Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 79 (2009) 2051–2063
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Linear programming support vector regression with wavelet kernel: A new approach to nonlinear dynamical systems identification

Zhao Lu [a], Jing Sun [b,*], Kenneth R. Butts [c]

[a] *Department of Electrical Engineering, Tuskegee University, Tuskegee, AL 36088, USA*
[b] *Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA*
[c] *North America Technical Center, Toyota Motor Corporation, Ann Arbor, MI 48105, USA*

Received 6 September 2007; received in revised form 2 August 2008; accepted 30 October 2008
Available online 18 November 2008

Abstract

Wavelet theory has a profound impact on signal processing as it offers a rigorous mathematical framework to the treatment of multiresolution problems. The combination of soft computing and wavelet theory has led to a number of new techniques. On the other hand, as a new generation of learning algorithms, support vector regression (SVR) was developed by Vapnik et al. recently, in which ε-insensitive loss function was defined as a trade-off between the robust loss function of Huber and one that enables sparsity within the SVs. The use of support vector kernel expansion also provides us a potential avenue to represent nonlinear dynamical systems and underpin advanced analysis. However, for the support vector regression with the standard quadratic programming technique, the implementation is computationally expensive and sufficient model sparsity cannot be guaranteed. In this article, from the perspective of model sparsity, the linear programming support vector regression (LP-SVR) with wavelet kernel was proposed, and the connection between LP-SVR with wavelet kernel and wavelet networks was analyzed. In particular, the potential of the LP-SVR for nonlinear dynamical system identification was investigated.

Keywords: Support vector regression; Wavelet kernel; Nonlinear systems identification; Linear programming

1. Introduction

Mathematical models that capture the behavior of dynamical systems are of great importance in almost all fields of science and engineering, specifically in control, signal processing and information science. Given that most systems encountered in the real world are complex and nonlinear, one challenge in developing a useful model is to achieve a proper trade-off between model simplicity and accuracy. Since a model is always only an approximation of real phenomena, having an approximation theory which allows for the analysis of model quality is of substantial importance. A fundamental principle in system modeling is the well-recognized Occam's razor hypothesis: 'plurality should not be posited without necessity', or in other words, the simpler a solution is, the more reasonable it is. This concept, known as the parsimonious principle, which ensures the simplest possible model that explains the data, is particularly relevant in nonlinear model building because the size of a nonlinear model can easily become explosively large.

* Corresponding author.
E-mail address: jingsun@umich.edu (J. Sun).

0378-4754/$36.00 © 2008 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2008.10.011

Forward selection using the orthogonal least squares (OLS) is an effective construction method that is capable of producing parsimonious linear-in-the-weights nonlinear models with excellent generalization performance [3]. Alternatively, the state-of-the-art sparse kernel modeling techniques, such as the support vector machine (SVM) [17,18] and relevant vector machine [9], have been gaining popularity in data modeling applications. SVM algorithms yield prediction functions that are expanded on a subset of training vectors, or support vectors, hence their names. Sparsity, defined as the ratio of the number of support vectors over the number of data points in the training set, is used to measure the model size and simplicity, thereby allowing the evaluation of the model quality against the parsimonious principle. For linear approximation, it has been pointed out in [2] that the solution found by SVM for regression is a trade-off between sparsity of the representation and closeness to the data. SVMs extend this linear interpretation to nonlinear approximation by mappings to a higher-dimensional feature space. This space can be of very high dimension, even infinite, because the parameters of weights are not explicitly calculated. By using certain kernel functions in the approximation function, nonlinear mappings can be made from input space to output, while the training procedure is concerned with linear mappings in an implied feature space.

In the conventional quadratic programming support vector machines (QP-SVMs), the prediction function yielded often contains redundant terms. The economy of an SVM prediction model is dependent on a sparse subset of the training data being selected as support vectors by the optimization technique. In many practical applications, the inefficiency of the conventional SVMs scheme for selecting support vectors could lead to infeasible models. This is particularly apparent in regression applications where the entire training set can be selected as support vectors if error insensitivity is not included [5].

A recent study has compared the standard SVM and uniformly regularized orthogonal least squares (UROLS) algorithms using time series prediction problems, and has found that both methods have similar excellent generalization performance but the resulting model from SVM is not sparse enough [12]. It is explained that the number of support vectors found by quadratic programming algorithm in a SVM is only an upper bound on the number of necessary and sufficient support vectors, and this is due to the linear dependencies between support vectors in feature space. Some efforts have been made attempt to control the sparsity in support vector machines [5].

Among a number of successful applications of SVMs in practice, it has been shown that the use of support vector kernel expansion also provides us a potential avenue to represent nonlinear dynamical systems and underpin advanced analysis [6,7,16]. Although it is believed that the formulation of SVM embodies the structural risk minimization principle, thus combining excellent generalization properties with a sparse model representation, data modeling practicians have begun to realize that the capability for the standard quadratic programming SVR (QP-SVR) method to produce sparse models has perhaps been overstated. For example, it has been shown that the standard SVM technique is not always able to construct parsimonious models in system identification [6].

In this article, for the purpose of developing an innovative and efficient identification algorithm for complex nonlinear dynamical systems, the issue of model sparsity was addressed from two different perspectives. First, the linear programming support vector regression (LP-SVR) is used to capitalize on the advantages of the model sparsity, the flexibility in using more general kernel functions, and the computational efficiency of linear programming [8,10], as compared to OP-SVR. The idea of LP-SVR is to use the kernel expansion as an ansatz for the solution, but to use a different regularizer, namely the ℓ_1 norm of the coefficient vector. In other words, for LP-SVR, the nonlinear regression problem is treated as a linear one in the kernel space, rather than in the feature space as in the case of QP-SVR. Second, considering the transient characteristic of nonlinear dynamical systems, an appropriate kernel function which is capable to capture the underlying nonstationary dynamics accurately might be expected to yield a more compact and sparse representation. Due to the localization feature in both frequency and time domains, wavelets have been successfully used to represent a much larger class of signals than Fourier representation [1]. Unlike Fourier-based analyses that use global sine and cosine functions as bases, wavelet analysis use bases that are localized in time and frequency to represent nonstationary signals more effectively. As a result, a wavelet expansion representation is much more compact and easier to implement.

This paper focuses on developing a new machine learning algorithm by combining the wavelet kernel function with LP-SVR, and particularly exploring their strength in identification of complex nonlinear dynamical systems. Special attention is paid to the sparsity of the generated model and its role in reducing the generalization error. This paper is organized as follows. In the next section, a brief review about wavelet and wavelet networks are given. The algorithm of LP-SVR with wavelet kernel is developed and discussed in Section 3. A case study with application to nonlinear dynamical system identification is conducted in Section 4, with concluding remarks in Section 5.

The following generic notations will be used throughout this paper: lower case symbols such as x, y, α, ... refer to scalar valued objects, lower case boldface symbols such as $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\beta}$, ... refer to vector valued objects, and finally capital boldface symbols will be used for matrices.

2. Wavelet and wavelet networks

2.1. Wavelet decomposition

Wavelet theory has been extensively studied in recent years and has found many applications in various areas throughout science and engineering, such as numerical analysis and signal processing. The main reason for the popularity of wavelet is its effectiveness in representing transient signals. The most significant property of wavelets, compared to other basis functions, is their ability to capture the local behavior of signals both in frequency and time. This localization feature makes many functions and operators using wavelets "sparse" when transformed into the wavelet domain. This sparseness, in turn, results in a number of useful applications such as data compression, detecting features in images, and removing noise from time series. Also, they lend themselves for adaptation in the sense that one can add or remove wavelet coefficients depending on the accuracy required, without affecting the remaining coefficients.

Two categories of wavelet functions, namely, orthogonal wavelets and wavelet frames, were developed separately by different groups. Orthogonal wavelet decomposition is usually associated with the theory of multiresolution analysis. The fact that orthogonal wavelets cannot be expressed in closed form is a serious drawback for their application to function approximation and process modeling. Conversely, wavelet frames are constructed by simple operations of translation and dilation of a single fixed function called the *analyzing wavelet* or *mother wavelet*, which must satisfy conditions that are less stringent than orthogonality conditions.

A wavelet $\phi_j(x)$ is derived from its mother wavelet $\phi(z)$ by the relation

$$\phi_j(x) = \phi\left(\frac{x - m_j}{d_j}\right) = \phi(z_j) \tag{1}$$

where the translation factor m_j and the dilation factor d_j are real numbers in $\Re$ and $\Re_+^*$, respectively. The family of functions generated by ϕ can be defined as

$$\Omega_c = \left\{ \frac{1}{\sqrt{d_j}}\phi\left(\frac{x - m_j}{d_j}\right),\ \ m_j \in \Re \ \text{ and } \ d_j \in \Re_+^* \right\}. \tag{2}$$

A family Ω_c is said to be a frame of $L^2(R)$ if there exists two constants A and B, $0 < A \le B < \infty$, such that for any square integrable function f, the following inequalities hold:

$$A||f||^2 \le \sum_{\substack{j \\ \phi_j \in \Omega_c}} |\langle \phi_j, f\rangle|^2 \le B||f||^2 \tag{3}$$

where $||f||$ denotes the norm of function f and $\langle f, g\rangle$ the inner product of functions f and g. Families of wavelet frames of $L^2(R)$ are universal approximators.

For the modeling of multivariable processes, the multi-dimensional wavelet must be defined. In the present work, we use a dictionary composed of tensor product wavelet functions, and the elements of our wavelet dictionaries are of the form

$$\Phi_j\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \prod_{k=1}^{n} \phi\left(\frac{x_k - m_{jk}}{d_j}\right) \tag{4}$$

where $\boldsymbol{m_j} = [m_{j1}, m_{j2}, \ldots, m_{jn}]^T$ is the translation vector and d_j is the dilation parameter, respectively.

2.2. Wavelet networks

Though the attractive theory of wavelet decomposition has offered efficient algorithms for various purposes, their implementations are usually limited to wavelets of small dimension. The reason is that constructing and storing wavelet basis of large dimension are of prohibitive cost [23]. In order to handle problems of larger dimension, it is necessary to develop algorithms whose implementations are less sensitive to the dimension. It is known that neural networks are powerful tools for handling problems of large dimension. Hence, in recent years the offspring of wavelet theory and neural networks – wavelet networks – have emerged and grown vigorously both in research and applications [15,19,23,24]. The wavelet networks were developed due to the similarity between wavelet decomposition and one-hidden-layer neural networks, where wavelets were introduced as activation functions of the hidden neurons in traditional feedforward neural networks with a linear output neuron. The multi-dimensional wavelet networks is written as follows

$$f_p(\boldsymbol{x}) = \sum_{j=1}^{p} w_j \Phi_j(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ and w_j are the weights of the network. Wavelet networks have been used in classification and identification problems with some success. The strength of wavelet networks lies in their capabilities of catching essential features in "frequency-rich" signals. In wavelet networks, both the position and the dilation of the wavelets are optimized in addition to the weights.

From the practical point of view, the initialization of networks topological structure and parameters presents a major problem with wavelet networks. A good initialization of wavelet neural networks is extremely important to obtain a fast convergence of the algorithm. Similar to radial basis function networks (and in contrast to neural networks using sigmoidal functions), a random initialization of all the parameters to small values (as usually done with neural networks) is not desirable since this may make some wavelets too local (small dilations) and make the components of the gradient of the cost function very small in areas of interest. In general, one wants to take advantage of the input space domains where the wavelets are not zero. Another problem that needs to be considered for training a wavelet network is how to determine the initial number of wavelets associated with the network. In our paper, from the perspective of model sparsity, these issues are addressed within a unified framework of LP-SVM with wavelet kernel, which is fundamentally different from the methods proposed in Refs. [4,22,25], where the wavelet kernel was used in the context of conventional quadratic programming SVM.

3. Linear programming SVM with wavelet kernel

3.1. Soft-constrained linear programming SVM

Conceptually there are some similarities between the LP-SVR and QP-SVR. Both algorithms adopt the ε-insensitive loss function, and use kernel functions in feature space.

Consider regression in the following set of functions

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \varphi(\boldsymbol{x}) + b \tag{6}$$

with given training data, $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ where ℓ denotes the total number of exemplars, $\boldsymbol{x_i} \in R^n$ are the input and $y_i \in R$ are the target output data. The nonlinear mapping $\varphi: R^n \to R^m$ $(m > n)$ maps the input data into a so-called high dimensional feature space (which can be infinite dimensional) and $\boldsymbol{w} \in R^m$, $b \in R$. In ε-SV regression, the goal is to find a function $f(\boldsymbol{x})$ that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time, is as smooth as possible. In the support vector method one aims at minimizing the empirical risk subject to elements of the structure

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*), \\ &\text{subject to} \quad \begin{cases} y_i - \langle \boldsymbol{w}, \varphi(\boldsymbol{x_i}) \rangle - b \le \varepsilon + \xi_i \\ \langle \boldsymbol{w}, \varphi(\boldsymbol{x_i}) \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \end{aligned} \tag{7}$$

where the ξ_i and ξ_i^* are the slack variables, corresponding to the size of the excess positive and negative deviation, respectively. This is a classic quadratic optimization problem with inequality constraints, and the optimization criterion penalizes data points whose y-values differ from $f(\boldsymbol{x})$ by more than ε. The constant $C>0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated. By defining the following ε-insensitivity loss function,

$$L(y_i - f(\boldsymbol{x_i})) = \begin{cases} 0, & \text{if } |y_i - f(\boldsymbol{x_i})| \le \varepsilon \\ |y_i - f(\boldsymbol{x_i})| - \varepsilon, & \text{otherwise} \end{cases} \tag{8}$$

the optimization problem (7) is equivalent to the following regularization problem,

$$\text{minimize} \quad R_{reg}[f] = \sum_{i=1}^{\ell} L(y_i - f(\boldsymbol{x_i})) + \lambda ||\boldsymbol{w}||^2 \tag{9}$$

where $f(\boldsymbol{x})$ is in the form of (6) and $\lambda||\boldsymbol{w}||^2$ is the regularization term. According to the well-known Representer Theorem [18], the solution to the regularization problem (9) can be written as the following SV kernel expansion provided $k(\boldsymbol{x_i}, \boldsymbol{x_i}) = 1$

$$f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \beta_i k(\boldsymbol{x}, \boldsymbol{x_i}) \tag{10}$$

where $k(\boldsymbol{x_i}, \boldsymbol{x})$ is the kernel function. Three commonly used kernel functions in literature are

- Gaussian radial basis function (GRBF) kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\frac{-||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\sigma^2}\right); \tag{11}$$

- Polynomial kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = (1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^q; \tag{12}$$

- Sigmoid kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\alpha \langle \boldsymbol{x}, \boldsymbol{x}' \rangle + \gamma) \tag{13}$$

where σ, q, α, γ are the adjustable parameters of the above kernel functions. The kernel function provides an elegant way of working in the feature space avoiding all difficulties inherent in high dimensions, and this method is applicable whenever an algorithm can be cast in terms of dot products. Defining

$$\boldsymbol{\beta} = [\beta_1\ \beta_2\ \ldots\ \beta_\ell]^T, \tag{14}$$

LP-SVR replaces (9) by

$$\text{minimize} \quad R_{reg}[f] = \sum_{i=1}^{\ell} L(y_i - f(\boldsymbol{x_i})) + \lambda ||\boldsymbol{\beta}||_1 \tag{15}$$

where $f(\boldsymbol{x})$ is in the form of (10) and $||\boldsymbol{\beta}||_1$ denotes the ℓ_1 norm in coefficient space. This regularization problem is equivalent to the following constrained optimization problem

$$\begin{aligned} &\text{minimize } \frac{1}{2}||\boldsymbol{\beta}||_1 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*), \\ &\text{subject to } \begin{cases} y_i - \sum_{j=1}^{\ell}\beta_j k(\boldsymbol{x_j}, \boldsymbol{x_i}) \le \varepsilon + \xi_i, \\ \sum_{j=1}^{\ell}\beta_j k(\boldsymbol{x_j}, \boldsymbol{x_i}) - y_i \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0. \end{cases} \end{aligned} \tag{16}$$

From the geometric perspective, it can be followed that $\xi_i\xi_i^* = 0$ in SV regression. Therefore, it is sufficient to just introduce slack variable ξ_i in the constrained optimization problem (16). Thus, we arrive at the following formulation of SV regression with fewer slack variables

$$\begin{aligned} &\text{minimize } \frac{1}{2}||\boldsymbol{\beta}||_1 + 2C\sum_{i=1}^{\ell}\xi_i, \\ &\text{subject to } \begin{cases} y_i - \sum_{j=1}^{\ell}\beta_j k(\boldsymbol{x_j}, \boldsymbol{x_i}) \le \varepsilon + \xi_i, \\ \sum_{j=1}^{\ell}\beta_j k(\boldsymbol{x_j}, \boldsymbol{x_i}) - y_i \le \varepsilon + \xi_i, \\ \xi_i \ge 0. \end{cases} \end{aligned} \tag{17}$$

In an attempt to convert the optimization problem above into a linear programming problem, we decompose β_i and $|\beta_i|$ as follows

$$\beta_i = \alpha_i^+ - \alpha_i^-, \quad |\beta_i| = \alpha_i^+ + \alpha_i^- \tag{18}$$

where $\alpha_i^+, \alpha_i^- \ge 0$. It is worth noting that the decompositions in (18) are unique, i.e., for a given β_i there is only one pair (α_i^+, α_i^-) which fulfills both equations. Furthermore, both variables cannot be larger than zero at the same time, i.e., $\alpha_i^+ \cdot \alpha_i^- = 0$. In this way, the ℓ_1 norm of β can be written as

$$||\boldsymbol{\beta}||_1 = \left(\underbrace{1, 1, \ldots, 1}_{\ell}, \underbrace{1, 1, \ldots, 1}_{\ell}\right)\begin{pmatrix}\boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^-\end{pmatrix} \tag{19}$$

where $\boldsymbol{\alpha}^+ = (\alpha_1^+, \alpha_2^+, \ldots, \alpha_\ell^+)^T$ and $\boldsymbol{\alpha}^- = (\alpha_1^-, \alpha_2^-, \ldots, \alpha_\ell^-)^T$. Furthermore, the constraints in the formulation (17) can also be written in the following vector form

$$\begin{pmatrix}\boldsymbol{K} & -\boldsymbol{K} & -\boldsymbol{I} \\ -\boldsymbol{K} & \boldsymbol{K} & -\boldsymbol{I}\end{pmatrix} \cdot \begin{pmatrix}\boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \\ \boldsymbol{\xi}\end{pmatrix} \le \begin{pmatrix}\boldsymbol{y} + \varepsilon \\ \varepsilon - \boldsymbol{y}\end{pmatrix} \tag{20}$$

where $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_\ell)^T$ and $\boldsymbol{I}$ is $\ell \times \ell$ identity matrix. Thus, the constrained optimization problem (17) can be implemented by the following linear programming problem with the variables

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^T \begin{pmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \\ \boldsymbol{\xi} \end{pmatrix}, \\ &\text{subject to} \quad \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{K} & -\boldsymbol{I} \\ -\boldsymbol{K} & \boldsymbol{K} & -\boldsymbol{I} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \\ \boldsymbol{\xi} \end{pmatrix} \leq \begin{pmatrix} \boldsymbol{y} + \varepsilon \\ \varepsilon - \boldsymbol{y} \end{pmatrix} \end{aligned} \tag{21}$$

where $\boldsymbol{c} = \left(\underbrace{1, 1, \ldots, 1}_{\ell}, \underbrace{1, 1, \ldots, 1}_{\ell}, \underbrace{2C, 2C, \ldots, 2C}_{\ell} \right)^T$.

In the QP-SVR case, the set of points not inside the tube coincides with the set of SVs. While, in the LP context, this is no longer true—although the solution is still sparse, any point could be an SV, even if it is inside the tube [21]. Actually, the sparse solution can still be obtained in LP-SVR even though the size of the insensitive tube is set to zero [5], due to the usage of soft constraints; however, usually sparser solution can be obtained by using non-zero ε.

3.2. Linear programming SVM with wavelet kernel

As the cornerstone in nonlinear support vector algorithm, the kernels provide a general framework to represent data. In this section, we address the use of wavelet kernel in linear programming support vector regression, and this also provide us an natural way to determine the topological structure and translation parameters of wavelet networks by efficient optimization approach. From Eq. (4), the wavelet kernel can be defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \prod_{k=1}^{n} \phi\left(\frac{x_k - x_k'}{d} \right), \tag{22}$$

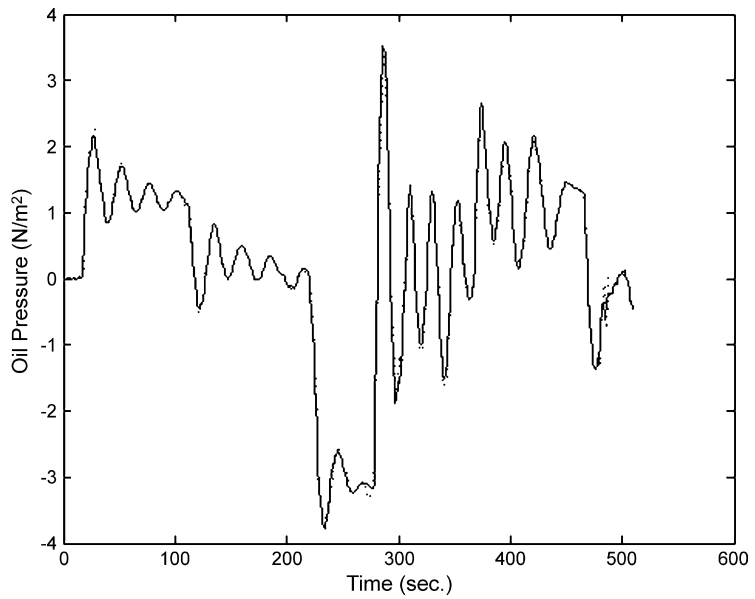

Fig. 1. Identification by LP-SVR with RBF kernel on the training set (solid line: observation, dotted line: model from LP-SVR).

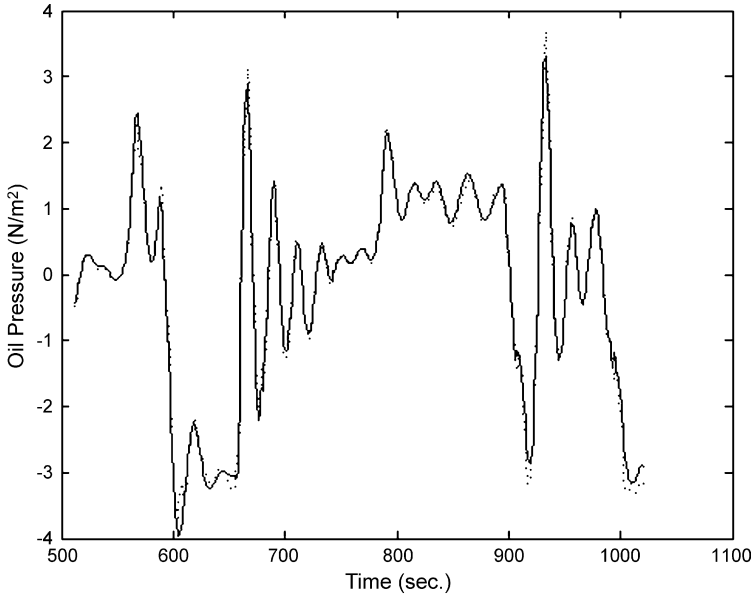

Fig. 2. Identification by LP-SVR with RBF kernel on the validation set (solid line: observation, dotted line: model from LP-SVR).

and the corresponding SV wavelet kernel expansion (10) can be written as

$$f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \beta_i \prod_{k=1}^{n} \phi\left(\frac{x_k - x_{ik}}{d_i}\right) = \sum_{i=1}^{\ell} (\alpha_i^+ - \alpha_i^-) \prod_{k=1}^{n} \phi\left(\frac{x_k - x_{ik}}{d_i}\right) \tag{23}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ is the input vector and $\boldsymbol{x_i} = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$ are the translation vectors, identical to the support vectors. The model sparsity in Eq. (23) is measured by the ratio of non-zero components in the vector $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \ldots\ \beta_\ell]^T$, i.e., the ratio of support vectors. In the realm of nonlinear systems identification, training data are

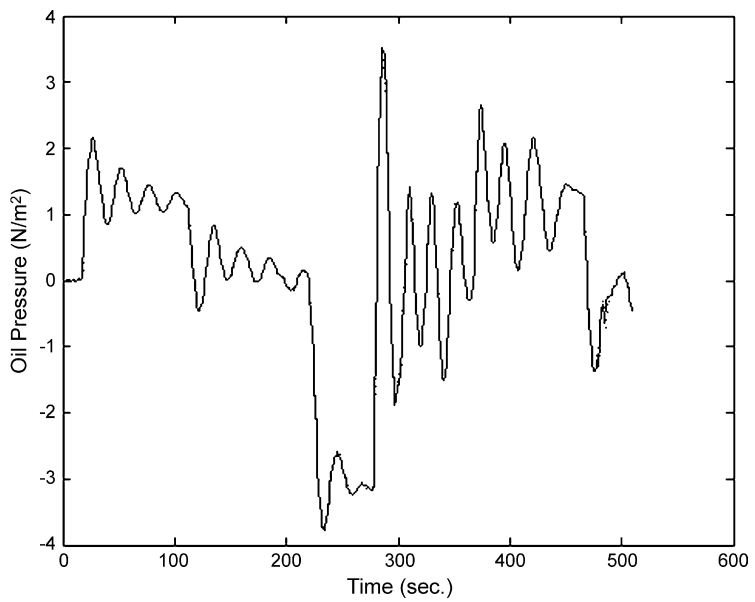

Fig. 3. Identification by QP-SVR with RBF kernel on the training set (solid line: observation, dotted line: model from QP-SVR).

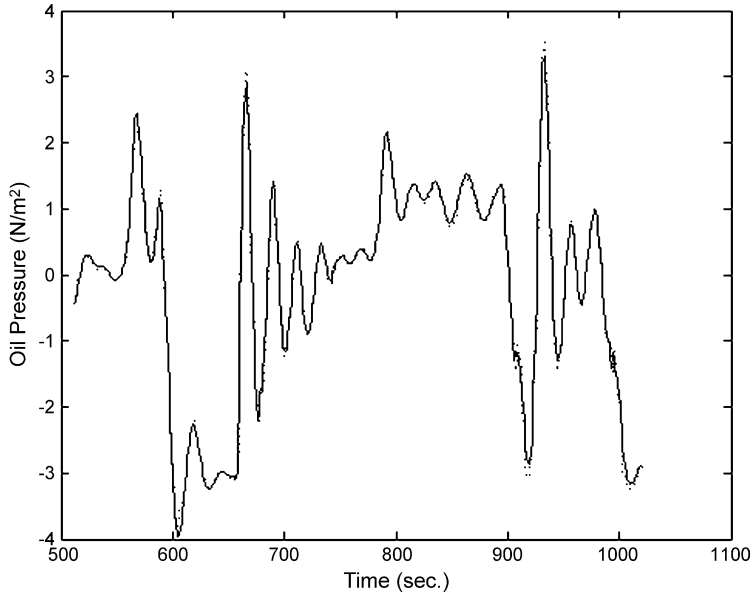

Fig. 4. Identification by QP-SVR with RBF kernel on the validation set (solid line: observation, dotted line: model from QP-SVR).

usually finite and non-uniformly sampled, so the problem is consequently ill-posed. Conversion to a well-posed problem is typically achieved with some form of capacity control, which aims to balance the fitting of the data with constraints on the model flexibility, producing a robust model that generalizes successfully. In practice, such an optimization is accomplished by searching for the minimum number of the basis functions under the Occam's razor arguing that the model should be no more complex than is required to capture the underlying systems dynamics. Hence, in an attempt to achieve the highest generalization capability and the lowest system complexity, the ℓ_1 norm of the weights in the model (23) was employed for model capacity control. In light of the derivation given in Section 3.1, the optimal compact representation, including the number of support vectors, the weights and the translation factors could be determined

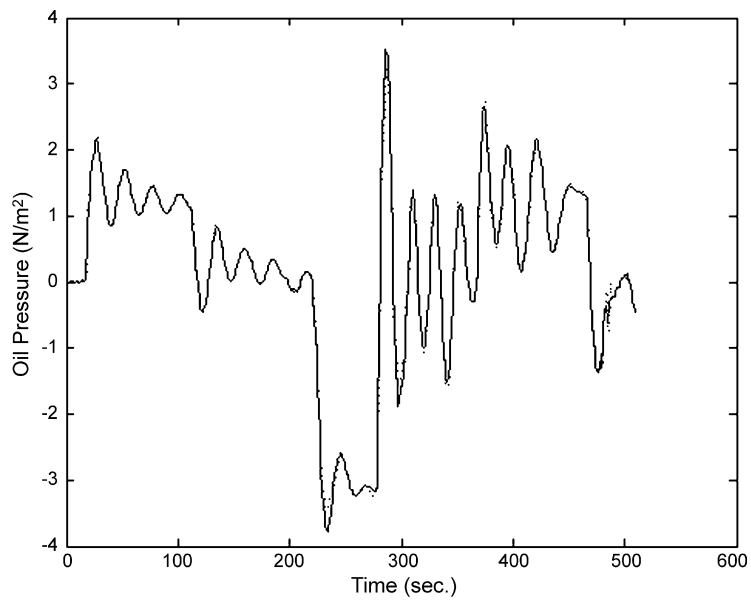

Fig. 5. Identification by LP-SVR with wavelet kernel on the training set (solid line: observation, dotted line: model from LP-SVR).

by solving the following linear programming problem

$$\begin{aligned} &\text{minimize } \boldsymbol{c}^T \begin{pmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \\ \boldsymbol{\xi} \end{pmatrix}, \\ &\text{subject to } \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{K} & -\boldsymbol{I} \\ -\boldsymbol{K} & \boldsymbol{K} & -\boldsymbol{I} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \\ \boldsymbol{\xi} \end{pmatrix} \leq \begin{pmatrix} \boldsymbol{y} + \varepsilon \\ \varepsilon - \boldsymbol{y} \end{pmatrix} \end{aligned} \tag{24}$$

where $\boldsymbol{K}$ denotes the wavelet kernels matrix. The resulting SV wavelet kernel expansion (23) is essentially consistent with the expression of wavelet networks (5), and therefore the construction of the optimal wavelet network is fulfilled by solving the linear programming problem.

4. Application

The hydraulic robot arm has been posed as a benchmark problem for nonlinear systems identification, and it has been used widely for testing various identification methods [7,20]. For this dynamical system, the input $u(t)$ represents the size of the valve through which oil flow into the actuator, and the output $y(t)$ is a measure of oil pressure which determines the robot arm position. For the purpose of comparison, we use the same regressor

$$\boldsymbol{x}(t) = [\, y(t-1) \quad y(t-2) \quad y(t-3) \quad u(t-1) \quad u(t-2) \,]^T \tag{25}$$

as that in [7,20], i.e.,

$$y(t) = f(\boldsymbol{x}(t)) = f(y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)), \tag{26}$$

which is a multi-dimensional regression model. We also use half of the data set containing 511 training data pairs for training, and half as validation data, again following the procedure used in [7,20]. The generalization capability and accuracy of regression algorithms could be evaluated using the root mean square (RMS) error

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}[\hat{f}(\boldsymbol{x_i}) - f(\boldsymbol{x_i})]^2} \tag{27}$$

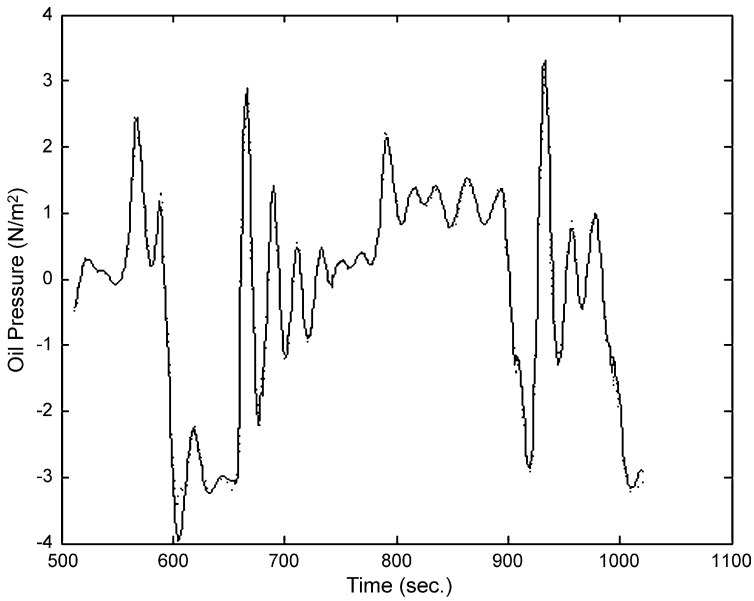

Fig. 6. Identification by LP-SVR with wavelet kernel on the validation set (solid line: observation, dotted line: model from LP-SVR).

where $\hat{f}(\boldsymbol{x_i})$ is the estimated value at point $\boldsymbol{x_i}$ from the SVR model. In this section, for the purpose of validating the superiority of wavelet kernel in LP-SVR, the wavelet kernel is compared with the RBF kernel, which provides universal nonlinear mapping capabilities and computational convenience [11,16].

Example 1. Identification of hydraulic robot arm dynamics by SVR with RBF kernel.

In this example, the soft-constrained LP-SVR and QP-SVR with RBF kernel are applied to model the dynamic behavior of a hydraulic robot arm respectively. The same kernel width parameter $\sigma = 1$ and tolerance for accuracy $\varepsilon = 0.03$ are used for LP-SVR and QP-SVR, and $C = 0.5$ for LP-SVR. For QP-SVR, due to the fact that C is the upper bound of the absolute value of the weights in support vectors expansion, it is set to a different value $C = 10$ to achieve the optimal performance. The identification results by LP-SVR are illustrated in Figs. 1 and 2, where the RMS error on validation data is 0.1528 and the ratio of SVs to the training vectors is 6.7%. Figs. 3 and 4 visualize the identification results by QP-SVR, where the RMS error on validation set is 0.1174 and the ratio of SVs to the training vectors is 37.2%.

In this example, the prediction accuracy from LP-SVR is comparable with that from QP-SVR, and the LP-SVR is around 25 times faster than QP-SVR for training.

Example 2. Identification of hydraulic robot arm dynamics by SVR with wavelet kernel.

For the sake of validating the performance of wavelet kernel in nonlinear identification, the soft-constrained LP-SVR and QP-SVR with wavelet kernel are used to model the dynamic behavior of a hydraulic robot arm respectively. Without loss of generality, a translation-invariant wavelet kernel could be constructed by using the following wavelet function [25]

$$\phi(x) = \cos(1.75x)\exp\left(-\frac{x^2}{2}\right) \tag{28}$$

The learning parameters ε and C for LP-SVR and QP-SVR are taken the same values as in Example 1. The translation vectors in the wavelet kernel, being equal to the support vectors, can be determined by the LP or QP algorithms. Since SVMs cannot optimize the dilation parameters of the wavelet kernel, usually it is difficult to determine ℓ parameters $d_i, \quad i = 1, 2, \ldots, \ell$ in Eq. (23). For simplicity, let $d_i = d$ such that the number of parameters becomes one [25]. In our experiment, the parameter d is set to 3 by the widely used technique of cross-validation.

The identification results by LP-SVR are illustrated in Figs. 5 and 6, where the RMS error on the validation data is 0.1455 and the ratio of SVs to the training vectors is 3.3%. Figs. 7 and 8 visualize the identification results

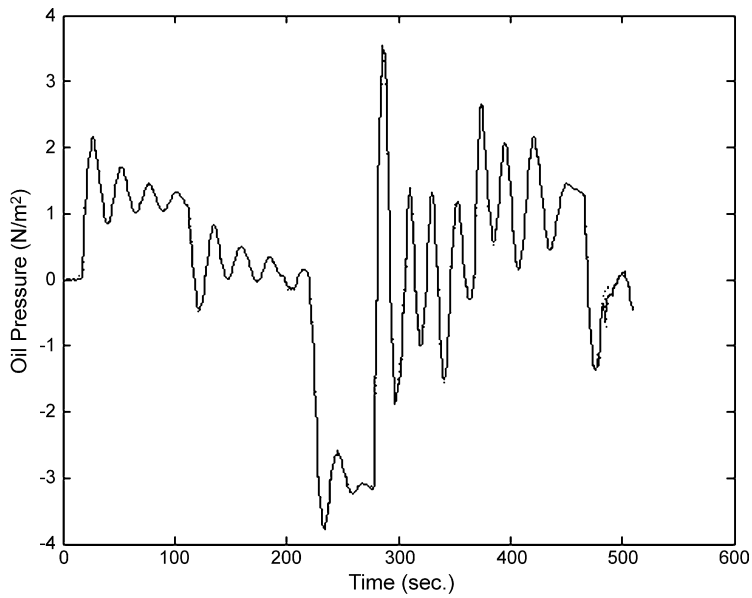

Fig. 7. Identification by QP-SVR with wavelet kernel on the training set (solid line: observation, dotted line: model from QP-SVR).

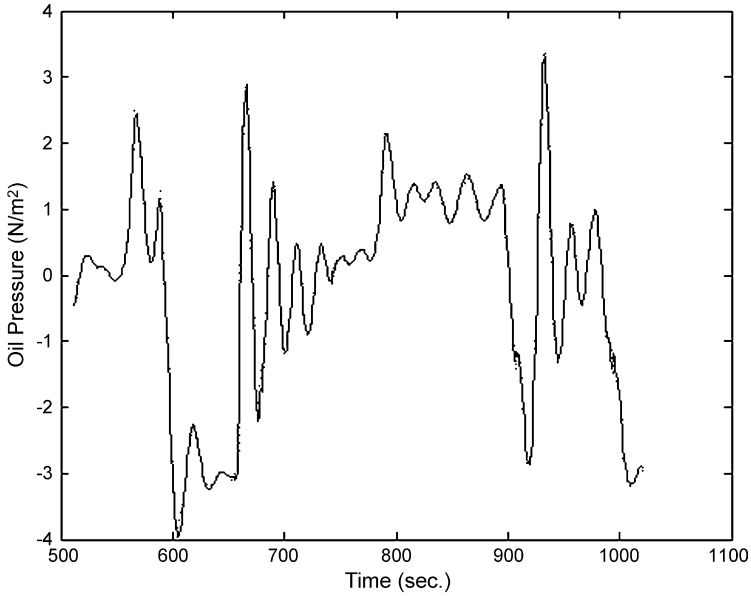

Fig. 8. Identification by QP-SVR with wavelet kernel on the validation set (solid line: observation, dotted line: model from QP-SVR).

Table 1
Comparison on model sparsity and RMS error of different SVRs.

SVR algorithm	SV ratio	RMS error on validation set
LP-SVR with RBF kernel	6.7%	0.1528
LP-SVR with wavelet kernel	3.3%	0.1455
QP-SVR with RBF kernel	37.2%	0.1174
QP-SVR with wavelet kernel	37.4%	0.1027

by QP-SVR, where the RMS error on the validation set is 0.1027 and the ratio of SVs to the training vectors is 37.4%.

The experimental results obtained in Examples 1 and 2 are summarized in Table 1.

It can be followed from Table 1 that much sparser model can be generated from LP-SVR than that from QP-SVR with comparable prediction accuracy, and much better sparsity can be obtained by using the LP-SVR with wavelet kernel. Particularly, the LP-SVR is around 25 times faster than QP-SVR for training, and as a nonlinear programming problem, the computing resources required by QP-SVR may be prohibitively expensive with the increase of the size of training set.

In Refs. [7,20], the RMS error on validation set acquired by using other identification algorithms are reported, such as 0.579 from wavelet networks, 0.467 from one-hidden-layer sigmoid neural networks with 10 hidden nodes, 0.280 from ν-support vector regression. Evidently, better prediction accuracy is obtained by using the soft-constrained LP-SVR and QP-SVR.

5. Conclusion and future work

In this article, from the perspective of model sparsity, the use of wavelet kernel in linear programming support vector regression for nonlinear dynamical systems identification was proposed and investigated. The proposed method enjoys the excellent generalization capability inherent in support vector learning and compact model expression. It could also be used to construct wavelet networks, and the idea behind our method also has the potential to be used in the realms of image compression and speech signal processing.

Our future research will concentrate on the development of on-line iterative algorithms for linear programming support vector regression with wavelet kernel, and the investigation of some intelligent optimization methods,

such as chaotic optimization algorithm [13,14], to determine the optimal dilation parameters in the generated model.

Acknowledgement

This research is funded by Toyota Motor Corporation.

References

[1] D. Allingham, Wavelet reconstruction of nonlinear dynamics, Int. J. Bifurcat. Chaos 8 (1998) 2191–2201.
[2] N. Ancona, Properties of support vector machines for regression, Technical Report, Center for Biological and Computational Learning, Massachusetts Institute of Technology, MA, 1999.
[3] S. Chen, Local regularization assisted orthogonal least squares regression, Neurocomputing 69 (2006) 559–585.
[4] G.Y. Chen, W.F. Xie, Pattern recognition with SVM and dual-tree complex wavelets, Image Vision Comput. 25 (2007) 960–966.
[5] P.M.L. Drezet, R.F. Harrison, A new method for sparsity control in support vector classification and regression, Pattern Recogn. 34 (2001) 111–125.
[6] P.M.L. Drezet, R.F. Harrison, Support vector machines for system identification, in: UKACC International Conference on Control, 1998.
[7] A. Gretton, A. Doucet, R. Herbrich, P.J.W. Rayner, B. Schölkopf, Support vector regression for black-box system identification, in: Proceedings of the 11th IEEE Workshop on Statistical Signal Processing, 2001.
[8] I. Hadzic, V. Kecman, Support vector machines trained by linear programming: theory and application in image compression and data classification, in: IEEE 5th Seminar on Neural Network Applications in Electrical Engineering, 2000.
[9] R. Herbrich, Learning Kernel Classifiers, MIT Press, Cambridge, 2002.
[10] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, MIT Press, 2001.
[11] S.S. Keerthi, C.J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Comput. 15 (2003) 1667–1689.
[12] K.L. Lee, S.A. Billings, Time series prediction using support vector machines, the orthogonal and the regularized orthogonal least-squares algorithms, Int. J. Syst. Sci. 33 (2002) 811–821.
[13] B. Li, W.S. Jiang, Optimizing complex functions by chaos search, Cybern. Syst. 29 (1998) 409–419.
[14] Z. Lu, L.S. Shieh, J. Chandra, Tracking control of nonlinear systems: a sliding mode design via chaotic optimization, Int. J. Bifurcat. Chaos 14 (4) (2004) 1343–1355.
[15] Y. Oussar, Training Wavelet networks for nonlinear dynamic input–output modeling, Neurocomputing 20 (1998) 173–188.
[16] J.L. Rojo-Alvarez, M. Martinez-Ramon, M. Prado-Cumplido, A. Artes-Rodriguez, A.R. Figueiras-Vidal, Support vector machines for nonlinear kernel ARMA system identification, IEEE Trans. Neural Netw. 17 (2006) 1617–1622.
[17] V.D. Sanchez, Advanced support vector machines and kernel methods, Neurocomputing 55 (2003) 5–20.
[18] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, 2002.
[19] S. Sitharama Lyengar, E.C. Cho, V.V. Phoha, Foundations of Wavelet Networks and Applications, Chapman & Hall/CRC, 2002.
[20] J. Sjöberg, Q. Zhang, L. Ljung, A. Berveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, Automatica 31 (1995) 1691–1724.
[21] A.J. Smola, B. Schölkopf, G. Rätsch, Linear programs for automatic accuracy control in regression, in: 9th International Conference on Artificial Neural Networks, London, 1999, pp. 575–580.
[22] Y. Tong, D. Yang, Q. Zhang, Wavelet kernel support vector machines for sparse approximation, J. Electron. (China) 23 (2006) 539–542.
[23] Q. Zhang, Using wavelet network in nonparametric estimation, IEEE Trans. Neural Netw. 8 (1997) 227–236.
[24] J. Zhang, Wavelet neural networks for function learning, IEEE Trans. Signal Process. 43 (1995) 1485–1497.
[25] L. Zhang, W. Zhou, L. Jiao, Wavelet support vector machine, IEEE Trans. Syst. Man Cybern., Part B: Cybern. 34 (2004) 34–39.