Learning a function and its derivative forcing the support vector expansion

Marcelino Lázaro, *Member, IEEE,* Fernando Pérez-Cruz, *Member, IEEE,* and Antonio Artés-Rodríguez, *Senior Member, IEEE*

AUTHORS: COMPLETE CONTACT INFORMATION

Marcelino Lázaro*.
Dpto de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid.
Av. de la Universidad nº30, 28911 Leganés, Madrid. Spain.
e-mail:marce@ieee.org Phone: +34 91 624 8769 Fax: +34 91 624 8749

Fernando Pérez-Cruz.
Gatsby Computational Neuroscience Unit, UCL.
Alexandra House, 17 Queen Square, London WC1N 3AR. United Kingdom.
e-mail:fernandop@ieee.org Phone: +44 (0)20 7679 1191 Fax: +44 (0)20 7679 1173

Antonio Artés-Rodríguez.
Dpto de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid.
Av. de la Universidad nº30, 28911 Leganés, Madrid. Spain.
e-mail:antonio@ieee.org Phone: +34 91 624 8741 Fax: +34 91 624 8749

Abstract

In this paper, a new method for the simultaneous learning of a function and its derivative is presented. The method, setting out the problem inside of the Support Vector Machine (SVM) framework, relies on the kernel-based Support Vector (SV) expansion. The resultant optimization problem is solved by a computationally efficient Iterative Re-Weighted Least Squares (IRWLS) algorithm.

EDICS Category: 1-NEUR

Index Terms

SVM, IRWLS, Support Vectors, function approximation.

This work was partially supported by CICYT grant TIC2003-2602

I. Introduction

The reconstruction of a function subject to a suitable approximation of its derivatives is a common problem in a number of applications. Nonlinear modeling of microwave (MESFET/HEMT) transistors to predict the intermodulation distortion (IMD) behavior [1], analysis of chaotic systems or economy are some examples. In other applications, such as telemetry or aerial traffic control, the information about the derivatives appears naturally and it can be interesting to use such information in the reconstruction task.

In a previous paper [2], we developed a non-linear Support Vector Machine (SVM) method to simultaneously approximate a function and its derivative. Although the accuracy of the method is excellent, the number of support vectors is relatively high, and the proposed expansion requires to evaluate, besides the kernel function underlying the SVM framework, the derivative of the kernel. In this communication, we force the solution to adopt the conventional support vector expansion, and we use the same loss function and regularizer as in the SVM method in [2]. Finally, the obtained optimization problem is solved by an Iterative Re-Weighted Least Squares (IRWLS) algorithm [3].

The rest of the paper is organized as follows. The problem is stated in Section II. Section III presents the proposed method, and Section IV develops the IRWLS algorithm used to solve the constrained optimization problem. Some experimental results show the performance of the method in Section V. Finally, the main conclusions are expounded in Section VI.

II. Problem Statement

For simplicity, a one-dimensional input space and only the first derivative are considered. In this case, the problem can be stated as follows: given a set of N samples of a function and its derivative

$$x_i \rightarrow (y_i, y_i'), \qquad i = 1, \cdots, N, \tag{1}$$

the goal is to simultaneously fit both the function and the derivative using a support vector machine based approach. In [2], we proposed a first solution to this problem by extending the support vector machine method for regression employing the Vapnik's ε-insensitive loss function [4]. In the general case, a different insensitive margin is employed for the function (ε) and for the derivative (ε'), leading to the following constrained optimization problem: to minimize

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

subject to

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

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

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

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

$$\xi_i, \xi_i^*, \tau_i, \tau_i^* \geq 0. \tag{7}$$

Here, $\boldsymbol{\phi}(x)$ is the nonlinear transformation function that underlies in the SVM theory to map the input space to a higher dimensional feature space, and $\boldsymbol{\phi}'(x)$ denotes the derivative of $\boldsymbol{\phi}(x)$. The positive slack variables ξ_i, ξ_i^*, τ_i and τ_i^* are responsible for penalizing errors greater than ε and ε', respectively, in the function and derivative. The solution, according to the Karush-Kuhn-Tucker (KKT) conditions, takes the form (see [2] for details)

$$f(x) = \sum_{i=1}^{N} \alpha_i \langle \boldsymbol{\phi}(x_i), \boldsymbol{\phi}(x) \rangle + \sum_{i=1}^{N} \gamma_i \langle \boldsymbol{\phi}'(x_i), \boldsymbol{\phi}(x) \rangle + b. \tag{8}$$

Typically, the inner product $\langle \boldsymbol{\phi}(x), \boldsymbol{\phi}(y) \rangle$ is replaced by $K(x, y)$, a kernel function satisfying the Mercer Theorem [4]. Under this assumption, it is easy to demonstrate that $\langle \boldsymbol{\phi}'(x), \boldsymbol{\phi}(y) \rangle$ is given by the derivative of $K(x, y)$ with respect to x.

III. Proposed Method

The method in [2] provides outstanding results and has shown the benefits of including the derivatives in the reconstruction of a function [2], [5]. However, in most applications it is preferable to provide a more compact expansion in order to minimize the evaluation cost. In this paper we propose to force the solution to adopt the SV expansion [6],

$$f(x) = \sum_{i=1}^{N} \beta_i K(x_i, x) + b, \tag{9}$$

instead of (8). This expansion can also be seen as a single hidden-layer feedforward network. Under very mild constraints for $K(\cdot, \cdot)$, this kind of networks have the capability to approximate with any given accuracy an arbitrary function and its derivatives [7]. To force the expansion (9) means that the weight vector has to be

$$\mathbf{w} = \sum_{i=1}^{N} \beta_i \boldsymbol{\phi}(x_i). \tag{10}$$

Taking into account the relationship between kernel and transformation, the problem reduces now to solve the following constrained optimization problem: to minimize

$$L_R(\boldsymbol{\beta}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\tau}, \boldsymbol{\tau}^*) = \frac{1}{2}\boldsymbol{\beta}^T \mathbf{K}\boldsymbol{\beta} + C_1 \sum_{i=1}^{N}(\xi_i + \xi_i^*) + C_2 \sum_{i=1}^{N}(\tau_i + \tau_i^*), \tag{11}$$

subject to

$$\sum_{j=1}^{N} \beta_j K(x_j, x_i) + b - y_i \leq \varepsilon + \xi_i, \tag{12}$$

$$y_i - \sum_{j=1}^{N} \beta_j K(x_j, x_i) - b \leq \varepsilon + \xi_i^*, \tag{13}$$

$$\sum_{j=1}^{N} \beta_j K'(x_j, x_i) - y_i' \leq \varepsilon_1 + \tau_i, \tag{14}$$

$$y_i' - \sum_{j=1}^{N} \beta_j K'(x_j, x_i) \leq \varepsilon_1 + \tau_i^*, \tag{15}$$

$$\xi_i, \xi_i^*, \tau_i, \tau_i^* \geq 0. \tag{16}$$

Here, $\mathbf{K}$ is the kernel matrix with elements $(\mathbf{K})_{ij} = K(x_i, x_j)$, $K'(x_i, x_j)$ is the derivative of the kernel function with respect to the second variable, and ^{T} denotes transpose. Note that now $||\mathbf{w}||^2 = \boldsymbol{\beta}^T \mathbf{K}\boldsymbol{\beta}$.

In order to solve this problem, a Lagrangian functional is introduced leading to a quadratic optimization problem in terms of the Lagrange multipliers

$$\begin{aligned} L(\boldsymbol{\beta}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\tau}, \boldsymbol{\tau}^*, \boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\theta}) &= \frac{1}{2}\boldsymbol{\beta}^T \mathbf{K}\boldsymbol{\beta} + C_1 \sum_{i=1}^{N}(\xi_i + \xi_i^*) + C_2 \sum_{i=1}^{N}(\tau_i + \tau_i^*) \\ - \sum_{i=1}^{N} \alpha_i \left[\varepsilon + \xi_i - \left(\sum_{j=1}^{N} \beta_j K(x_j, x_i) + b - y_i\right)\right] &- \sum_{i=1}^{N} \alpha_i \left[\varepsilon + \xi_i^* - \left(y_i - \sum_{j=1}^{N} \beta_j K(x_j, x_i) + b\right)\right] \\ - \sum_{i=1}^{N} \lambda_i \left[\varepsilon' + \tau_i - \left(\sum_{j=1}^{N} \beta_j K'(x_j, x_i) - y_i'\right)\right] &- \sum_{i=1}^{N} \lambda_i \left[\varepsilon' + \tau_i^* - \left(y_i' - \sum_{j=1}^{N} \beta_j K'(x_j, x_i)\right)\right] \\ &- \sum_{i=1}^{N} \left(\mu_i \xi_i + \mu_i^* \xi_i^* + \theta_i \tau_i + \theta_i^* \tau_i^*\right). \end{aligned} \tag{17}$$

The expression for $\boldsymbol{\beta}$ is obtained by equating the derivative of the Lagrangian (17) to zero, which gives the solution

$$\boldsymbol{\beta} = (\boldsymbol{\alpha}^* - \boldsymbol{\alpha}) + \mathbf{K}^{-1}\mathbf{K}'(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}), \tag{18}$$

where $\mathbf{K}'$ is the matrix with elements $(\mathbf{K}')_{ij} = K'(x_i, x_j)$, and $\boldsymbol{\alpha}^*$, $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}^*$ and $\boldsymbol{\lambda}$ are the Lagrange multipliers vectors. Note that any non-null value on the $(\boldsymbol{\lambda}^* - \boldsymbol{\lambda})$ vector, given the structure of the

$\mathbf{K}$ and $\mathbf{K}'$ matrices, potentially implies that all the $\boldsymbol{\beta}$ coefficients would be non-null. This means that forcing the SV expansion in practice eliminates the property of providing a number of support vectors by allowing to discard the remaining vectors (the vectors associated to zero valued Lagrange multipliers in the conventional SVM framework).

IV. IRWLS ALGORITHM

The Lagrangian (17) can be solved, as usual in the SVM framework [6], finding the dual quadratic constrained optimization problem for the Lagrange multipliers α_i and λ_i, which can be solved by Quadratic Programming (QP) techniques. However, in this case we have used an Iterative Re-Weighted Least Square (IRWLS) procedure [3]. The IRWLS algorithm has shown to require a lower computational burden than conventional QP techniques to solve support vector classifiers [8] and has been recently proven to converge to the SVM solution [9]. The proposed algorithm follows the same basic idea proposed in [3] and followed too in [2]: the Lagrangian is rearranged to group the terms depending on the Lagrange multipliers and is simplified taking into account the KKT conditions. This leads to a functional that can be written as

$$L = \frac{1}{2}\boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta} + \frac{1}{2}\sum_{i=1}^{N} a_i e_i^2 + a_i^*(e_i^*)^2 + \frac{1}{2}\sum_{i=1}^{N} s_i d_i^2 + s_i^*(d_i^*)^2, \tag{19}$$

where

$$e_i = \boldsymbol{\beta}^T K_i + b - y_i - \varepsilon, \quad a_i = \frac{\alpha_i}{e_i}, \tag{20}$$

$$e_i^* = y_i - \boldsymbol{\beta}^T K_i - b - \varepsilon, \quad a_i^* = \frac{\alpha_i^*}{e_i^*}, \tag{21}$$

$$d_i = \boldsymbol{\beta}^T K_i' - y_i' - \varepsilon', \quad s_i = \frac{\lambda_i}{d_i}, \tag{22}$$

$$d_i^* = y_i' - \boldsymbol{\beta}^T K_i' - \varepsilon', \quad s_i^* = \frac{\lambda_i^*}{d_i^*}, \tag{23}$$

$K_i = [K(x_1,x_i),\cdots,K(x_N,x_i)]^T$, and $K_i' = [K'(x_1,x_i),\cdots,K'(x_N,x_i)]^T$. This expression can be seen as a weighted-least-squares functional. Since the weights depend on the previous solution, the system must be iterated until the fixed point solution is reached. At each step, the solution can be obtained equating the derivatives of the functional L with respect to $\boldsymbol{\beta}$ and to b to zero, which leads to a matrix system of two equations. The second equation can be replaced by the simpler constraint $\sum_i(\alpha_i^* - \alpha_i) = 0$, which is obtained from the KKT conditions. The first equation can be written as

$$\left[\mathbf{K}\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{K}^T + \mathbf{K}'\mathbf{D}_{\mathbf{s}+\mathbf{s}^*}\mathbf{K}'^T + \mathbf{K}\right]\boldsymbol{\beta} = \left[\mathbf{K},\mathbf{K}'\right]\begin{bmatrix} \mathbf{D}_{\mathbf{a}+\mathbf{a}^*}(\mathbf{y}-\mathbf{1}b) + \mathbf{D}_{\mathbf{a}-\mathbf{a}^*}\mathbf{1}\varepsilon \\ \mathbf{D}_{\mathbf{s}+\mathbf{s}^*}\mathbf{y}' + \mathbf{D}_{\mathbf{s}-\mathbf{s}^*}\mathbf{1}\varepsilon' \end{bmatrix}. \tag{24}$$

Replacing $\boldsymbol{\beta}$ by

$$\boldsymbol{\beta} = \left[\mathbf{I}, \mathbf{K}^{-1}\mathbf{K}'\right] \begin{bmatrix} \boldsymbol{\alpha}^* - \boldsymbol{\alpha} \\ \boldsymbol{\lambda}^* - \boldsymbol{\lambda} \end{bmatrix}, \qquad (25)$$

pre-multiplying the equation by

$$\left(\left[\mathbf{K}, \mathbf{K}'\right]^T\right)^+ = \left(\begin{bmatrix} \mathbf{K}^T \\ \mathbf{K}'^T \end{bmatrix} \left[\mathbf{K}, \mathbf{K}'\right] \right)^{-1} \begin{bmatrix} \mathbf{K}^T \\ \mathbf{K}'^T \end{bmatrix}, \qquad (26)$$

and introducing the second equation, it is straightforward to see that the system becomes

$$\begin{bmatrix} \mathbf{H} + \begin{bmatrix} \mathbf{D}_{\mathbf{a}+\mathbf{a}^*} & 0 \\ 0 & \mathbf{D}_{\mathbf{s}+\mathbf{s}^*} \end{bmatrix}^{-1}, & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \left[\mathbf{1}^T, \mathbf{0}^T\right] & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^* - \boldsymbol{\alpha} \\ \boldsymbol{\lambda}^* - \boldsymbol{\lambda} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} + \frac{\mathbf{a}-\mathbf{a}^*}{\mathbf{a}+\mathbf{a}^*}\varepsilon \\ \mathbf{y}' + \frac{\mathbf{s}-\mathbf{s}^*}{\mathbf{s}+\mathbf{s}^*}\varepsilon' \\ 0 \end{bmatrix}, \qquad (27)$$

where $\frac{\mathbf{a}-\mathbf{a}^*}{\mathbf{a}+\mathbf{a}^*}$ and $\frac{\mathbf{s}-\mathbf{s}^*}{\mathbf{s}+\mathbf{s}^*}$ denote, respectively, the column-vector containing $(a_i - a_i^*)/(a_i + a_i^*)$ and $(s_i - s_i^*)/(s_i + s_i^*)$ in the i^{th} row, and

$$\mathbf{H} = \begin{bmatrix} \mathbf{K}^T & \mathbf{K}' \\ \mathbf{K}'^T & \mathbf{K}'^T\mathbf{K}^{-1}\mathbf{K}' \end{bmatrix}. \qquad (28)$$

Finally, Table I shows the algorithmic implementation of the IRWLS procedure. The expressions to actualize the coefficients at the fourth step arise straightforward from the KKT conditions (see [3] for details).

V. Experimental Results

To evaluate the accuracy of the proposed method, labeled SVR-dr in the following, we have compared it with the conventional support vector machine for regression (SVR) and with the method proposed in [2], labeled SVR-d. Gaussian kernels of size σ have been employed for all methods. In terms of accuracy, basically the proposed method exhibits the same capability of SVR-d. Therefore, the advantages obtained by including the derivative in the reconstruction, which were shown in [2] and [5], are kept by this method. Here we will remark the main differences obtained between both models and we include the SVR reference as in [2] and [5].

We have selected a set of band-limited functions to test the model: 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, 50 equally spaced sampling points in the range 0-5 have been employed by the SVR (50 samples of the function), and 25 points by the SVR-d and SVR-dr methods

(25 samples of the function + 25 samples of the derivative). In this way, the number of total available data is the same for all methods. 1000 independent noisy experiments have been considered, with the same signal to noise ratio (SNR) 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 as a function of the kernel size σ. Results are provided for two different SNR values: 30 and 10 dB, respectively. The optimal parameters ε and C_1 were obtained by cross-validation for each method. In this case, $\varepsilon' = \pi\varepsilon$ and $C_2 = C_1/\pi$ have 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). This leaves only three free parameters for all methods, ε, C_1 and σ. For SNR=30 dB, the methods including the derivative provide better results than the SVR method. In this case, the only difference between the SNR-d and the proposed method is that this exhibits a slightly higher sensitivity to the selection of the kernel size. For SNR=10 dB, the 3 methods provide a similar capability of approximation and the sensitivity to the choice of σ is not noticeable now.

Figure 2 shows the SER in the reconstruction of the derivative. In this case the methods including the derivative clearly outperform SVR for both SNR's. Again the SVR-dr shows a slightly higher sensitivity to σ than the SVR-d for the high SNR, while for 10 dB the performance curves practically overlap.

To compare the complexity of the models, we have looked at the number of support vectors for each method. In this case, the proposed method has a fixed number of support vectors, which is given by the sampling points, 25 in this example. Results over 1000 independent test signals have been averaged. Different parameters ε, C_1 and σ (composing a multidimensional grid) have been tested for each signal. Table II compares the minimum number of support vectors necessary to achieve a given SER in the reconstruction of the function and of the derivative (this can be seen as the number of support vectors for the optimal parameter choice). The table shows that a trade-off between number of support vectors and accuracy can be done. Although SVR and SVR-d are able to produce reasonable results with less support vectors than SVR-dr, it can be seen that for the higher accuracy the number of required support vector is greater than 25, the number of fixed support vectors for SVR-dr.

Moreover, this results rely in an optimal parameter selection. Table III compares the mean number of support vectors obtained with all the parameter choices that gave a SER into a given interval. In this case, the mean number of necessary support vectors is clearly higher.

VI. DISCUSSION

A new method for regression of a function and its derivative has been presented. The proposed approach provides a compact kernel-based expansion. We have presented an IRWLS algorithm to solve the constrained optimization problem proposed to penalize errors in the reconstruction of both the function and the derivative. By introducing the information of the derivative, we have shown that the accuracy of conventional support vector regression is increased, especially in the reconstruction of the derivative, as happened with the expansion presented in [2].

When comparing it with the method in [2], the proposed method has a higher training load (basically because of the evaluation of $\boldsymbol{\beta}$ by (25) at each step). However, this is not a problem in most of applications involving regression, which usually are solved off-line. For off-line applications the main computational constraints appear at evaluation time, and therefore the more compact expansion is a clear advantage. Moreover, it also can reduce the number of parameters (support vectors) of the representation when high accuracy is necessary, as shown by the presented experiments, thus reducing again the computational complexity of evaluation.

Although we have only considered the one-dimensional and one derivative case, the extension of the method to higher order derivatives and higher dimensional input spaces is straightforward.

REFERENCES

[1] A. M. Crosmun and S. A. Maas, "Minimization of intermodulation distortion in GaAs MESFET small-signal amplifiers," *IEEE Transactions Microwave Theory and Techniques*, vol. 37, no. 9, pp. 1411–1417, 1989.

[2] M. Lázaro, I. Santamaría, F. Pérez-Cruz, and A. Artés-Rodríguez, "SVM for the simultaneous approximation of a function and its derivative," in *Proceedings of the 2003 IEEE International Workshop on Neural Networks for Signal Processing (NNSP)*, Toulouse, France, 2003.

[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] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[5] M. Lázaro, I. Santamaría, F. Pérez-Cruz, and A. Artés-Rodríguez, "Support vector regression for the simultaneous learning of a multivariate function and its derivatives," *Neurocomputing, Special Issue on NNSP03,* Submitted 2004.

[6] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[7] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, pp. 551–560, 1990.

[8] F. Pérez-Cruz, P. L. Alarcón-Diana, A. Navia-Vázquez, and A. Artés-Rodríguez, "Fast training of support vector classifiers," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: M.I.T. Press, Nov. 2000.

[9] F. Pérez-Cruz, C. Bousoño-Calzón, and A. Artés-Rodríguez, "Convergence of the IRWLS procedure to the support vector machine solution," *Neural Computation*, no. Accepted for Publication, 2004.

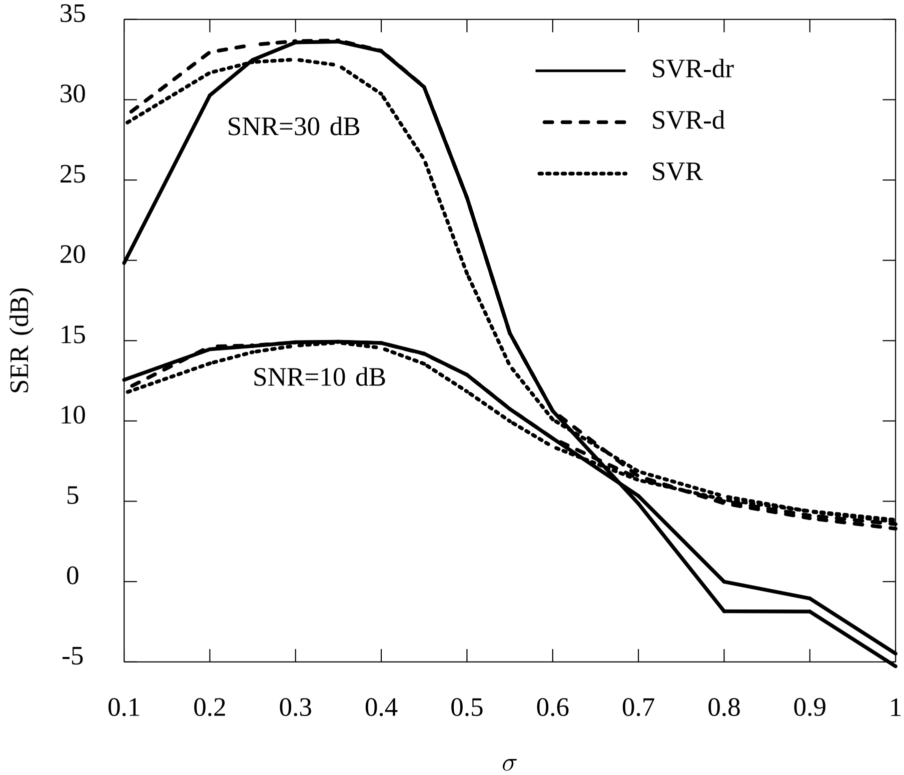

Fig. 1. SER (dB) in the reconstruction of the function as a function of the kernel size

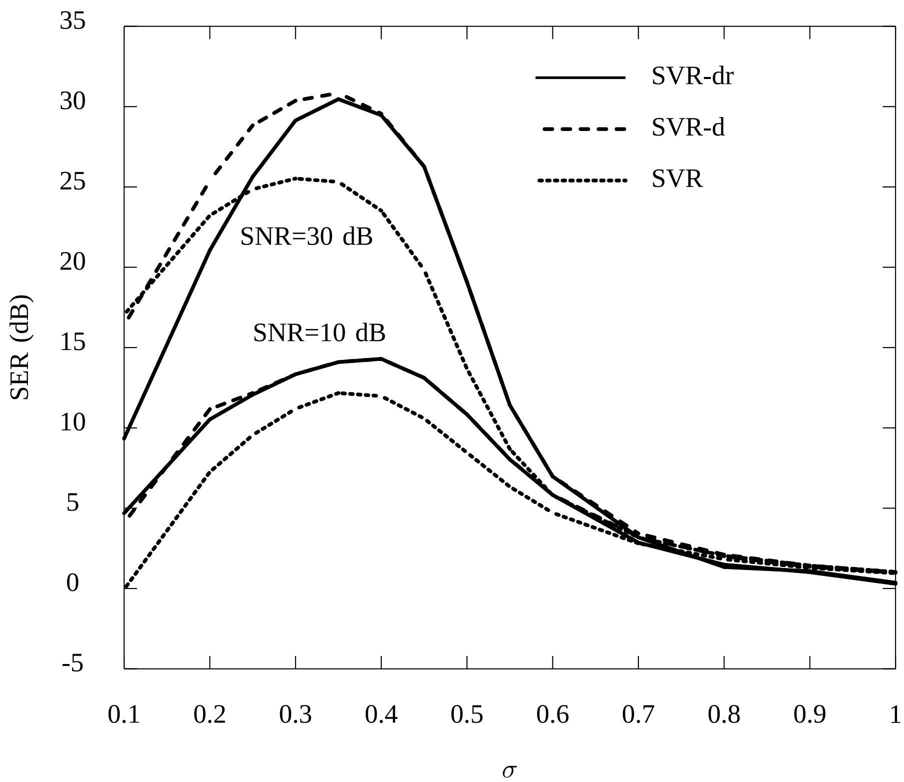

Fig. 2. SER (dB) in the reconstruction of the derivative as a function of the kernel size

1) Initialization:
   - Compute $\mathbf{H}$ (from $\mathbf{K}$, $\mathbf{K}'$)
   - $a_i = C_1,\ s_i = C_2$ for odd i; $a_i^* = C_1,\ s_i^* = C_2$ for even i.
2) Solve (27) to obtain $\boldsymbol{\alpha}^* - \boldsymbol{\alpha}$ and $\boldsymbol{\lambda}^* - \boldsymbol{\lambda}$.
3) Evaluate

$$\boldsymbol{\beta} = \left[\mathbf{I}, \mathbf{K}^{-1}\mathbf{K}'\right] \begin{bmatrix} \boldsymbol{\alpha}^* - \boldsymbol{\alpha} \\ \boldsymbol{\lambda}^* - \boldsymbol{\lambda} \end{bmatrix}$$

$$\mathbf{e} = \mathbf{K}^T\boldsymbol{\beta} + \mathbf{1}b - \mathbf{y} - \mathbf{1}\varepsilon, \qquad \mathbf{e}^* = \mathbf{y} - \mathbf{K}^T\boldsymbol{\beta} - \mathbf{1}b - \mathbf{1}\varepsilon$$
$$\mathbf{d} = \mathbf{K}'^T\boldsymbol{\beta} - \mathbf{y}' - \mathbf{1}\varepsilon', \qquad \mathbf{d}^* = \mathbf{y}' - \mathbf{K}'^T\boldsymbol{\beta} - \mathbf{1}\varepsilon'$$

4) Recalculate a_i, a_i^*, s_i^* and s_i^*

$$a_i = \frac{C_1}{e_i},\ a_i^* = \frac{C_1}{e_i^*},\ s_i = \frac{C_2}{d_i},\ d_i^* = \frac{C_2}{d_i^*}$$

5) Go to step 2 until convergence is achieved.

TABLE I

IRWLS ALGORITHM PSEUDOCODE.

	SER for function				SER for derivative			
Model	20 dB	25 dB	30 dB	33 dB	20 dB	24 dB	26 dB	30 dB
SVR	15	18	27	–	18	27	–	–
SVR-d	17	19	29	29	17	19	19	29

TABLE II

MINIMUM NUMBER OF SUPPORT VECTORS NECESSARY TO ACHIEVE A GIVEN SER.

	SER for function				SER for derivative			
Model	20-25 dB	25-30 dB	30-33 dB	More than 33 dB	20-24 dB	24-26 dB	26-30 dB	More than 30 dB
SVR	25	28	39	–	29	39	–	–
SVR-d	28	29	38	40	20	30	37	39

TABLE III

MEAN NUMBER OF SUPPORT VECTORS OBTAINED WITH ALL THE PARAMETER CHOICES THAT GAVE A SER INTO A GIVEN INTERVAL