Multi-dimensional Function Approximation And Regression Estimation

Fernando Pérez-Cruz[1], Gustavo Camps-Valls[2], Emilio Soria-Olivas[2], Juan José Pérez-Ruixo[3], Aníbal R. Figueiras-Vidal and Antonio Artés-Rodríguez[1]*

[1] University Carlos III, Department of Signal Theory and Communications
Avda. Universidad 30, 28911 Leganés (Madrid) SPAIN
fernandop@ieee.org
[2] Grup de Processament digital de Senyals, Universitat de València.
[3] Servei de Farmàcia. Hospital Universitari Dr. Peset, València.

Abstract. In this communication, we generalize the Support Vector Machines (SVM) for regression estimation and function approximation to multi-dimensional problems. We propose a multi-dimensional Support Vector Regressor (MSVR) that uses a cost function with a hyperspherical insensitive zone, capable of obtaining better predictions than using an SVM independently for each dimension. The resolution of the MSVR is achieved by an iterative procedure over the Karush-Kuhn-Tucker conditions. The proposed algorithm is illustrated by computers experiments.

1 Introduction

The regression estimation problem, given two variables $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$, that are connected by a functional dependency or, in its more general form, by a probability density function $p(\mathbf{x}, y)$, is defined as the mathematical expectation of y given $\mathbf{x}$ [5], which can be obtained by the $\omega^* \in \Omega$ that minimizes:

$$R_{emp}(\omega) = \frac{1}{n} \sum_{i=1}^{n} \|y_i - r(\mathbf{x}_i, \omega)\|^2 \tag{1}$$

which is the empirical functional to be minimized if $p(\mathbf{x}, y)$ is unknown instead a set of i.i.d. samples are sampled from it [5].

The multi-dimensional regression estimation problem arises when the dependent variable is a vector $\mathbf{y} \in \mathbb{R}^k$ instead of a scalar quantity. This problem appears in many situations in which the output variables are coupled, such as econometric and biomedical time series prediction, and data imputation. The multi-dimensional regression problem can be divided into k 1D problems, and in some frameworks, like minimum variance estimation is equivalent to the multi-dimensional estimate. But, for SVM this is not the case, due to the insensitive zone defined around the estimate will not equally treat every training sample.

* This work has been partially supported by CICYT grant TIC2000-0380-C03-03.

In this paper we propose a novel approach for SVM regression in which a hyper-spherical insensitive zone is defined around the estimate, following the Euclidean norm the regression estimation problem defines [5]. This hyper-spherical insensitive zone will allow us to equally treat every sample, being them penalized by the same factor if they lie outside this insensitive zone. The SVM for multi-regression problem can not be solved using quadratic programming, and thus an iterative procedure similar to the one in [2] is used.

The rest of the paper is outlined as follows. In section 2, we state the multi-dimensional regression estimation problem and we formulate it as a constrained optimization problem. In section 3, we show that the Multidimensional Support Vector Regressor (MSVR) can be solved using an iterative procedure. We present some simulations results in Section 4. The paper ends with some concluding remarks in Section 5.

2 Multi-dimensional Support Vector Regressor

The SVR for regression estimation defines an insensitive zone around the estimate, if a sample is further than ε than the estimate is penalized by C and is not otherwise [4]. If the 1D SVR is applied over each direction of the multi-dimensional problem, we will have samples as close as ε that will be penalized and samples as far as $\sqrt{k}\varepsilon$, in Euclidean norm, that will not. Moreover, the samples will not suffer an equal penalty, because if the sample is further than ε in every direction will be penalized by $k \times C$ and only by C if it is only in one direction. These limitations are illustrated for a 2D problem in Figure 1a, in which the square represents the insensitive zone for the two 1D-SVR. The ideal result would be the one pictured by the circle in Figure 1a, in which the samples inside will not be penalized and the samples further that ε_2 from $r(\mathbf{x})$ will.

The Multi-dimensional SVR (MSVR) with a hyper-spherical insensitive zone can then be similarly stated as the regular SVR. The MSVR, given a labeled training data set $((\mathbf{x}_i, y_i)\ \forall i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R})$ and a nonlinear transformation to a higher dimensional space ($\boldsymbol{\phi}(\cdot)$, $\mathbb{R}^d \xrightarrow{\boldsymbol{\phi}(\cdot)} \mathbb{R}^H$ and $d \leq H$), solves:

$$\min_{\mathbf{w}^j, b^j, \xi_i} \sum_{j=1}^{k} \|\mathbf{w}^j\|^2 + C \sum_{i=1}^{n} \xi_i \tag{2}$$

subject to

$$\|\mathbf{y}_i - \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{b}\|^2 \leq \varepsilon + \xi_i \qquad \forall i = 1, \ldots, n \tag{3}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{4}$$

where $\mathbf{W} = [\mathbf{w}^1, \ldots, \mathbf{w}^k]^T$ and $\mathbf{b} = [b^1, \ldots, b^k]^T$ define the k-dimensional linear regressor in the H-dimensional feature space.

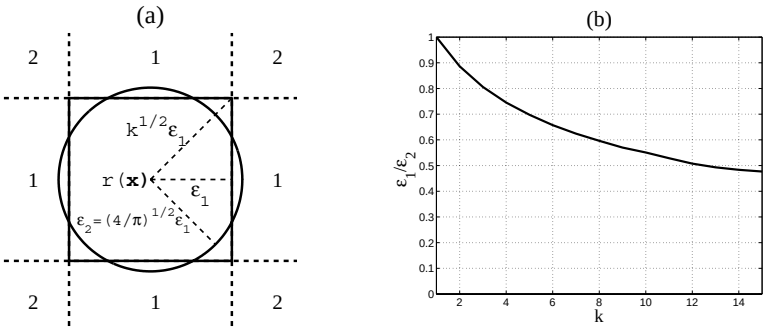

Fig. 1. A hyper-cubic and a hyper-spherical insensitive zones are shown in (a) for a bi-dimensional regression estimation problem, in which they have equal areas. The areas marked with a 1 will be penalized just once and the samples in 2 will be penalized twice. In (b) we show the ratio between ε_1 and ε_2, respectively, the values of the hyper-cubic and hyper-spherical insensitive zone, as a function of k, in order to assure equal hyper-volume.

3 Resolution of the MSVR

In order to solve the MSVR, we first introduce the constrains (3) and (4) into (2), making use of the Lagrange multipliers, leading to the minimization of

$$L_P = \sum_{j=1}^{k} \|\mathbf{w}^j\|^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (\varepsilon + \xi_i - \|\mathbf{y}_i - \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{b}\|^2) - \sum_{i=1}^{n} \mu_i \xi_i \quad (5)$$

with respect to $\mathbf{w}^j$, b^j and ξ_i and its maximization with respect the Lagrange multipliers, α_i and μ_i. The solution to this problem is given by the Karush-Kuhn-Tuker Theorem [1], that states the following conditions (KKT conditions); namely (3), (4), and

$$\frac{\partial L_P}{\partial \mathbf{w}^j} = 2\mathbf{w}^j - 2\boldsymbol{\Phi}^T \mathbf{D}_{\boldsymbol{\alpha}} [\mathbf{y}^j - \boldsymbol{\Phi}\mathbf{w}^j - \mathbf{1}b^j] = \mathbf{0} \qquad \forall j = 1, \ldots, k \quad (6)$$

$$\frac{\partial L_P}{\partial b^j} = \boldsymbol{\alpha}^T [\mathbf{y}^j - \boldsymbol{\Phi}\mathbf{w}^j - \mathbf{1}b^j] = 0 \qquad \forall j = 1, \ldots, k \quad (7)$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \mu_i - \alpha_i = 0 \qquad \forall i = 1, \ldots, n \quad (8)$$

$$\mu_i, \alpha_i \geq 0 \qquad \forall i = 1, \ldots, n \quad (9)$$

$$\alpha_i \{\varepsilon + \xi_i - \|\mathbf{y}_i - \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{b}\|^2\} = 0 \qquad \forall i = 1, \ldots, n \quad (10)$$

$$\mu_i \xi_i = 0 \qquad \forall i = 1, \ldots, n \quad (11)$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^T$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$, $\mathbf{y}^j = [y_{1j}, y_{2j}, \ldots, y_{nj}]^T$ and $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{ik}]^T$. The MSVR can not be solved as the regular SVR, since the KKT conditions can not be replaced into the functional in (5) to obtain a quadratic problem linearly restricted. We can instead use an iterative

algorithm, similar to the one proposed to solve the SVR in [2]. This procedure first obtains $\mathbf{w}$ and b, considering α_i fixed and, secondly, recomputes α_i from the achieved solution in the first step. The algorithm iterates until the SVR solution is reached.

In order to apply this algorithm for the MSVR, we have joined together (6) and (7) to obtain $\mathbf{w}^j$ and b^j from the linear system in which α_i must remain unchanged.

$$\begin{bmatrix} \boldsymbol{\Phi}^T\mathbf{D}_{\boldsymbol{\alpha}}\boldsymbol{\Phi} + \mathbf{I} & \boldsymbol{\Phi}^T\boldsymbol{\alpha} \\ \boldsymbol{\alpha}^T\boldsymbol{\Phi} & \boldsymbol{\alpha}^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{w}^j \\ b^j \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^T\mathbf{D}_{\boldsymbol{\alpha}}\mathbf{y}^j \\ \boldsymbol{\alpha}^T\mathbf{y}^j \end{bmatrix} \tag{12}$$

Once, we have solved (12), we recompute the α_i values fulfilling the remaining KKT conditions, so when the algorithm stops, we will be at the MSVR solution. The value of α_i can be computed from the KKT conditions (8)-(11):

$$\alpha_i = \begin{cases} 0, & e_i < 0 \\ C, & e_i > 0 \end{cases} \tag{13}$$

where we have defined the error over each sample as $e_i = \varepsilon - \|\mathbf{y}_i - \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{b}\|^2$.

3.1 Reproducing Kernels Hilbert Space

The system in (12) needs to know the nonlinear mapping $\boldsymbol{\phi}(\cdot)$, which usually is not. In order to work with kernels ($\kappa(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j)$) as the SVR does, we use the Representer theorem [3], which states that the best solution can be expressed as a linear combination of the training samples in the feature space:

$$\mathbf{w}^j = \sum_{i=1}^{n} \beta_i^j \boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\Phi}^T\boldsymbol{\beta}^j \tag{14}$$

If we reorder (6), collecting the terms depending on $\mathbf{w}^j$ in one side of the equation and replace (14) in it and multiply it by the pseudo-inverse of $\boldsymbol{\Phi}\mathbf{D}_{\boldsymbol{\alpha}}$, we obtain:

$$(\mathbf{D}_{\boldsymbol{\alpha}}\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{D}_{\boldsymbol{\alpha}})^{-1}\mathbf{D}_{\boldsymbol{\alpha}}[\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{D}_{\boldsymbol{\alpha}}\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Phi}\boldsymbol{\Phi}^T]\boldsymbol{\beta}^j = [\mathbf{y}^j - \mathbf{1}b^j] \tag{15}$$

which can be simplified to:

$$[\mathbf{H} + \mathbf{D}_{\boldsymbol{\alpha}}^{-1}]\boldsymbol{\beta}^j = [\mathbf{y}^j - \mathbf{1}b^j] \tag{16}$$

where we have defined the kernel matrix as $\mathbf{H} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$. We can now join together (16) and (7) to form a linear system of equations that it does not depend on the nonlinear mapping $\boldsymbol{\phi}$, just on its kernel.

$$\begin{bmatrix} \mathbf{H} + \mathbf{D}_{\boldsymbol{\alpha}}^{-1} & \mathbf{1} \\ \boldsymbol{\alpha}^T\mathbf{H} & \boldsymbol{\alpha}^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}^j \\ b^j \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\alpha}^T\mathbf{y}^j \end{bmatrix} \tag{17}$$

The resolution of the system in (17) for each set of $\boldsymbol{\beta}^j$ is equal for every dimension of $\mathbf{y}$, except for the independent term. So its resolution will be much simpler than that of k one-dimensional SVR.

4 Examples

We first illustrate the MSVR with a synthetic example and afterwards we will solve a multi-dimensional estimation with real data. For the synthetic problem, the input vector is 2D and each component were generated independently from a Gaussian distribution function of zero mean and standard deviation equal to 10. The output vector $\mathbf{y} \in \mathbb{R}^5$, described by: $y_{i1} = 4\sin(x_{i1}) - 2sinc(x_{i2}) + 5 + n_{i1}$, $y_{i2} = 3\sin(x_{i1}) - 3\cos(x_{i2}) + 2 + n_{i2}$, $y_{i3} = -5sinc(x_{i1}) + 4\sin(x_{i2}) + 1 + n_{i3}$, $y_{i4} = -2\sin(x_{i1}) - 4\sin(x_{i2}) - 5 + n_{i4}$ and $y_{i5} = 4sinc(x_{i1}) - 2\cos(x_{i2}) - 3 + n_{i5}$, where n_{ij} are random Gaussian variables with zero mean and standard deviation equal to 0.5. We have solved this problem for several values of ε_2 and have run 10 independent trials. We have used an RBF kernel with $\sigma = 0.5$ and $C = 10$. The achieved results are shown in Figure 2 in which we have plotted the Mean Square Error (MSE), $MSE = \sum_{i=1}^{N_ts} \|\mathbf{y}_i - \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{b}\|^2$, and the fraction of Support Vectors for the different values of ε_2. The value of ε_1, used for the 1D SVR, is ε_1=0.6974 $\times$ ε_2, which is the one that gives the same hyper-volume for both insensitive zones.

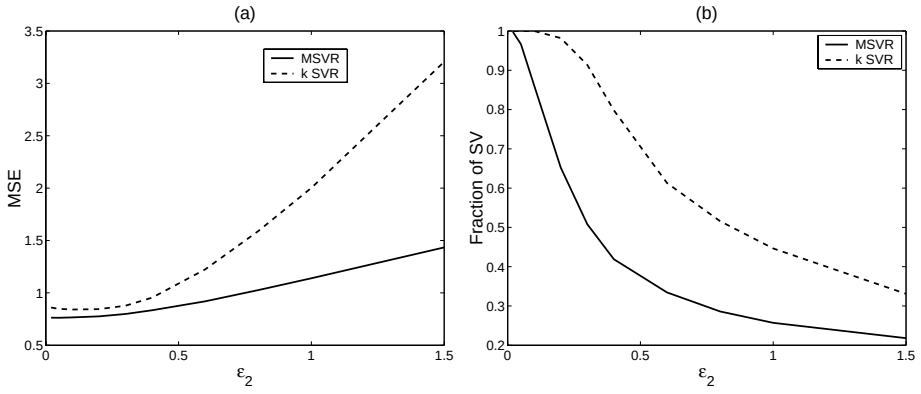

Fig. 2. The MSE is shown for the MSVR and for the k SVR in (a). In (b) we have plotted the fraction of support vector.

The MSVR gives a lower prediction error and fewer support vectors than using 5 SVRs. Furthermore, the MSVR is very robust when the value of ε is not correctly set as illustrated in Figure 2.

The second experiment deals with the simultaneous one-step ahead prediction of four clinical variables commonly used in therapeutic drug monitoring of patients who have undergone kidney transplantation. Eighteen lagged inputs formed by anthropometrical, clinical and biochemical data of patients are used to predict Cyclosporine blood concentration, the daily dosage, alkaline phosphates and the creatinine clearance. The data was split into a training set (665 samples) and a validation set (442 samples). We have used an RBF kernel with a $\sigma = 13.4$ and a $C = 35$, which were obtained using 8-fold CV over the training set. In Figure 3 we show the MSE and the fraction of support vectors for the MSVR

and 4 one-dimensional SVR. The input data were preprocessed to present zero mean and unit standard deviation.

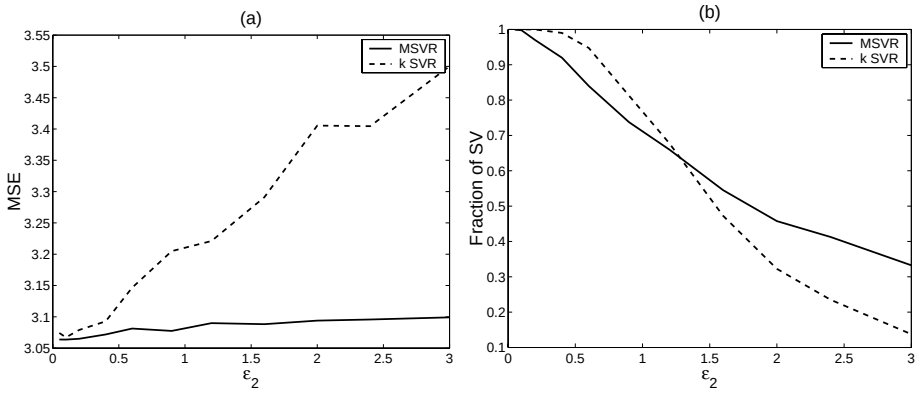

Fig. 3. The MSE is shown for the MSVR and for the k SVR in (a). In (b) we have plotted the fraction of support vectors.

As in the synthetic data we can see that the MSVR does not only achieves the best solution as ε varies but its solution is more robust and with fewer support vectors than using the SVR independently for each component.

5 Conclusions

We have presented a novel approach for solving multiple output regression problems in which we have defined a hyper-sphere insensitivity zone, that allow us to penalize only once the samples that are not placed inside the insentivity zone. The MSVR does not only achieves better predictions but it is also more robust than the use of SVR independently for each varaible to be estimated.

References

1. R. Fletcher. *Practical Methods of Optimization.* Wiley, Second Ed., 1987.
2. F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez. An IRWLS procedure for SVR. In *Proc. of the EUSIPCO'00*, Finland, Sept 2000.
3. B. Schoelkopf and A. Smola. *Learning with kernels.* M.I.T. Press, 2001.
4. V. N. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche, eds, *NIPS 1997*, pages 169–184.
5. V. N. Vapnik. *Statistical Learning Theory.* Wiley, 1998.