A NEW OPTIMIZING PROCEDURE FOR ν-SUPPORT VECTOR REGRESSOR

Fernando Pérez-Cruz

Universidad de Alcalá. Dpto. Teoría de la Señal y Comunicaciones. Campus Universitario Alcala de Henares 28871, (Madrid) SPAIN. fernando.perezc@uah.es

Antonio Artés-Rodríguez

Universidad Carlos III Madrid. Dpto. de Tecnologías de las comunicaciones Avda. Universidad 30, Leganes 28911 (Madrid) SPAIN antonio@ieee.org

ABSTRACT

We present a novel approach to solve the ν-SVR. It is based on an Iterative Re-Weighted Least Squares (IRWLS) procedure, which is simple to implement and can be tuned to the usual ν-SVR solution. The IRWLS procedure is much more efficient (computational load) than Quadratic Programming techniques, which are usually employed to solve it.

1. INTRODUCTION

The Support Vector Machine (SVM) is a state-of-the-art tool to solve linear and nonlinear regression and classification problems [1]. The Support Vector Regressor (SVR) has not been used as widely as its classification counterpart (SVC), because its parameters are hard to set and because its training time is prohibitively large. The ν-SVR [2] is a different implementation of the SVR which allows to easily tune the SVR parameters, simplifying its use.

In this communication we will lead with the other ν-SVR limitation, its computational complexity, which limits its applicability for solving signal processing applications. The SVM is usually solved using Quadratic Programming (QP) procedures that are hard to implement, present several numerical problems and high computational load [3]. In this communication we propose to solve the ν-SVR by iteratively solving weighted least square problems. The IRWLS procedure has been successfully applied to solve the SVC [4] and the SVR [5].

The outline of the paper is as follows. The ν-SVR is introduced in Section 2. The development of the IRWLS is done in Section 3, together with its algorithmic implementation. Computer experiments are shown in Section 4. We end with some concluding remarks in section 5.

2. ν-SUPPORT VECTOR REGRESSOR

The ν-SVR, given a labeled training data set $((\mathbf{x}_1, y_1), \dots, \mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$) and a nonlinear transformation to a higher dimensional space ($\boldsymbol{\phi}(\cdot)$, $\mathbb{R}^d \xrightarrow{\phi(\cdot)} \mathbb{R}^H$ and $d \le H$), needs to solve the following optimizing problem, in order to find the regression function between $\mathbf{x}$ and y,

$$\min_{\mathbf{w},b,\xi_i,\xi_i^*,\varepsilon} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i(\xi_i + \xi_i^*) + Cn\nu\varepsilon \quad (1)$$

subject to

$$y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b \le \varepsilon + \xi_i \qquad \forall i = 1, \dots, n \quad (2)$$

$$\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b - y_i \le \varepsilon + \xi_i^* \qquad \forall i = 1, \dots, n \quad (3)$$

$$\xi_i, \xi_i^* \ge 0 \qquad \forall i = 1, \dots, n \quad (4)$$

$$\varepsilon \ge 0 \quad (5)$$

where ξ_i and ξ_i^{*}[1] are positive slack variables, introduced to deal with prediction error greater than ε, and C is the penalization of such deviations.

The usual SVR optimizes (1) (without the last term) subject to (2), (3) and (4). The ν-SVR introduces the value of ε into the minimizing functional, penalizing its growth by a factor of $Cn\nu$. This factor is chosen because the value of ν gives a lower limit in the expected fraction of Support Vectors [2], being $\nu \in [0, 1]$.

In order to solve the ν-SVR, the linear restrictions are introduced in the minimizing functional, making use of the Lagrange multipliers. We then need to minimize

$$\begin{aligned}L_P = {}& \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i(\xi_i + \xi_i^*) + Cn\nu\varepsilon \\ & -\sum_i \alpha_i(\varepsilon + \xi_i - y_i + \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) - \sum_i(\mu_i\xi_i + \mu_i^*\xi_i^*) - \\ & - \sum_i \alpha_i^*(\varepsilon + \xi_i^* + y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b) - \lambda\varepsilon \end{aligned} \quad (6)$$

with respect to $\mathbf{w}$, b, $\xi_i^{(*)}$ and ε and maximize it with respect to the Lagrange multipliers, $\alpha_i^{(*)}$, $\mu_i^{(*)}$ and λ.

[1] For short, we will refer to ξ_i and ξ_i^* as $\xi_i^{(*)}$, as well as for the others varibles with and without $*$.

The solution to this problem is unique because it is a convex optimization problem linearly restricted. Its solution is determined by the Kuhn-Tucker (KT) Theorem [6]. This Theorem imposes the following conditions to the solution of (6) (The KT conditions), namely: (2)-(5), and

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_i (\alpha_i - \alpha_i^*)\boldsymbol{\phi}(\mathbf{x}_i) = 0 \tag{7}$$

$$\frac{\partial L_P}{\partial b} = \sum_i (\alpha_i^* - \alpha_i) = 0 \tag{8}$$

$$\frac{\partial L_P}{\partial \xi_i^{(*)}} = C - \alpha_i^{(*)} - \mu_i^{(*)} = 0 \quad \forall i \tag{9}$$

$$\frac{\partial L_P}{\partial \varepsilon} = Cn\nu - \sum_i (\alpha_i + \alpha_i^*) - \lambda = 0 \tag{10}$$

$$\alpha_i^{(*)}, \mu_i^{(*)}, \lambda \geq 0 \quad \forall i \tag{11}$$

$$\alpha_i\{\xi_i + \varepsilon - y_i + \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b\} = 0 \quad \forall i \tag{12}$$

$$\alpha_i^*\{\xi_i^* + \varepsilon + y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b\} = 0 \quad \forall i \tag{13}$$

$$\mu_i^{(*)}\xi_i^{(*)} = 0 \quad \forall i \tag{14}$$

$$\lambda\varepsilon = 0 \tag{15}$$

The usual ν-SVR resolution introduces the KT conditions (7)-(10) into (6), leading to a maximizing functional, that can be solved using QP procedures [2].

3. ITERATIVE RE-WEIGHTED LEAST SQUARES

We will now follow a different path to solve the ν-SVR. Our aim is to transform the functional (6) into a weighted least square one, in which the weights depend on the solution ($\mathbf{w}$, b and ε). This least square problem is iteratively solved to reach the ν-SVR solution. In order to do so, we rearrange (6) collecting the terms depending on $\xi_i^{(*)}$ in two independent sums

$$\begin{aligned} L_P = {} & \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i \alpha_i(y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b - \varepsilon) \\ & + \sum_i \alpha_i^*(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b - y_i - \varepsilon) + \varepsilon(Cn\nu - \lambda) \\ & + \sum_i \xi_i(C - \alpha_i - \mu_i) + \sum_i \xi_i^*(C - \alpha_i^* - \mu_i^*) \end{aligned}$$

which can be simplified, if the KT conditions in (9) are replaced into it, reaching

$$\begin{aligned} L'_P = {} & \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i \alpha_i(y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b - \varepsilon) + \\ & + \sum_i \alpha_i^*(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b - y_i - \varepsilon) + \varepsilon(Cn\nu - \lambda) = \\ = {} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\sum_i (a_i e_i^2 + a_i^*(e_i^*)^2) + \varepsilon(Cn\nu - \lambda) \end{aligned} \tag{16}$$

where we have defined

$$e_i = y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b - \varepsilon \tag{17}$$

$$e_i^* = \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b - y_i - \varepsilon \tag{18}$$

$$a_i = \frac{2\alpha_i}{y_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - b - \varepsilon} = \frac{2\alpha_i}{e_i} \tag{19}$$

$$a_i^* = \frac{2\alpha_i^*}{\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b - y_i - \varepsilon} = \frac{2\alpha_i^*}{e_i^*} \tag{20}$$

$e_i^{(*)}$ measure the deviation between the actual output (y_i) and the regressor output ($\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b$), and $a_i^{(*)}$ are the weights associated to each error. Due to $a_i^{(*)} = a_i^{(*)}(e_i^{(*)})$, we will use a Iterative Re-Weighted Least Square (IRWLS) procedure [7] to solve (16), which consists in:

1. Minimizing (16) with respect to $\mathbf{w}$, b and ε, considering the $a_i^{(*)}$ fixed.
2. Recalculating $a_i^{(*)}$ with $\mathbf{w}$, b and ε from step one.
3. Repeating until convergence.

The minimum of (16) with respect to $\mathbf{w}$, b and ε, considering fixed $a_i^{(*)}$, is given by:

$$\begin{aligned} \frac{\partial L'_P}{\partial \mathbf{w}} = {} & \mathbf{w} - \boldsymbol{\Phi}^T\mathbf{D_a}[\mathbf{y} - \boldsymbol{\Phi}\mathbf{w} - \mathbf{1}(b + \varepsilon)] + \\ & + \boldsymbol{\Phi}^T\mathbf{D_{a^*}}[\boldsymbol{\Phi}\mathbf{w} + \mathbf{1}(b - \varepsilon) - \mathbf{y}] = \mathbf{0} \end{aligned} \tag{21}$$

$$\begin{aligned} \frac{\partial L'_P}{\partial b} = {} & -\mathbf{a}^T[\mathbf{y} - \boldsymbol{\Phi}\mathbf{w} - \mathbf{1}(b + \varepsilon)] + \\ & + (\mathbf{a}^*)^T[\boldsymbol{\Phi}\mathbf{w} + \mathbf{1}(b - \varepsilon) - \mathbf{y}] = 0 \end{aligned} \tag{22}$$

$$\begin{aligned} \frac{\partial L'_P}{\partial \varepsilon} = {} & Cn\nu - \lambda - \mathbf{a}^T[\mathbf{y} - \boldsymbol{\Phi}\mathbf{w} - \mathbf{1}(b + \varepsilon)] - \\ & - (\mathbf{a}^*)^T[\boldsymbol{\Phi}\mathbf{w} + \mathbf{1}(b - \varepsilon) - \mathbf{y}] = 0 \end{aligned} \tag{23}$$

where

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \dots \boldsymbol{\phi}(\mathbf{x}_n)]^T$$

$$\mathbf{a}^{(*)} = [a_1^{(*)}, a_2^{(*)} \dots, a_n^{(*)}]^T$$

$$(\mathbf{D}_{\mathbf{a}^{(*)}})_{ij} = a_{ij}^{(*)}\delta[i - j] \qquad \forall i, j = 1, \dots, n$$

In order to recalculate $a_i^{(*)}$, we use the KT conditions (9), (11)-(14), which must hold at the solution, leading to

$$a_i^{(*)} = \begin{cases} 0, & e_i^{(*)} < 0 \\ \dfrac{2C}{e_i^{(*)}}, & e_i^{(*)} \geq 0 \end{cases} \tag{24}$$

where we have used the following relation between $e_i^{(*)}$ and $\xi_i^{(*)}$

$$\xi_i^{(*)} = \begin{cases} 0, & e_i^{(*)} < 0 \\ e_i^{(*)}, & e_i^{(*)} \geq 0 \end{cases}$$

1. Initialization:
   Compute $\mathbf{H}$ as in (27). Set $S2 = \varnothing$ and $S3 = \varnothing$. And $G_b = 0$, $G_\nu = 0$, $\mathbf{G}_{13} = \mathbf{0}$ and $\mathbf{a}_{S1} = \mathbf{2}C$.

2. Solve:
$$\begin{bmatrix} \mathbf{H}_{S1,S1} + \mathbf{D}^{-1}_{(\mathbf{a}_{S1}+\mathbf{a}^*_{S1})} & \mathbf{1} & \mathbf{E}_{S1} \\ \mathbf{1}^T & 0 & 0 \\ \mathbf{E}^T_{S1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{S1} \\ b \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \\ Cn\nu - \lambda \end{bmatrix} - \begin{bmatrix} \mathbf{G}_{13} \\ G_b \\ G_\nu \end{bmatrix}$$

3. Compute: $\mathbf{e}^{(*)}$ as in (17) and (18).

4. Compute: $\mathbf{a}^{(*)}$ as in (24).

5. Reorder sets:
   (a) Move every β_i form S1 with $e_i^{(*)} < 0$ to S2.
   (b) Move every $\beta_i = \pm C$ form S1 to S3.
   (c) Move every β_i form S2 with $e_i^{(*)} < 0$ to S2.
   (d) Move every β_i form S2 with $e_i > 0$ or $e_i^* > 0$ to S2.

6. Compute: $\mathbf{G}_{13} = \mathbf{H}_{S1,S3}\boldsymbol{\beta}_{S3}$, $G_b = \mathbf{1}^T\boldsymbol{\beta}_{S3}$ $G_\nu = \mathbf{E}^T_{S3}\boldsymbol{\beta}_{S3}$ and $\boldsymbol{\beta}_{S2} = 0$

7. Back to 2 until there are no further changes in $\boldsymbol{\beta}$, b or ε.

Table 1. IRWLS-ν-SVR Procedure.

3.1. Reproducing Kernels Hilbert Space

The SVM is not regularly solved using the nonlinear mapping ($\boldsymbol{\phi}(\cdot)$), which in most cases is unknown. In the QP functional, the nonlinear transformation appears as an inner product with each other ($\boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$), so we just need to specify the Reproducing Kernel Hilbert Space (RKHS) $K(\cdot,\cdot)$, not the nonlinear mapping $\boldsymbol{\phi}(\cdot)$. In order to solve the ν-SVR with the RKHS, we force $\mathbf{w}$ to be

$$\mathbf{w} = \sum_i \beta_i \boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\Phi}^T\boldsymbol{\beta} \tag{25}$$

Once (25) has been replaced into (21), (22) and (23), we obtain the equation system to be solved in the first step of the IRWLS procedure:

$$\begin{bmatrix} \mathbf{H} + \mathbf{D}^{-1}_{(\mathbf{a}+\mathbf{a}^*)} & \mathbf{1} & \mathbf{E} \\ \mathbf{1}^T & 0 & 0 \\ \mathbf{E}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \\ Cn\nu - \lambda \end{bmatrix} \tag{26}$$

where

$$(\mathbf{H})_{ij} = \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{27}$$

$$\mathbf{E} = \left[\frac{a_1 - a_1^*}{a_1 + a_1^*}, \dots, \frac{a_n - a_n^*}{a_n + a_n^*}\right]^T \tag{28}$$

which does not depend on $\boldsymbol{\phi}(\cdot)$. The necessary algebraic transformations to obtain (26) from (21), (22), (23) and (25) are detailed in the Appendix.

In the ν-SVR solution most β_i are either zero or $\pm C$, so if we are able to detect which are those β_i, they can be dropped form the linear equation system, simplifying its resolution. In Table 1, we show an algorithmic implementation of the IRWLS procedure for the ν-SVR in which the training samples are divided in three sets. The first set (S1) contains the β_i that are neither zero nor $\pm C$, the second one (S2) contains the β_i that are zero, and the third set (S3) contains the β_i that are $\pm C$. In order to form these sets, we must recall that the sample that its corresponding e_i and e_i^* are negative, must present a zero β_i. And the samples, whose e_i or e_i^* are positive and its β_i is equal to $\pm C$, must be in S3. The other samples must lie in S1.

4. COMPUTER EXPERIMENTS

In order to try out the proposed procedure we have solved a prediction problem obtained from the *"UCI Machine Learning Repository"* (http://www.ics.uci.edu/˜mlearn/ MLSummary.html), where we need to estimate the cost of a house in the Boston metropolitan area from 13 attributes. This data set contains 506 instances and has been used to show the insensitivity in the election of ν in [2]. We have preprocessed it so every attribute presents zero mean and unity standard deviation. We have used this data set to compare the training time between the IRWLS-ν-SVR procedure in Table 1 against the QP-ν-SVR. To solve the QP-ν-SVR we have used Matlab's `qp.m`. We have plotted the the training time mean value for 20 independent trials in Figure 1 for both procedures. We have used an RBF RKHS, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{2\sigma}\right)$, and have set $\sigma = 8$, $C = 1$ and $\nu = 0.5$.

In this figure one can understand that the IRWLS procedure is significantly faster than the QP procedure (for n=140 the IRWLS is over 300 times faster than the QP). This advantage is due to the IRWLS only needs to compute the value of β_i for a few training samples and the QP procedures needs to operate with all of them.

5. CONCLUSIONS

We have presented an IRWLS procedure to solve the ν-SVR, whose computational load is much lower than the

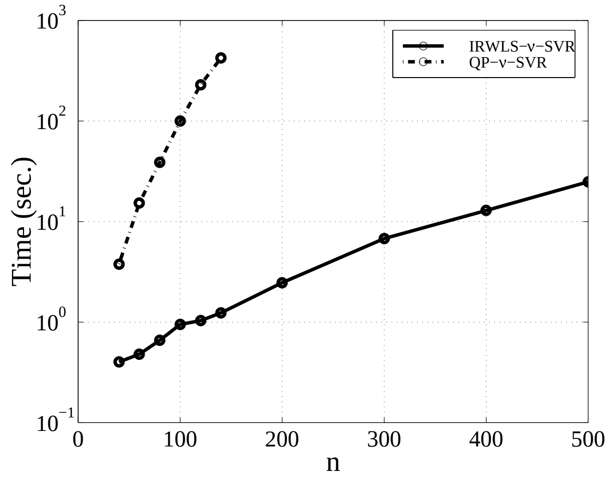

Fig. 1. The Training time for both procedures as a function of the training set size.

QP one. Moreover the proposed procedure is easily implemented, so we have been able to use the ν-SVR solution (most β_i are either 0 or $\pm C$) in order to reduce its computational load.

The IRWLS procedure will now help to use the ν-SVR in signal processing applications where a non-parametric tool as the SVM might be desirable, such as time series prediction.

APPENDIX

In order to obtain (26) from (21), (22), (23) and (25). We first need to substitute (25) into (21) and multiply it by the pseudo-inverse of $\mathbf{\Phi}^T$, obtaining

$$(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}[\mathbf{I}+\mathbf{\Phi}^T\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{\Phi}]\mathbf{\Phi}^T\boldsymbol{\beta} = \\ = \mathbf{D}_{\mathbf{a}+\mathbf{a}^*}[\mathbf{y}-\mathbf{1}b]-\mathbf{D}_{\mathbf{a}-\mathbf{a}^*}\mathbf{1}\varepsilon \quad (29)$$

Using the definition of $\mathbf{H}$ in (27) and multiplying (29) by the inverse of $\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}$, we reach

$$(\mathbf{D}_{\mathbf{a}+\mathbf{a}^*})^{-1}\mathbf{H}^{-1}[\mathbf{H}+\mathbf{H}\mathbf{D}_{\mathbf{a}+\mathbf{a}^*}\mathbf{H}]\boldsymbol{\beta} = \\ = [\mathbf{y}-\mathbf{1}b]-(\mathbf{D}_{\mathbf{a}+\mathbf{a}^*})^{-1}\mathbf{D}_{\mathbf{a}-\mathbf{a}^*}\mathbf{1}\varepsilon \quad (30)$$

Once it has been simplified, it can be expressed as

$$\begin{bmatrix} \mathbf{H}+(\mathbf{D}_{\mathbf{a}+\mathbf{a}^*})^{-1} & \mathbf{1} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \\ \varepsilon \end{bmatrix} = \mathbf{y} \quad (31)$$

Secondly, we rearrange (22) as

$$\begin{bmatrix} (\mathbf{a}+\mathbf{a}^*)^T\mathbf{H} & (\mathbf{a}+\mathbf{a}^*)^T\mathbf{1} & (\mathbf{a}-\mathbf{a}^*)^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \\ \varepsilon \end{bmatrix} = \\ = (\mathbf{a}+\mathbf{a}^*)^T\mathbf{y} \quad (32)$$

and finally, (23) can be transformed into:

$$\begin{bmatrix} (\mathbf{a}-\mathbf{a}^*)^T\mathbf{H} & (\mathbf{a}-\mathbf{a}^*)^T\mathbf{1} & (\mathbf{a}+\mathbf{a}^*)^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \\ \varepsilon \end{bmatrix} = \\ = (\mathbf{a}-\mathbf{a}^*)^T\mathbf{y} - Cn\nu + \lambda \quad (33)$$

In order to reach (32) and (33), we have replaced $\mathbf{w}$ for (25) and we must recall that $\mathbf{H} = \mathbf{\Phi}\mathbf{\Phi}^T$ (see (27)).

When we join together (31), (32) and (33), we do not reach (26). In order to obtain (26), further simplifications are needed. We will replace (32) and (33) by two KT conditions that will reduce the complexity of the equation system to be solved in the first step of the IRWLS procedure.

The relationship between α_i and α_i^* with β_i can be obtained from (25) and the KT condition in (7), being $\beta_i = \alpha_i - \alpha_i^*$, because at the solution the KT conditions must hold. The KT conditions in (8) and (10) can be also imposed over $\boldsymbol{\beta}$, leading, respectively, to

$$\sum_i \beta_i = \mathbf{1}^T\boldsymbol{\beta} = 0 \quad (34)$$

$$\mathbf{1}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) = \mathbf{E}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) = \mathbf{E}^T\boldsymbol{\beta} = Cn\nu - \lambda \quad (35)$$

which can be joined in a linear equation system with (31), reaching (26).

6. REFERENCES

[1] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[2] B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett, "New support vector algorithms," Tech. Rep. NC-TR-98-031, 1998, To appear in *Neural Compt.*.

[3] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 2, pp. 121–167, 1998.

[4] F. Pérez-Cruz, A. Navia-Vázquez, J. L. Rojo-Álvarez, and A. Artés-Rodríguez, "A new training algorithm for support vector machines," in *Proc. of the* 5^{th} *Bayona Workshop on Emerging Tech. in Telecom.*, Baiona, Spain, Sept. 1999, pp. 116–120.

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

[6] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, second edition, 1987.

[7] P. W. Holland and R. E. Welch, "Robust regression using iterative re-weighted least squares," *Comm. of Stat. Theory Methods*, vol. A6, no. 9, pp. 813–27, 1977.