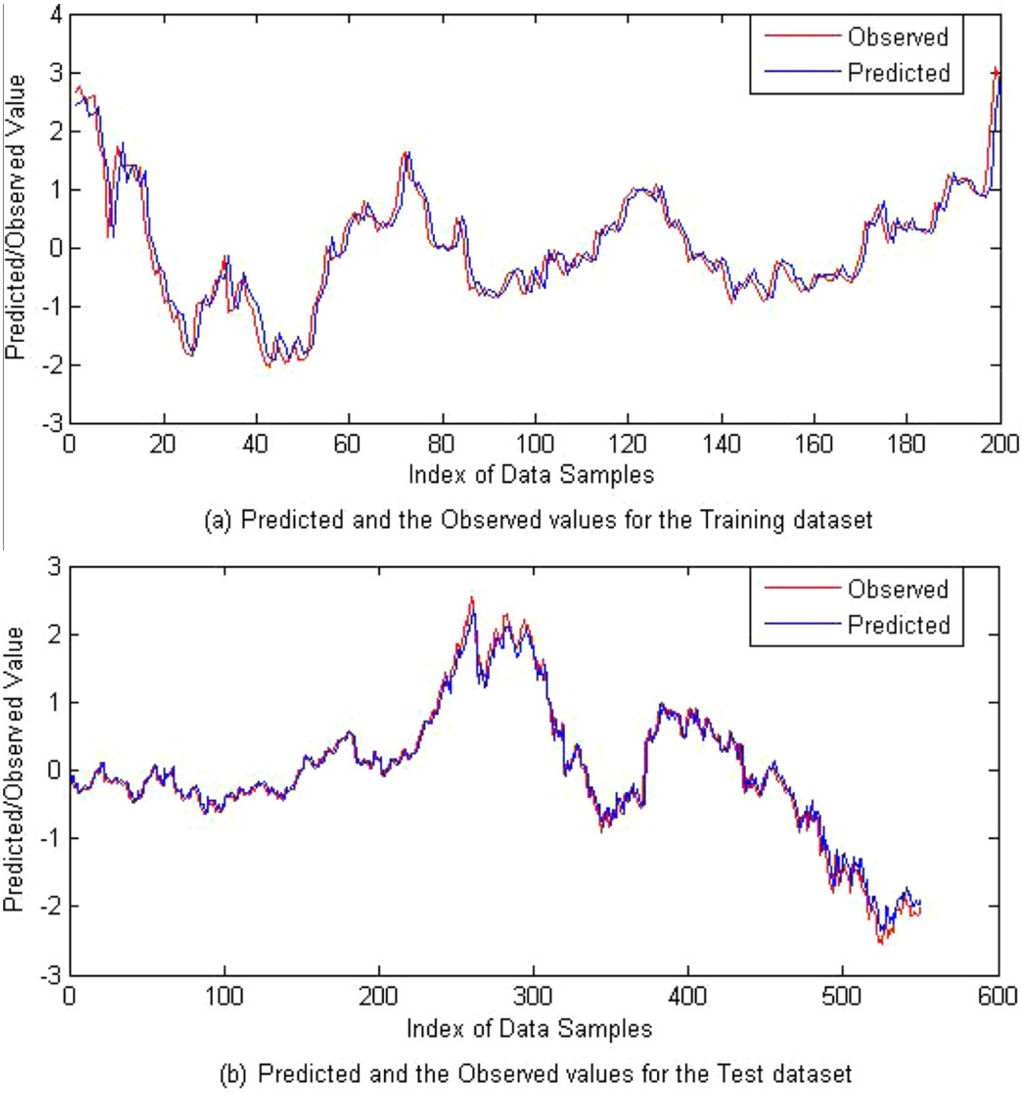
$$\left(\frac{I}{\nu}+2GG^t\right)^{-1} = \nu\left(I - 2G\left(\frac{I}{\nu}+2G^tG\right)^{-1}G^t\right).$$

When $n \ll m$ this form has the advantage that it is sufficient to compute the inverse of the matrix $(\frac{I}{\nu}+2G^tG)$ of order $(n+1)$ instead of a matrix of much larger size $m \times m$. Note that for the case of nonlinear regression the kernel matrix K will become a square matrix and thus SMW identity will be of no use.

Now we state the global linear convergence of the iterative scheme in the next theorem whose proof will follow from the result of Mangasarian and Musicant (2001a).

Theorem 3 (Mangasarian and Musicant, 2001a). *Assume that*

$$0 < \alpha < \frac{2}{\nu}.$$



(a) Predicted and the Observed values for the Training dataset

(b) Predicted and the Observed values for the Test dataset

Fig. 5. Accuracy plots for the Google dataset with $\nu = 1000$ & $\mu = 2^{-10}$ using the number of training and test samples equal to 200 and 550 respectively.

Then, for any starting point $u^0 \in R^{2m}$, the iterative scheme (16) *converges to the unique solution $\bar{u}$ at the linear rate satisfying the condition*

$$\|Qu^{i+1} - Q\bar{u}\| \leqslant \|I - \alpha Q^{-1}\| \quad \|Qu^i - Q\bar{u}\|.$$

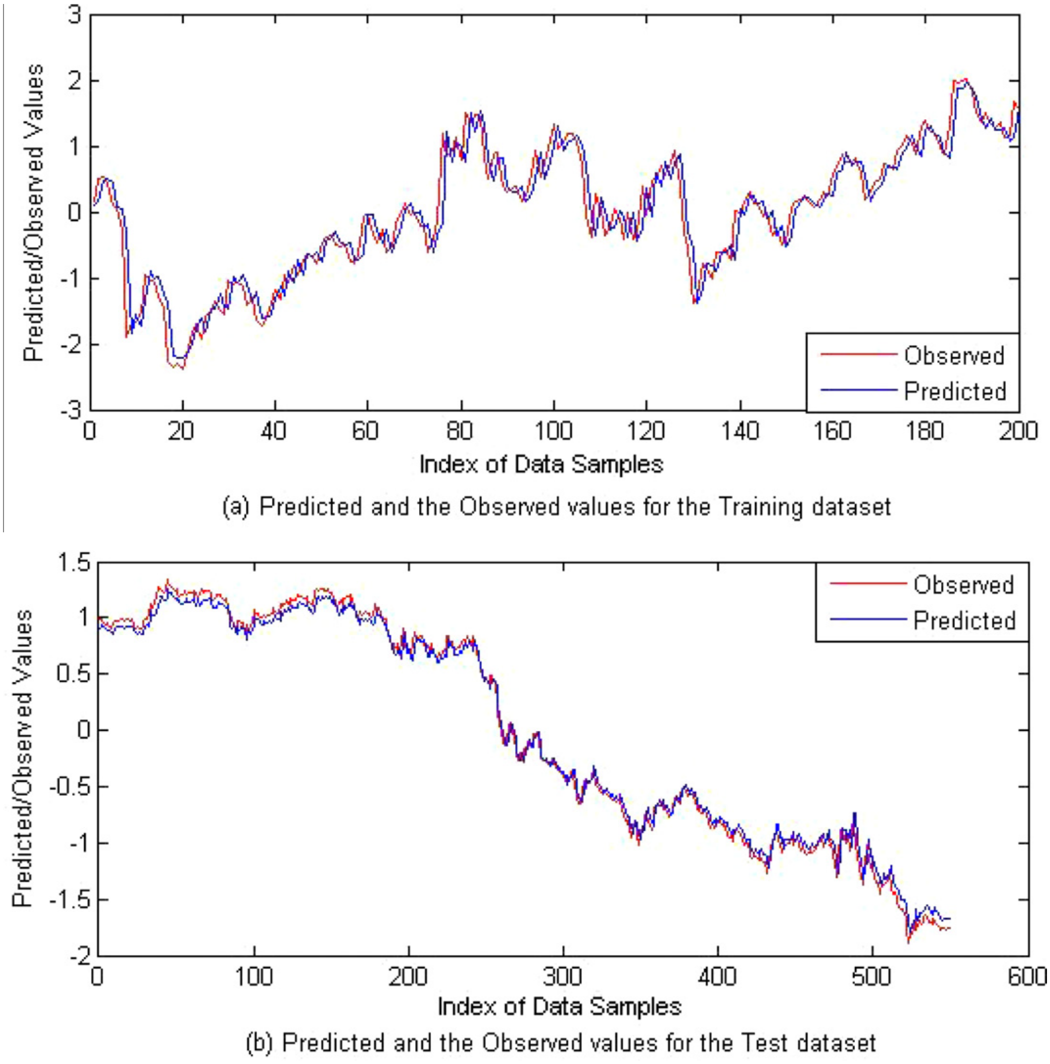
4. Experimental results

In order to demonstrate the effectiveness of the proposed Lagrangian SVR method experiments are performed on a number of interesting datasets namely Bodyfat, the time series generated by the Mackey Glass delay differential equation with different delays (Casdagli, 1989), IBM, Google and Citigroup. The Bodyfat dataset is obtained from Statlib collection: http://lib.stat.cmu.edu/datasets/ The IBM, Google and Citigroup are financial datasets of stock index taken from the Yahoo financial website: http://finance.yahoo.com. In all our experiments the original data is normalized with zero mean and standard deviation equals to one. The value of ε is taken to be 0.01. We applied the 2-norm relative error to evaluate the prediction performance. For an observed vector y and its corresponding prediction vector $\tilde{y}$ the relative error RE is calculated using the following formula

$$RE = \frac{\|y - \tilde{y}\|}{\|y\|}.$$

For all the examples considered Gaussian nonlinear kernel function is used. The optimal values for the regularization parameter ν and the kernel parameter μ are chosen using 10-fold cross validation. The best prediction performance on the training dataset was obtained by varying the regularization parameter $\nu = \{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$ and the kernel parameter $\mu = \{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$. In all the figures red color line shows the observed value and the predicted value is shown in blue color. For all the time series datasets considered the current output value is predicted using the five previous values.

Bodyfat is a dataset with the observed value being the estimation of the body fat from the body density values and is taken from Statlib collection. It consists of 252 data points having 14 number of attributes. The first 150 samples are taken for training and the remaining 102 samples for testing. Using 10-fold cross validation the optimal values obtained for the regularization parameter ν and the kernel parameter μ being 100 and 2^{-10} respectively. The observed and the predicted values for the training and the test datasets are shown in Fig. 1(a) and (b) respectively. For these optimal values of ν and μ the RE error is calculated on the test dataset and is obtained to be 0.1678.



(a) Predicted and the Observed values for the Training dataset

(b) Predicted and the Observed values for the Test dataset

Fig. 6. Accuracy plots for the CitiGroup dataset with $\nu = 1000$ & $\mu = 2^{-10}$ using the number of training and test samples equal to 200 and 550 respectively.

Table 1
The Relative Errors obtained using our algorithm with Gaussian Kernel on the test datasets.

Dataset	Training Set Size	Testing Set Size	$(\nu,\mu\ ,\varepsilon)$	Relative Error
Bodyfat (252 × 14)	(150 × 14)	(102 × 14)	$(100,2^{-10},0.01)$	0.1678
MG_{17} (1495 × 5)	(500 × 5)	(995 × 5)	$(1000,2^{-3},0.01)$	0.0583
MG_{30} (1495 × 5)	(500 × 5)	(995 × 5)	$(1000,2^{-2},0.01)$	0.1102
IBM (750 × 5)	(200 × 5)	(550 × 5)	$(1000,2^{-10},0.01)$	0.1412
Google (750 × 5)	(200 × 5)	(550 × 5)	$(1000,2^{-10},0.01)$	0.1412
Citigroup (750 × 5)	(200 × 5)	(550 × 5)	$(1000,2^{-10},0.01)$	0.0746

The Mackey–Glass time delay differential equation (Casdagli, 1989; Mukherjee et al., 1997) is given by

$$\frac{\partial x(t)}{\partial t} = -bx(t) + a\frac{x(t-\tau)}{1+x(t-\tau)^{10}},$$

where a, b are parameters and τ is the time delay. We perform our experiment on two time series generated by the above differential equation using the parameter values $a = 0.2$, $b = 0.1$ with $\tau = 17$ and $\tau = 30$ where τ is the time delay. We call the time series corresponding to $\tau = 17$ and $\tau = 30$ as MG_{17} and MG_{30} respectively. They are widely used as benchmark dataset values for analyzing the generalization ability of the method of prediction. These datasets are taken from http://www.cse.ogi.edu/~ericwan. The first 500 number of data values are considered for training and the remaining 995 data values for testing. By applying 10-fold cross validation on the training datasets the optimal values of ν and μ obtained for MG_{17} and MG_{30} are ($\nu = 10^3$, $\mu = 2^{-3}$) and ($\nu = 10^3$, $\mu = 2^{-2}$) respectively. Further the relative error RE on the test data for MG_{17} is calculated to be 0.0583 and similarly for MG_{30} it is 0.1102. The observed values and their prediction for the training and the test datasets for MG_{17} are shown in Fig. 2(a) and (b) respectively and similarly for MG_{30} they are shown in Fig. 3(a) and (b) respectively.

For the IBM, Google and Citigroup financial datasets, 750 closing prices starting from 01-01-2006 to 31-12-2008 are considered. In all the three examples, the initial 200 closing prices are assumed for training and the remaining 550 reserved for testing. By applying 10-fold cross validation on the training datasets the optimal value of (ν,μ) obtained for IBM, Google and Citigroup will be ($\nu = 10^3$, $\mu = 2^{-10}$) for all the three cases. With the selection of the optimal values for ν and μ, the RE is calculated on the test dataset. The observed values and their prediction for the training and the test datasets are shown respectively in Fig. 4(a) and (b) for IBM, in Fig. 5(a) and (b) for Google and in Fig. 6(a) and (b) for Citigroup. The relative error RE on the test data is computed to be 0.1412, 0.1412 and 0.0746 for IBM, Google and Citigroup respectively.

The number of training and test samples, the number of attributes, the input parameter values and relative errors obtained using our method for all datasets considered are summarized in Table 1. Our results clearly show that the proposed algorithm is a powerful method of solution for regression problems.

5. Conclusions

A new iterative Lagrangian support vector regression algorithm is proposed in this paper. The effectiveness of the proposed method is demonstrated by performing numerical experiments on a number of interesting datasets. The algorithm requires the inverse of a matrix of order equals to twice the number of input samples, i.e., the number of non-negativity constraints of the dual variables, at the beginning of the algorithm. However by considering this matrix as a block matrix the algorithm is reformulated so that the solution is obtained by taking the inverse of a matrix of order equals to the number of input samples. Future work will be on the study of the implicit Lagrangian formulation (Mangasarian & Solodov, 1993) for the above dual problem considered.

References

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*(2), 121–167.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica D, 35*, 335–356.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and kernel based learning methods*. Cambridge University Press.

Fung, G., & Mangasarian, O. L. (2003). Finite Newton method for Lagrangian support vector machine. *Neurocomputing, 55*, 39–55.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations* (3rd ed.). The Johns Hopkins University Press.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machine. *Machine Learning, 46*, 389–422.

Lee, Y. J., Hsieh, W.-F., & Huang, C.-M. (2005). ε-SSVR: A smooth support vector machine for ε-insensitive regression. *IEEE Transactions on Knowledge and Data Engineering, 17*(5), 678–685.

Lee, Y. J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine for classification. *Computational Optimization and Applications, 20*(1), 5–22.

Mangasarian, O. L. (1994). *Nonlinear programming*. PA: SIAM Philadelphia.

Mangasarian, O. L. (2002). A finite newton method for classification. *Optimization Methods and Software, 17*, 913–929.

Mangasarian, O. L., & Musicant, D. R. (2001a). Lagrangian support vector machines. *Journal of Machine Learning Research, 1*, 161–177.

Mangasarian, O. L., & Musicant, D. R. (2001b). Active set support vector machine classification. In T. K. Leen, T. G. Dietterich, & V. Tesp (Eds.). *Advances in neural information processing systems* (Vol. 13, pp. 577–586). MIT Press.

Mangasarian, O. L., & Solodov, M. V. (1993). Nonlinear complementarity as unconstrained and constrained minimization. *Mathematical Programming, Series B, 62*, 277–297.

Mukherjee, S., Osuna E., & Girosi, F. (1997). Nonlinear prediction of chaotic time series using support vector machines. In *NNSP'97: Neural networks for signal processing VII: Proceedings of IEEE signal processing society workshop, Amelia Island, FL, USA* (pp. 511–520).

Muller, K. R., Smola, A. J., Ratsch, G., Schölkopf, B., & Kohlmorgen, J. (1999). Using support vector machines for time series prediction. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in Kernel methods-support vector learning* (pp. 243–254). Cambridge, MA: MIT Press.

Musicant, D. R., & Feinberg, A. (2004). Active set support vector regression. *IEEE Transactions on Neural Networks, 15*(2), 268–275.

Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: An application to face detection. In *Proceedings of computer vision and pattern recognition* (pp. 130–136).

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1994). *Numerical recipes in C* (2nd ed.). Cambridge University Press.

Tay, F. E. H., & Cao, L. J. (2001). Application of support vector machines in financial time series with forecasting. *Omega, 29*(4), 309–317.

Vapnik, V. N. (2000). *The nature of statistical learning theory* (2nd ed.). New York: Springer.