SUPPORT VECTOR MACHINE FOR THE SIMULTANEOUS APPROXIMATION OF A FUNCTION AND ITS DERIVATIVE

M. Lázaro[1], I. Santamaría[2], F. Pérez-Cruz[1], A. Artés-Rodríguez[1]

[1]Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III, Leganés, 28911, Madrid, Spain
e-mail: {marce,fernandop,antonio}@ieee.org
[2]Departamento de Ingeniería de Comunicaciones
Universidad de Cantabria, 39005 Santander, Spain
e-mail: nacho@gtas.dicom.unican.es

Abstract. In this paper, the problem of simultaneously approximating a function and its derivative is formulated within the Support Vector Machine (SVM) framework. The problem has been solved by using the ε-insensitive loss function and introducing new linear constraints in the approximation of the derivative. The resulting quadratic problem can be solved by Quadratic Programming (QP) techniques. Moreover, a computationally efficient Iterative Re-Weighted Least Square (IRWLS) procedure has been derived to solve the problem in large data sets. The performance of the method has been compared with the conventional SVM for regression, providing outstanding results.

INTRODUCTION

Regression approximation of a given data set is a very common problem in a number of applications. In some of these applications, like economy, device modeling, telemetry, etc, it is necessary to fit not only the underlying characteristic function but also its derivatives, which are often available. Some methods have been employed to simultaneously approximate a set of samples of a function and its derivative: splines, neural networks or filter bank-based methods are some examples (see [1] and references therein).

On the other hand, Support Vector Machines are state-of-the-art tools for linear and nonlinear input-output knowledge discovery [5, 4]. The Support Vector Machines, given a labeled dataset $((\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$) and a function $\boldsymbol{\phi}(\cdot)$ that nonlinearly transforms the input vector $\mathbf{x}_i$ to a higher dimensional space, solve either classification ($y_i \in \{\pm 1\}$) or regression ($y_i \in \mathbb{R}$) problems.

0-7803-8178-5/03/$17.00

In this paper, we will deal with the regression approximation problem and we will extend the SVM framework when prior knowledge about the derivative of the functional relation between $\mathbf{x}$ and y is known. We will solve this problem for one-dimensional problem ($d = 1$), but it can be readily extended to multidimensional input and the gradient information, as we will show herein. We would like to find the functional relation between x and y giving a labeled data set $((x_i, y_i, y_i^{'})$, where $y_i \in \mathbb{R}$ and $y_i^{'} \in \mathbb{R}$, which is the derivative of the function to be approximated at x_i).

We will solve it using the ε-insensitive loss function and it will lead to a solution similar to the SVM in which we have Support Vectors related to the function value (y_i) and Support Vectors related to the derivative value ($y_i^{'}$), and both together form the complete SVM expansion for regression approximation with information about the derivative of the function.

The solution to the proposed algorithm is obtained using an Iterative Re-Weighted Least Square (IRWLS) procedure that it is able to easily solve the SVM. This procedure has been successfully applied to the regular SVM for classification [2] for regression [3].

The rest of the paper is outlined as follows. The modification of the SVM for regression approximation to include the derivative information is presented in Section 2. In Section 3, we deal with the IRWLS procedure for linear machines. The use of nonlinear kernel and the application of the kernel trick to the IRWLS procedure is presented in Section 4. The generalization of the method to higher order input spaces is outlined in Section 5. Experimental results are presented in Section 6. We conclude the paper in Section 7 with some final comments and possible further works.

PROPOSED SVM-BASED APPROACH

As it has been outlined before, the proposed method is an extension of the Support Vector Machine for Regression (SVM-R) employing the Vapnik's ε-insensitive loss function [5]. The SVM-R obtains a linear regressor in the transformed space (feature space)

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

where $\mathbf{w}$ and b define the linear regression, which is nonlinear in the input space (unless $\phi(x)$ is linear). Roughly speaking, the SVM-R minimizes the squared norm of the weight vector $\mathbf{w}$, while linearly penalizes the deviations greater than ε.

With respect to the conventional SVM-R cost function, the proposed method adds a new penalty term: the errors in the derivative that are out of its associated insensitive region. In the general case, a different parameter is employed to define the insensitive region size for the function (ε) and for the derivative (ε'). Taking this extension into account, the proposed approach minimizes

$$J(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\tau}, \boldsymbol{\tau}^*) = \frac{1}{2}||\mathbf{w}||^2 + C_1 \sum_{i=1}^{N}(\xi_i + \xi_i^*) + C_2 \sum_{i=1}^{N}(\tau_i + \tau_i^*), \quad (2)$$

subject to

$$\mathbf{w}^T \phi(x_i) + b - y_i \leq \varepsilon + \xi_i, \quad (3)$$

$$y_i - \mathbf{w}^T \phi(x_i) - b \leq \varepsilon + \xi_i^*, \quad (4)$$

$$\mathbf{w}^T \phi'(x_i) - y_i' \leq \varepsilon' + \tau_i, \quad (5)$$

$$y_i' - \mathbf{w}^T \phi'(x_i) \leq \varepsilon' + \tau_i^*, \quad (6)$$

$$\xi_i, \xi_i^*, \tau_i, \tau_i^* \geq 0. \quad (7)$$

The positive slack variables ξ, ξ^*, τ and τ^* are responsible for penalizing errors greater than ε and ε', respectively, in the function and derivative, and $\phi'(x)$ denotes the derivative of $\phi(x)$. To solve this problem, the following Lagrangian functional is employed, introducing the previous linear constraints

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\tau}, \boldsymbol{\tau}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\lambda}, \boldsymbol{\lambda}^*) = \frac{1}{2}||\mathbf{w}||^2 + C_1 \sum_{i=1}^{N}(\xi_i + \xi_i^*) + C_2 \sum_{i=1}^{N}(\tau_i + \tau_i^*) \\ - \sum_{i=1}^{N} \alpha_i [\varepsilon + \xi_i - (\mathbf{w}^T \phi(x_i) + b - y_i)] - \sum_{i=1}^{N} \alpha_i^* [\varepsilon + \xi_i^* - (y_i - \mathbf{w}^T \phi(x_i) - b)] \\ - \sum_{i=1}^{N} \lambda_i [\varepsilon' + \tau_i - (\mathbf{w}^T \phi'(x_i) - y_i')] - \sum_{i=1}^{N} \lambda_i^* [\varepsilon' + \tau_i^* - (y_i' - \mathbf{w}^T \phi'(x_i))] \\ - \sum_{i=1}^{N} (\mu_i \xi_i + \mu_i^* \xi_i^* + \theta_i \tau_i + \theta_i^* \tau_i^*). \quad (8) \end{aligned}$$

This functional has to be minimized with respect to $\mathbf{w}$, b, ξ, ξ^*, τ and τ^*, and maximized with respect to the Lagrange multipliers. The solution to this problem can be obtained considering the Karush-Kuhn-Tucker (KKT) complementary conditions, which for this specific problem are

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} + \sum_{i=1}^{N} \alpha_i \phi(x_i) - \sum_{i=1}^{N} \alpha_i^* \phi(x_i) + \sum_{i=1}^{N} \lambda_i \phi'(x_i) - \sum_{i=1}^{N} \lambda_i^* \phi'(x_i) = 0, \quad (9)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0, \quad (10)$$

$$\frac{\partial L}{\partial \xi_i} = C_1 - \alpha_i - \mu_i = 0, \qquad \frac{\partial L}{\partial \xi_i^*} = C_1 - \alpha_i^* - \mu_i^* = 0, \quad (11)$$

$$\frac{\partial L}{\partial \tau_i} = C_2 - \lambda_i - \theta_i = 0, \qquad \frac{\partial L}{\partial \tau_i^*} = C_2 - \lambda_i^* - \theta_i^* = 0, \tag{12}$$

$$\alpha_i[\varepsilon + \xi_i - (\mathbf{w}^T\phi(x_i) + b - y_i)] = 0, \tag{13}$$

$$\alpha_i^*[\varepsilon + \xi_i^* - (y_i - \mathbf{w}^T\phi(x_i) - b)] = 0, \tag{14}$$

$$\lambda_i[\varepsilon' + \tau_i - (\mathbf{w}^T\phi'(x_i) - y_i')] = 0, \tag{15}$$

$$\lambda_i^*[\varepsilon' + \tau_i^* - (y_i' - \mathbf{w}^T\phi'(x_i))] = 0, \tag{16}$$

and

$$\mu_i\xi_i = 0, \quad \mu_i^*\xi_i^* = 0, \quad \theta_i\tau_i = 0, \quad \theta_i^*\tau_i^* = 0. \tag{17}$$

From (9), the weight vector $\mathbf{w}$ takes the form

$$\mathbf{w} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\phi(x_i) + \sum_{i=1}^{N}(\lambda_i^* - \lambda_i)\phi'(x_i), \tag{18}$$

which means that the regression is

$$f(x) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) < \phi(x_i), \phi(x) > + \sum_{i=1}^{N}(\lambda_i^* - \lambda_i) < \phi'(x_i), \phi(x) > + b. \tag{19}$$

Substituting (18) into (8), rearranging terms and taking into account (10)-(12), one thus arrives to the Wolfe's dual problem needing to maximize

$$\begin{aligned} W(\boldsymbol{\alpha}, \boldsymbol{\alpha^*}, \boldsymbol{\lambda}, \boldsymbol{\lambda^*}) = &-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) < \phi(x_i), \phi(x_j) > \\ &- \frac{1}{2}\sum_{i,j=1}^{N}(\lambda_i^* - \lambda_i)(\lambda_j^* - \lambda_j) < \phi'(x_i), \phi'(x_j) > \\ &- \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\lambda_j^* - \lambda_j) < \phi(x_i), \phi'(x_j) > \\ &- \frac{1}{2}\sum_{i,j=1}^{N}(\lambda_i^* - \lambda_i)(\alpha_j^* - \alpha_j) < \phi'(x_i), \phi(x_j) > \\ &- \sum_{i=1}^{N}[(\alpha_i + \alpha_i^*)\varepsilon + (\lambda_i + \lambda_i^*)\varepsilon'] + \sum_{i=1}^{N}[(\alpha_i^* - \alpha_i)y_i + (\lambda_i^* - \lambda_i)y_i'], \end{aligned} \tag{20}$$

subject to

$$0 \le \alpha_i, \alpha_i^* \le C_1, \tag{21}$$

$$0 \le \lambda_i, \lambda_i^* \le C_2, \tag{22}$$

and

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0. \tag{23}$$

It can be seen that (20) is a quadratic functional that only depends on the Lagrange multipliers α_i, α_i^*, λ_i and λ_i^*. This problem can be solved by Quadratic Programming (QP) techniques. Moreover, in the SVM framework, the nonlinear transformation $\phi(x)$ is not needed to be explicitly known and it can be replaced by its kernels. In this case, $< \phi(x_i), \phi(x_j) >$ is substituted by $K(x_i, x_j)$, a kernel satisfiying the Mercer Theorem [4]. From this definition for the kernel, it is easy to demonstrate that,

$$< \phi'(x_i), \phi(x_j) >= \frac{\partial K(x_i, x_j)}{\partial x_i} \triangleq K'(x_i, x_j), \tag{24}$$

$$< \phi(x_i), \phi'(x_j) >= \frac{\partial K(x_i, x_j)}{\partial x_j} \triangleq G(x_i, x_j), \tag{25}$$

and

$$< \phi'(x_i), \phi'(x_j) >= \frac{\partial^2 K(x_i, x_j)}{\partial x_i \partial x_j} \triangleq J(x_i, x_j). \tag{26}$$

Though K must be a Mercer Kernel, its derivatives do not have to. Therefore, using a valid kernel $K(x_i, x_j)$, once the Lagrange multipliers have been obtained, the regression estimate takes the form

$$f(x) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(x_i, x) + \sum_{i=1}^{N} (\lambda_i^* - \lambda_i) K'(x_i, x) + b. \tag{27}$$

IRWLS ALGORITHM

The QP solution of the system can be computationally expensive, especially when a large number of samples is employed. In this case the computational burden can make the problem unaffordable. In order to reduce the computational burden, an Iterative Re-Weighted Least Square (IRWLS) procedure has been developed. This IRWLS algorithm follows the same basic idea proposed in [3]. First at all, the Lagrangian (8) is rearranged to group the term depending on ξ_i, ξ_i^*, τ_i and τ_i^*. Taking into account (11)-(12), these terms can be eliminated. Therefore, (8) can be written as

$$\begin{aligned} L = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i [\varepsilon + y_i - \mathbf{w}^T \phi(x_i) - b] - \sum_{i=1}^{N} \alpha_i^* [\varepsilon + \mathbf{w}^T \phi(x_i) + b - y_i] \\ - \sum_{i=1}^{N} \lambda_i [\varepsilon' + y_i' - \mathbf{w}^T \phi'(x_i)] - \sum_{i=1}^{N} \lambda_i^* [\varepsilon' + \mathbf{w}^T \phi'(x_i) - y_i']. \end{aligned} \tag{28}$$

This functional can be rewritten as

$$L = \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{2} \sum_{i=1}^{N} (a_i e_i^2 + a_i^* (e_i^*)^2) + \frac{1}{2} \sum_{i=1}^{N} (s_i d_i^2 + s_i^* (d_i^*)^2), \tag{29}$$

where

$$e_i = \mathbf{w}^T\phi(x_i) + b - y_i - \varepsilon, \quad a_i = \frac{2\alpha_i}{e_i} \tag{30}$$

$$e_i^* = y_i - \mathbf{w}^T\phi(x_i) - b - \varepsilon, \quad a_i^* = \frac{2\alpha_i^*}{e_i^*} \tag{31}$$

$$d_i = \mathbf{w}^T\phi'(x_i) - y_i' - \varepsilon', \quad s_i = \frac{2\lambda_i}{d_i} \tag{32}$$

$$d_i^* = y_i' - \mathbf{w}^T\phi'(x_i) - \varepsilon', \quad s_i^* = \frac{2\lambda_i^*}{d_i^*} \tag{33}$$

This functional can be seen as a weighted least square one, where e_i, and e_i^* are the prediction error over the function, d_i and d_i^* are the prediction error over the derivative, and a_i, a_i^*, s_i and s_i^* are the corresponding weights. It is necessary to iterate because $a_i = a_i(e_i)$, $a_i^* = a_i^*(e_i^*)$, $s_i = s_i(d_i)$ and $s_i^* = s_i^*(d_i^*)$. The goal at each iteration is to minimize (28) with respect to $\mathbf{w}$ and b, supposing that a_i, a_i^*, s_i and s_i^* are fixed. Taking the derivative with respect to both variables, the following two equations are obtained:

$$\begin{bmatrix} \mathbf{\Phi}^T\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{\Phi} + \mathbf{\Phi}'^T\mathbf{D}_{\mathbf{s}+\mathbf{s}^*}\mathbf{\Phi}' + \mathbf{I} & \mathbf{\Phi}^T\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{1} \\ (\mathbf{a}+\mathbf{a}^*)^T\mathbf{\Phi} & (\mathbf{a}+\mathbf{a}^*)^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}^T[\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{y} + \mathbf{D}_{\mathbf{a}-\mathbf{a}^*}\mathbf{1}\varepsilon] + \mathbf{\Phi}'^T[\mathbf{D}_{\mathbf{s}+\mathbf{s}^*}\mathbf{y}' + \mathbf{D}_{\mathbf{s}-\mathbf{s}^*}\mathbf{1}\varepsilon'] \\ (\mathbf{a}+\mathbf{a}^*)^T\mathbf{y} + (\mathbf{a}-\mathbf{a}^*)^T\mathbf{1}\varepsilon \end{bmatrix}, \tag{34}$$

where $\mathbf{a}$, $\mathbf{a}^*$, $\mathbf{s}$ and $\mathbf{s}^*$ are the vectors containing the N corresponding weights in (29), $\mathbf{D}_\mathbf{a}$ denotes the diagonal matrix with vector $\mathbf{a}$ in the diagonal, and

$$\begin{matrix} \mathbf{\Phi} = [\phi(x_1), \phi(x_2), \cdots, \phi(x_N)]^T \\ \mathbf{\Phi}' = [\phi'(x_1), \phi'(x_2), \cdots, \phi'(x_N)]^T \end{matrix} . \tag{35}$$

IRWLS WITH KERNELS

The system (34) can be used when $\phi(x)$ is known. When working with kernels, it is necessary to obtain the Lagrange multipliers to provide the regression (27). In this case, it must be taken into account that the weight vector can be written as

$$\mathbf{w} = [\mathbf{\Phi}^T, \mathbf{\Phi}'^T] \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix}, \tag{36}$$

where $\boldsymbol{\beta} = \boldsymbol{\alpha}^* - \boldsymbol{\alpha}$ and $\boldsymbol{\gamma} = \boldsymbol{\lambda}^* - \boldsymbol{\lambda}$. Now (36) will be used to replace $\mathbf{w}$ in (34). The first equation can be written as

$$[\mathbf{\Phi}^T\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{\Phi} + \mathbf{\Phi}'^T\mathbf{D}_{\mathbf{s}+\mathbf{s}^*}\mathbf{\Phi}' + \mathbf{I}]\mathbf{w} = [\mathbf{\Phi}^T, \mathbf{\Phi}'^T] \begin{bmatrix} \mathbf{D}_{\mathbf{a}+\mathbf{a}^*}(\mathbf{y} - \mathbf{1}b) + \mathbf{D}_{\mathbf{a}-\mathbf{a}^*}\mathbf{1}\varepsilon \\ \mathbf{D}_{\mathbf{s}+\mathbf{s}^*}\mathbf{y}' + \mathbf{D}_{\mathbf{s}-\mathbf{s}^*}\mathbf{1}\varepsilon' \end{bmatrix}. \tag{37}$$

Substituting $\mathbf{w}$ by (36), multiplying in both sides by

$$([\mathbf{\Phi}^T, \mathbf{\Phi}'^T]^T)^+ = \left(\begin{bmatrix} \mathbf{\Phi} \\ \mathbf{\Phi}' \end{bmatrix} [\mathbf{\Phi}^T, \mathbf{\Phi}'^T] \right)^{-1} \begin{bmatrix} \mathbf{\Phi} \\ \mathbf{\Phi}' \end{bmatrix}, \tag{38}$$

making mathematical arrangements and taking into account that we have two sets of decoupled equations, this equation can be written down as

$$\mathbf{H}^{-1} \left\{ \mathbf{H} \begin{bmatrix} \mathbf{D_{a+a^*}} & 0 \\ 0 & \mathbf{D_{s+s^*}} \end{bmatrix} \mathbf{H} + \mathbf{H} \right\} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{D_{a+a^*}}(\mathbf{y} - \mathbf{1}b) + \mathbf{D_{a-a^*}}\mathbf{1}\varepsilon \\ \mathbf{D_{s+s^*}}\mathbf{y}' + \mathbf{D_{s-s^*}}\mathbf{1}\varepsilon' \end{bmatrix}. \tag{39}$$

$\mathbf{H}$ can be easily computed from the definition of $K(x_i, x_j)$ and (24)-(26)

$$\mathbf{H} = \begin{bmatrix} \mathbf{\Phi}\mathbf{\Phi}^T & \mathbf{\Phi}\mathbf{\Phi}'^T \\ \mathbf{\Phi}'\mathbf{\Phi}^T & \mathbf{\Phi}'\mathbf{\Phi}'^T \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{K}' & \mathbf{J} \end{bmatrix}, \tag{40}$$

where $\mathbf{K}|_{ij} = K(x_i, x_j)$, $\mathbf{K}'|_{ij} = K'(x_i, x_j)$, $\mathbf{G}|_{ij} = G(x_i, x_j)$ and $\mathbf{J}|_{ij} = J(x_i, x_j)$. Canceling $\mathbf{H}^{-1}\mathbf{H}$, multiplying by the inverse of the diagonal matrix in (39) and moving b to the first term, the equation can be simplified. Finally, instead of using the second equation in (34), the simpler constraint (10) can be used, leading to the following whole system

$$\begin{bmatrix} \mathbf{H} + \begin{bmatrix} \mathbf{D_{a+a^*}} & 0 \\ 0 & \mathbf{D_{s+s^*}} \end{bmatrix}^{-1} & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} \mathbf{1}^T, \mathbf{0}^T \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} + \frac{\mathbf{a}-\mathbf{a}^*}{\mathbf{a}+\mathbf{a}^*}\varepsilon \\ \mathbf{y}' + \frac{\mathbf{s}-\mathbf{s}^*}{\mathbf{s}+\mathbf{s}^*}\varepsilon' \\ 0 \end{bmatrix}, \tag{41}$$

where $\frac{\mathbf{a}-\mathbf{a}^*}{\mathbf{a}+\mathbf{a}^*}$ denotes the vector containing $(a_i - a_i^*)/(a_i + a_i^*)$ in the i-th row.

Recalculating the weights

After each iteration, the new values for the weights a_i, a_i^*, s_i and s_i^* have to be calculated. It is simple to demonstrate, looking at the KKT conditions, that the corresponding values are given by

$$a_i = \begin{cases} 0, & e_i < 0 \\ \dfrac{2C_1}{e_i}, & e_i > 0 \end{cases}, \qquad a_i^* = \begin{cases} 0, & e_i^* < 0 \\ \dfrac{2C_1}{e_i^*}, & e_i^* > 0 \end{cases} \tag{42}$$

$$s_i = \begin{cases} 0, & d_i < 0 \\ \dfrac{2C_2}{d_i}, & d_i > 0 \end{cases}, \qquad s_i^* = \begin{cases} 0, & d_i^* < 0 \\ \dfrac{2C_2}{d_i^*}, & d_i^* > 0 \end{cases} \tag{43}$$

It must be noted that in (41) the inverse of these weights is used. In any case, when the estimated errors are null or very low, a maximum limit has to be imposed to the weight constants. This has the effect of adding a small number in the diagonal of the system in this case. This guarantees the matrix system is invertible. With this numerical trick to ensure convergence, the IRWLS algorithm can be summarized in Table 1.

1. Initialization:
   - Compute $\mathbf{H}$ (from $\mathbf{K}$, $\mathbf{K}'$, $\mathbf{G}$ and $\mathbf{J}$)
   - $a_i = C_1$, $s_i = C_2$ for odd i; $a_i^* = C_1$, $s_i^* = C_2$ for even i.
2. To solve (41)
3. To evaluate

$$\begin{array}{ll} \mathbf{e} = \mathbf{K}^T\boldsymbol{\beta} + \mathbf{K}'^T\boldsymbol{\gamma} + \mathbf{1}b - \mathbf{y} - \mathbf{1}\varepsilon, & \mathbf{e}^* = \mathbf{y} - \mathbf{K}^T\boldsymbol{\beta} - \mathbf{K}'^T\boldsymbol{\gamma} - \mathbf{1}b - \mathbf{1}\varepsilon \\ \mathbf{d} = \mathbf{G}^T\boldsymbol{\beta} + \mathbf{J}^T\boldsymbol{\gamma} - \mathbf{y}' - \mathbf{1}\varepsilon', & \mathbf{d}^* = \mathbf{y}' - \mathbf{G}^T\boldsymbol{\beta} - \mathbf{J}^T\boldsymbol{\gamma} - \mathbf{1}\varepsilon' \end{array}$$

4. Recalculate a_i, a_i^*, s_i^* and s_i^* by (42) and (43) (with a maximum limit)
5. Go to step 2 until convergence is achieved

TABLE 1: IRWLS ALGORITHM PSEUDOCODE

EXTENSION OF THE METHOD

The proposed method can be easily extended to d-dimensional input spaces and to consider up to k-th order derivatives. It is only necessary to incorporate the corresponding constraints. Because of the space limitation we have omitted the development, but in this case the solution takes the form

$$\mathbf{w} = \sum_{i=1}^{N} \sum_{j_1=0}^{k} \cdots \sum_{j_d=0}^{k} (\lambda^*_{ij_1\cdots j_d} - \lambda_{ij_1\cdots j_d}) \frac{\partial^{(j_1+\cdots+j_d)}\phi(\mathbf{x}_i)}{\partial x_1^{j_1}, \cdots, x_d^{j_d}}, \tag{44}$$

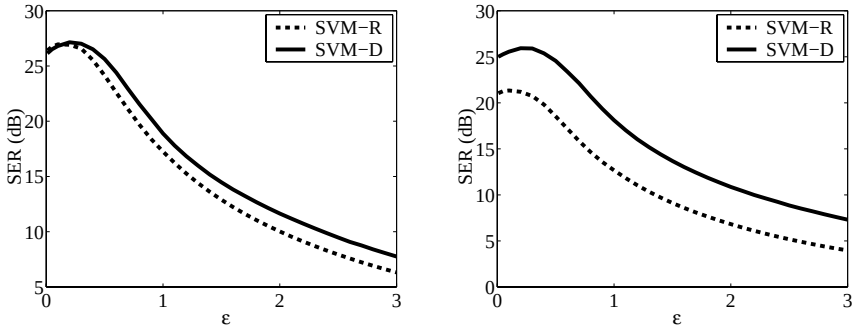
where $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{id}]^T$ and $\lambda^*_{ij_1\cdots j_d}$ and $\lambda_{ij_1\cdots j_d}$ are the Lagrange multipliers associated to the constraint in the i-th sample of

$$\frac{\partial^{(j_1+\cdots+j_d)} f(\mathbf{x})}{\partial x_1^{j_1}, \cdots, x_d^{j_d}}.$$

RESULTS

In this section, some experimental results show the advantages of this method in the reconstruction of the derivative with respect to the conventional SVM-R approach. As test functions, we have selected a set of bandlimited functions: specifically, in each experiment a linear combination of 100 sinusoids with random amplitudes, frequencies (between 0 and 1 Hz) and phases has been generated. In the first example, 100 equally spaced sampling points in the range 0-5 have been employed by the SVM-R (100 samples of the function), And 50 points by the proposed approach (50 samples of the function + 50 samples of the derivative). In this way, the number of total available data is the same. 1000 experiments have been considered, with a signal to noise ratio (SNR) of 20 dB in the samples of both the function and the derivative.

Figure 1 plots the mean values of signal to error ratio (SER) in the reconstruction of the function (a) and of the derivative (b) as a function of the insensitivity parameter ε. In this case, $\varepsilon' = \pi\varepsilon$ has been considered to take into account the different amplitude range of function and derivative (the mean amplitude of the derivative is π times higher than the mean amplitude of the function).



(a) Function (b) Derivative

Figure 1: SER in the reconstruction of function and derivative

It can be seen that the proposed method (labeled SVM-D) provides better results than the SVM-R, specially in the reconstruction of the derivative. Moreover, a similar number of support vectors has been observed for both methods in all simulations (in the proposed method: support vectors related to the function + support vectors related to the derivative). Therefore, the proposed method does not increment the storage requirements of the model. The number of support vectors, as a function of ε, is ploted in Figure 2.

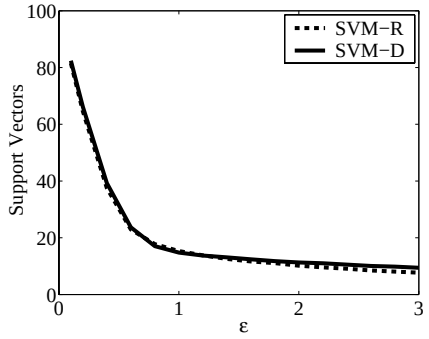
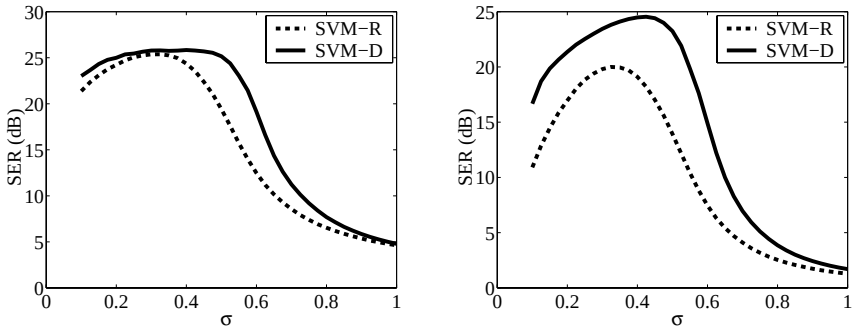


Figure 2: Number ot total support vectors as a function of ε

Finally, this method reduces the sensitivity of the method to the selection of σ for the Gaussian kernels. Figure 3 shows the SER in the reconstruction of the function as a function of σ for $\varepsilon = 0.5$. It can be seen that the σ range to obtain high SER values is increased by using the proposed method.



(a) Function (b) Derivative

Figure 3: SER as a function of the Kernel size

CONCLUSIONS

A new SVM-based method for the simultaneous reconstruction of a function and its derivative has been presented. A computationally efficient IRWLS algorithm has been derived to allow the application of the method to large data sets. This method provides better results than the conventional SVM-R approach in the reconstruction of function and derivative even when the same number of labeled data is employed in both methods. In this case, the proposed method needs a similar number of support vectors than conventional SVM-R. Moreover, the inclusion of the information of the derivatives reduces the dependence on the kernel size for Gaussian kernels. Expression have been presented for a one-dimensional input space and only the first derivative, but the extension of the method to higher order input spaces and derivative orders is straightforward by adding the corresponding constraints to the regression. Further work in necessary to simplify the parameter selection similarly to the ν-SVM [4].

REFERENCES

[1] M. Lázaro, I. Santamaría, C. Pantaleón, J. Ibáñez and L. Vielva, "A Regularized technique for the simultaneous reconstruction of a function and its derivatives with application to nonlinear transistor modeling," **Signal Processing**. In Press, 2003.

[2] F. Pérez-Cruz, A. Navia-Vázquez, J. L. Rojo-Álvarez and A. Artés-Rodríguez, "A New Training Algorithm for Support Vector Machines," in **Proceedings of the Fifth Bayona Workshop on Emerging Technologies in Telecommunications**, Baiona, Spain, Sept. 1999, pp. 116–120.

[3] F. Pérez-Cruz, A. Navia-Vázquez, P. Alarcón-Diana and A. Artés-Rodríguez, "An IRWLS procedure for SVR," in **Proceedings of the EUSIPCO'00**, Tampere, Finland, Sept. 2000.

[4] B. Schölkopf and A. Smola, **Learning with Kernels**, MIT Press, 2002.

[5] V. N. Vapnik, **Statistical Learning Theory**, New York: Wiley & Sons, 1998.