Online Algorithm for Orthogonal Regression

Roberto C. S. N. P. Souza[1], **Saul C. Leite**[1], **Raul Fonseca Neto**[1]

[1]Departamento de Ciência da Computação (DCC)
Instituto de Ciências Exatas (ICE) – Universidade Federal de Juiz de Fora (UFJF)
Rua José Lourenço Kelmer, s/n, São Pedro – 36.036-900 – Juiz de Fora – MG – Brasil

roberto.nalon@ice.ufjf.br, saul.leite@ufjf.edu.br

raulfonseca.neto@ufjf.edu.br

Área de pesquisa: Inteligência Computacional.

Ano de Ingresso: 2011

Abstract. *In this paper, we introduce a new online algorithm for orthogonal regression. The method is constructed via a stochastic gradient descent approach combined with the idea of a tube loss function, which is similar to the one used in support vector (SV) regression. The algorithm can be used in primal or in dual variables. The latter formulation allows the introduction of kernels and soft margins. In addition, an incremental strategy algorithm is introduced, which can be used to find sparse solutions and also an approximation to the "minimal tube" containing the data. The algorithm is very simple to implement and avoids quadratic optimization.*

Keywords: support vector machines, online algorithms, kernel methods, regression problem, orthogonal regression

1. Introduction

A regression problem consists of finding an unknown relationship between given points $x_i \in \mathbb{R}^n$ and their corresponding target values $y_i \in \mathbb{R}$. This problem is usually formulated as one of finding a function $f : \mathbb{R}^n \to \mathbb{R}$, which maps points to target values, that minimizes a certain loss function. Usually, it is assumed that noise is only present in the target values and the loss function measures deviations of $f(x_i)$ from its corresponding value y_i. This is the case of the classical regression formulation proposed by Gauss, which minimizes the sum of the quadratic deviation between targets and the estimated function [Huber 1972].

Aiming to solve the classical regression problem, Vapnik developed a formulation based on a loss function called ε-insensitive and introduced the concept of tube [Vapnik 1995]. These new elements, based on structural risk minimization principle, allowed the development of an specific support vector machine formulation for regression problems, called support vector (SV-) regression. This method has become quite popular due to its flexibility, specially with respect to the use of kernels [Smola and Schölkopf 2002].

Orthogonal regression has its origins with Adcock in 1877 [Adcock 1877] (see [Markovsky and Huffel 2007] for a historical overview). This regression problem appears in the literature under different names, for instance, it has been called total least-square [Golub 1973]and it is commonly called error-in-variables in the statistics community [Markovsky and Huffel 2007, Griliches and Ringstad 1970]. In this setting, noise can be present not only in the target values y_i but also in the data points x_i. This problem is motivated by several applications, as for instance, in audio [Hermus et al. 2005] and image [Luong et al. 2012]processing. The usual approach for solving the orthogonal regression problem is via a singular value decomposition of the associated data matrix [Markovsky and Huffel 2007].

In this work, we present a new formulation for orthogonal regression which adapts the idea of ε-insensitive loss function introduced by Vapnik. The problem is defined as the minimization of the empirical risk with respect to a novel tube loss function for orthogonal regression. An algorithm to solve this problem is proposed based on the stochastic gradient descent approach, similar to Rosenblatt's Perceptron [Rosenblatt 1958]. The proposed method can be used in primal or in dual variables, making it more flexible for different types of problems. In dual variables, the algorithm allows the introduction of kernels and margin flexibility. To the best of our knowledge, this is the first method that allows the introduction of kernels via the so-called "kernel-trick" for orthogonal regression. In addition, an incremental strategy is introduced, similar to the one proposed in [Leite and Fonseca Neto 2008], which can be used to find more sparse solutions and also an approximation of the "minimal tube" containing the data. It is important to mention that this minimal tube cannot be found using the standard SV-regression approach.

2. The Regression Problem and Loss Functions

Let $X_m := \{x_i\}_{i=1}^m$, where $x_i \in \mathbb{R}^n$, be the set of training points and let $Y_m := \{y_i\}_{i=1}^m$, with $y_i \in \mathbb{R}$ the corresponding set of target values. Let $Z_m := \{(y, x) : y \in Y_m$ and $x \in X_m\}$ be the training set. The general problem of regression can be stated as follows: suppose that the pairs $z_i := (y_i, x_i) \in Z_m$ are independent samples of a random vector

$Z := (Y, X)$, where Y and X are correlated and have an unknown joint distribution $\mathbb{P}_Z$. Given Z_m, the problem is to find an unknown relationship between points and their respective targets, given by a function $f : \mathbb{R}^n \to \mathbb{R}$ over a certain class $\mathcal{C}$ of functions, which minimizes the *expected risk*: $\mathbb{E}_Z[\ell(Y, X, f)]$, where the expectation is taken over the distribution $\mathbb{P}_Z$ and $\ell : \mathbb{R} \times \mathbb{R}^n \times \mathcal{C} \to \mathbb{R}$ is a *loss function*, which penalizes deviations between functional and target values.

One approach in this case is to use Z_m to estimate $\mathbb{P}_Z$, however, this generally turns out to be a more challenging task than the original problem. Therefore, it is common to consider the reduced problem of finding a function $f \in \mathcal{C}$ which minimizes the *empirical risk* given the training set Z_m, that is: $R_{\text{emp}}[Z_m] := \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, x_i, f)$.

Since we are interested in applying the so-called "kernel trick" later on, we restrict our class of functions $\mathcal{C}$ to linear functions of the form: $f_{(w,b)}(x) := \langle w, x\rangle + b$, where $w \in \mathbb{R}^n$ is the normal vector and $b \in \mathbb{R}$ is the bias term.

The most common choice for ℓ is the squared loss given by $\ell_2(y, x, f) := (y - f(x))^2$, which gives origin to the least-square regression. The rationale behind this approach is to minimize the sum of squared residuals $\delta y_i := y_i - f(x_i)$ in such a way that $y_i = f(x_i) + \delta y_i$. The common assumption is that *only* the target values are random. However, some practical problems may have noisy training points x_i as well. A natural generalization of the above procedure is to also minimize variation in the training points x_i, that is, minimize $\delta y_i^2 + \delta x_i^2$ in such a way that $y_i = f(x_i + \delta x_i) + \delta y_i$. This is commonly called total least-square or orthogonal regression [Markovsky and Huffel 2007]. Geometrically, this problem minimizes the sum of squared orthogonal distances between the points $z_i := (y_i, x_i)$ and the hyperplane, as opposed to the least-square formulation, which minimizes the sum of the squared direct differences between functional values and target values.

The corresponding loss function for the total least-square formulation can be written as: $\ell_t(y, x, f_{(w,b)}) := \frac{(y-\langle w,x\rangle-b)^2}{||(1,w)||^2}$, where the norm $|| \cdot ||$ with no lower index corresponds to the $l2$ norm $|| \cdot ||_2$.

Another common choice for the loss function, which is used in the SV-regression, is called the ε-insensitive (or ε-tube) loss. It is given by: $\ell_\varepsilon(y, x, f) := \max\{0, |y - f(x)| - \varepsilon\}$, where ε is interpreted as the *radius* of this tube. Therefore, this loss function penalizes solutions which leaves training points outside of this tube. A favorable feature of this loss function is that it gives *sparse solutions* when formulated in dual variables. In respect to this loss function, let us introduce some notation and terminology which will be used later on. For each fixed $\varepsilon > 0$, define the following set:

$$\mathcal{V}(Z_m, \varepsilon) := \{(w, b) \in \mathbb{R}^{n+1} : |y_i - \langle w, x_i\rangle - b| \leq \varepsilon, \forall (x_i, y_i) \in Z_m\},$$

which we call *version space*. When this set is not empty, we say that the problem accepts a tube of size ε, or an ε-tube.

In order to consider a tube loss function for the orthogonal regression problems, which is useful to maintain the sparsity of the dual solution, we propose the following loss function: for a $\rho > 0$, let $\ell_\rho(y, x, f_{(w,b)}) := \max\left\{0, \frac{|y-\langle w,x\rangle-b|}{||(1,w)||} - \rho\right\}$, which we call ρ-insensitive (or ρ-tube) loss. In a similar fashion to what was done for the ℓ_ε loss function, define the following version space:

$$\Omega(Z_m, \rho) := \{(w, b) \in \mathbb{R}^{n+1} : |y_i - \langle w, x_i \rangle - b| \leq \rho||(1, w)||, \forall (x_i, y_i) \in Z_m\},$$

for each $\rho > 0$. Again, we say that the problem accepts a ρ-tube if the version space is not empty.

For each fixed (w, b), notice that there is an interesting relationship between the orthogonal distances and the direct functional differences for each point (y_i, x_i). For that let $\varepsilon_i := y_i - \langle w, x_i \rangle - b$ and $\rho_i := (y_i - \langle w, x_i \rangle - b)/||(1, w)||$, then clearly $\varepsilon_i = \rho_i ||(1, w)||$.

3. Online Algorithms for Regression

In the online learning setting, one constructs the candidate functions $f \in \mathcal{C}$ (usually called *hypothesis*) minimizing the empirical risk by examining one training example (y_i, x_i) at a time. In this way, we start with an initial hypothesis f_0 and, at each iteration t, the algorithm examines one example and updates its current hypothesis f_t according to some specific update rule.

In order to derive this update rule, we follow the ideas of the Perceptron algorithm [Rosenblatt 1958] and use an stochastic gradient descent approach. Considering the empirical risk defined in the previous section, let us define the following cost:

$$J(f) := \sum_{(y_i, x_i) \in Z_m} \ell(y_i, x_i, f),$$

which should be minimized in respect to f. Therefore, for each pair of points (y_i, x_i), the following update rule is applied to the current hypothesis f_t:

$$f_{t+1} \longleftarrow f_t - \eta \partial_f \ell(y_i, x_i, f) \tag{1}$$

where $\eta > 0$ is usually called *learning rate* and ∂_f denotes the gradient of the loss function with respect to f.

3.1. Fixed Radius Perceptron (FRP)

In this section, the proposed online algorithm for orthogonal regression is presented. It is based on the online learning setting described in the previous section applied to the ρ-insensitive loss function. In order to do so, we begin by presenting an online algorithm for classical regression using the ε-tube loss function, which is the same used in the SV-regression formulation. The derivation of this algorithm helps to introduce the main ideas before presenting the algorithm for orthogonal regression. It is also used for comparison in the experimental section.

Let us apply the ideas of the previous section to the loss function ℓ_ε, restricting our class of functions $\mathcal{C}$ to linear functions $f_{(w,b)}$. Then the condition $\ell_\varepsilon(\cdot) > 0$ to update the hypothesis $f_{(w_t, b_t)}$ after example (y_i, x_i) becomes:

$$|y_i - \langle w_t, x_i \rangle - b_t| > \varepsilon. \tag{2}$$

For the update rule, the gradient in equation (1) is taken in respect to the parameters (w, b) that compose the function $f_{(w,b)}$. Therefore we have:

$$\begin{aligned} w_{t+1} &\longleftarrow w_t + \eta \text{ sign}(y_i - \langle w_t, x_i \rangle - b_t) x_i \\ b_{t+1} &\longleftarrow b_t + \eta \text{ sign}(y_i - \langle w_t, x_i \rangle - b_t), \end{aligned} \tag{3}$$

where sign$(x) := x/|x|$, for $x \in \mathbb{R} \setminus \{0\}$. We call this algorithm *Fixed ε-Radius Perceptron* (εFRP). A similar algorithm has been presented in [Kivinen et al. 2004], using a similar loss function. The algorithms are equivalent when the parameter ν, used in [Kivinen et al. 2004], is set to zero.

Since we are interested in orthogonal regression, we consider the ρ-tube loss function presented in Section 2. Following an analogous derivation, the condition to update the hypothesis after examining (y_i, x_i) becomes: $\frac{|y_i - \langle w_t, x_i \rangle - b_t|}{||(1, w_t)||} > \rho$.

The corresponding update rule has the following form:

$$\begin{aligned} w_{t+1} &\longleftarrow w_t \lambda_t + \eta \left(\frac{\text{sign}(y_i - \langle w_t, x_i \rangle - b_t) x_i}{||(1, w_t)||} \right) \\ b_{t+1} &\longleftarrow b_t + \eta \left(\frac{\text{sign}(y_i - \langle w_t, x_i \rangle - b_t)}{||(1, w_t)||} \right), \end{aligned} \tag{4}$$

where λ_t is given by

$$\lambda_t := \left(1 + \eta \frac{|y_i - \langle w_t, x_i \rangle - b_t|}{||(1, w_t)||^3} \right). \tag{5}$$

We call the corresponding algorithm *Fixed ρ-Radius Perceptron* (ρFRP).

4. Algorithm in dual variables

Suppose now that the training examples are in some abstract space $\mathcal{X}$. In addition, suppose that the functions $f \in \mathcal{C}$ accept the following representation: $f = f_{\mathcal{H}} + b$, for some $f_{\mathcal{H}} \in \mathcal{H}$ and $b \in \mathbb{R}$, where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) (e.g., [Smola and Schölkopf 2002]). Let $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the associated inner product and reproducing kernel, respectively. Then, the *reproducing property* of k implies that $k(x, \cdot) \in \mathcal{H}$ and, for any $f \in \mathcal{H}$, we have that $\langle f, k(x, \cdot) \rangle_{\mathcal{H}} = f(x)$ for all $x \in \mathcal{X}$. Another interesting property of RKHS is that any $f \in \mathcal{H}$ can be written as linear combination of the $k(x, \cdot)$. This fact is very useful for learning algorithms since we can write the hypothesis at iteration t as:

$$f_t(x) = \sum_{i=1}^{m} \alpha_{t,i} k(x_i, x) + b_t \tag{6}$$

for some $\alpha_t := (\alpha_{t,1}, \ldots, \alpha_{t,m})' \in \mathbb{R}^m$, $b_t \in \mathbb{R}$, $x \in \mathcal{X}$ and $x_i \in X_m$. In this sense, we can define $w_t := \sum_{i=1}^{m} \alpha_{t,i} k(x_i, \cdot)$ and interpret the function f_t, given in equation (6), as:

$$f_t(x) = \langle w_t, k(x, \cdot) \rangle_{\mathcal{H}} + b_t, \tag{7}$$

by the reproducing property of k. Let $|| \cdot ||_{\mathcal{H}}$ be the norm induced by the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, i.e. $||f||^2_{\mathcal{H}} := \langle f, f \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$. Then, the norm of w_t can be written as: $||w_t||^2_{\mathcal{H}} := \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_{t,i} \alpha_{t,j} k(x_i, x_j)$, by the reproducing property.

Usually, in practice, the above construction of the function class $\mathcal{C}$ is established by choosing a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which intuitively *measures similarities* between points in $\mathcal{X}$. If this function k attends Mercer's condition (e.g.

[Smola and Schölkopf 2002]), it can be shown that there is a corresponding reproducing kernel Hilbert space $\mathcal{H}$ that has k as its associated kernel. When $\mathcal{X} = \mathbb{R}^n$, one possible choice for k is the inner product $\langle \cdot, \cdot \rangle$ of $\mathbb{R}^n$. This leads us to the linear representation of f used in the previous sections.

In this sense, the update rule for ρFRP in dual variables is constructed.Following the update rule given in equation (4), notice that w_t is scaled by a factor of λ_t, given by equation (5). In dual variables this corresponds to scaling the vector α_t by the same factor λ_t before the associated component of α_t is updated. Hence, if the ith example (y_i, x_i) is outside of the ρ-tube, the update is done as follows: first α_t is scaled by λ_t, then the following update is performed:

$$\alpha_{t+1,i} \longleftarrow \alpha_{t,i} + \eta \left(\frac{\text{sign}(y_i - f_t(x_i))}{||(1, w_t)||} \right),$$

where we define $||(1, w_t)|| := \sqrt{1 + ||w_t||^2_{\mathcal{H}}}$. Notice that the update rule for the εFRP in dual variables can be constructed in a similar fashion.

5. Incremental strategy

In this section, we consider an incremental strategy, based on the one introduced in [Leite and Fonseca Neto 2008], which can be useful to give *sparse solutions* and to find an approximation to the *minimal tube* containing the data. For this, we restrict the discussion in this section to the ρFRP algorithm, although the same arguments can be extended directly to εFRP.

Given a training set Z_m and a fixed constant ρ, the ρFRP algorithm can be seen as one that finds a point (w, b) inside the version space $\Omega(Z_m, \rho)$. Suppose that one is able to construct a tube radius $\tilde{\rho}$ such that $\tilde{\rho} < \rho$ from a solution $(w, b) \in \Omega(Z_m, \rho)$ in such a way that the new version space $\Omega(Z_m, \tilde{\rho})$ is not empty. Then the ρFRP algorithm can be used to find a sequence of strictly decreasing tube radii $\rho_0, \rho_1, \ldots, \rho_n$ such that the corresponding version spaces are not empty.

One application of such strategy of constructing this sequence of decreasing tube radii is to identify *support vectors* or points that are the most outlining among the training set. Suppose for instance that a radius ρ_f is desired for a given problem. Then one can proceed in the following fashion: first one chooses a large ρ_0 and progressively decreases this radius up to a final radius ρ_n such that $\rho_n \leq \rho_f$. This way, as the radius shrinks, only the most outlining training points will have an effect on the hypothesis, contributing to the sparsity of the solution.

In the case where soft margins are not desired, one can use the above strategy to approximate the minimal tube containing the data, that is:

$$\rho^* := \inf\{\rho : \Omega(Z_m, \rho) \neq \emptyset\}.$$

This can be done by iteratively producing new radii ρ_n up to a final iteration N where $\rho_N \approx \rho^*$.

6. Experimental results

This section was constructed aiming to cover the technical features of the proposed method.

6.1. Treating variables symmetrically

An interesting technical feature of orthogonal regression models is that it treats the variables of the problem symmetrically [Ammann and Ness 1988]. Such procedure could be very useful when the problem actually does not have *independent* and *dependent* variables, and should instead be treated equally.

This first toy example shows that the proposed method presents the behavior discussed above. Figure 1 shows three regression lines. The solid line corresponds to the orthogonal regression using the ρFRP. The dashed line is the standard regression using εFRP to predict Y (as dependent variable) from X (as independent variable). The dash-dot line is the standard regression using εFRP, where the dependent and independent variables where swapped. That is, the regression is predicting X from Y. Also, Figure 1 depicts the lines of the respective distances that are measured from a typical point. It is important to mention that if we swap the variables and run the orthogonal regression with ρFRP, the same solid line is obtained.

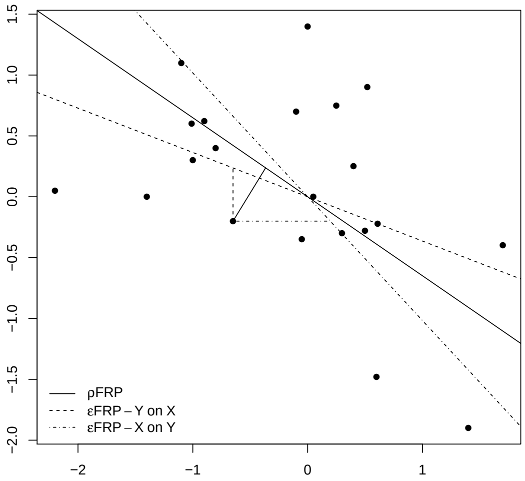

Figura 1. Orthogonal regression (ρFRP) and standard regression (εFRP).

6.2. Sparsity

In this group of experiments, we applied the ρFRP and εFRP algorithms combined with the incremental strategy algorithm (ISA) to different datasets. In order to observe this, we also tested both methods without ISA. The results of SVM-light [Joachims 1999] are also presented for comparison.

The training sets were generated from a chosen function and polluted with different noise intensities in both variables. For the testing set, we generated a new dataset using the same function, with different points distributed over the same input range, with no noise added. The testing set has twice the number of points of the training set. The sets are described below for each chosen function.

The *Linear1* training set has 51 examples generated from the function $y = 2x + 0.1$, where $x \in [-5, 5]$. Both variables are polluted with a Gaussian noise with standard deviation $\sigma_x = \sigma_y = 0.2$. The *Exp1* training set has 51 examples of the function $y = e^{-x^2}$, where $x \in [-1, 1]$ and Gaussian noise is introduced with $\sigma_x = \sigma_y = 0.1$.

In order to compare the solutions, we present the following data obtained from the experiments: the number of support vectors (sv), shown to observe sparsity; the orthogonal (ρ) and vertical (ε) tube radii; and the norm of the solution ($n = ||(1, w)||$). For the FRP methods, we also present the total number of iterations (it) and total number of updates (up) performed by the algorithms. In order to measure the quality of the fit, we use two error measures. The first is the *root mean squared error* (RMSE). The RgMSE criterion takes the root of the squared orthogonal deviation.

The test presented in this section were performed in the following way. First, we calculated the range of the target values $r := \max_{i=1,\dots,m} y_i - \min_{i=1,\dots,m} y_i$ in the training set. Then, we set the value of ε to $0.1r$ for the εFRP and SVM algorithms. Also, this value was used as an stop criterion for εFRP$_{ISA}$. In order to compare the results, we calculated the respective ρ obtained for the εFRP solution, by the equation $\varepsilon = \rho||(1, w)||$, and used it to run the ρFRP and ρFRP$_{ISA}$ algorithms. The capacity control parameter C was set to $C/m = 10$, where m is the number of training examples, as suggested in [Smola and Schölkopf 2002]. Also, we set the learning rate to $\eta = 0.01$. For the *Linear1* the linear kernel $k(x_i, x_j) := \langle x_i, x_j \rangle$ was used. For *Exp1*, we used a polynomial kernel $k(x_i, x_j) := (s \langle x_i, x_j \rangle + c)^d$ with $d = 2$, $s = 1$ and $c = 0$. In each case, we saved the model and used it to perform the tests. The RMSE and RgMSE criteria are measured in the training and testing sets. The results are presented in Table 1.

Tabela 1. Results from orthogonal (ρFRP) and classical regression (εFRP and SVM).

	sv	$\rho/\varepsilon/n$	it/u	Training		Testing	
				RMSE	RgMSE	RMSE	RgMSE
Linear1							
ρFRP	44	0.626/1.344/2.149	7/65	0.48068	0.22367	0.28514	0.13268
ρFRP$_{ISA}$	5	0.625/1.327/2.122	92/46	0.56054	0.26415	0.41071	0.19354
εFRP	37	0.626/1.952/3.120	4/50	0.88479	0.28359	0.81896	0.26249
εFRP$_{ISA}$	6	0.639/1.928/3.015	74/37	1.00106	0.33201	0.94416	0.31314
SVM	2	0.998/1.928/1.932	-/-	1.09079	0.56468	1.04105	0.53893
Exp1							
ρFRP	40	0.094/0.240/2.566	1105/5789	0.11911	0.04642	0.03528	0.01375
ρFRP$_{ISA}$	5	0.094/0.230/2.460	2580/3894	0.11343	0.04611	0.03103	0.01261
εFRP	41	0.094/0.098/1.043	26304/164166	0.11382	0.10912	0.04679	0.04486
εFRP$_{ISA}$	22	0.093/0.098/1.055	25727/130023	0.11272	0.10685	0.03840	0.03640
SVM	22	0.093/0.080/1.158	-/-	0.11289	0.09748	0.04033	0.03483

First notice that ISA performs well in respect to maintaining the sparsity of the solution. The FRP methods have less support vectors when combined with the ISA. Secondly, when using ISA, the FRP algorithms show a higher number of iterations, as expected. However, they make fewer number of updates.

Considering the quality of the fit, the orthogonal regression methods present the best results. For dataset *Exp1*, the results are very similar for the RMSE measure, however the εFRP and εFRP$_{ISA}$ performed a large number of iterations and updates.

7. Conclusions

In this paper we introduced a new formulation for orthogonal regression based on an online training approach using the stochastic gradient descent. When formulated in dual variables the algorithm allows the introduction of kernels, via the kernel trick, and margin flexibility. The algorithm is entirely based on perceptron which makes it simple to understand and implement. To the best of our knowledge, this is the first online algorithm for orthogonal regression with kernels.

Also we presented an incremental strategy algorithm that can be combined with the proposed method in order to find sparse solutions and also the minimal tube containing the data. The experimental results show that the proposed method works well in comparisons with classical regression approach when different scenarios of noise are presented.

Referências

Adcock, R. (1877). Note on the method of least squares. *Analyst 4*, pages 183–184.

Ammann, L. and Ness, J. V. (1988). A routine for converting regression algorithms into corresponding orthogonal regression algorithms. *ACM Transactions on Mathematical Software*, 14:76–87.

Golub, G. H. (1973). Some Modified Matrix Eigenvalue Problems. *SIAM Review*, 15:318–334.

Griliches, Z. and Ringstad, V. (1970). Error-in-the-variables bias in nonlinear contexts. *Econometrica*, 38(2):pp. 368–370.

Hermus, K., Verhelst, W., Lemmerling, P., Wambacq, P., and Huffel, S. V. (2005). Perceptual audio modeling with exponentially damped sinusoids. *Signal Processing*, 85(1):163 – 176.

Huber, P. J. (1972). Robust statistics: A review. *The Annals of Mathematical Statistics*, 43:1041–1067.

Joachims, T. (1999). Making large-scale support vector machine learning practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in kernel methods*, pages 169–184. MIT Press.

Kivinen, J., Smola, A., and Williamson, R. (2004). Online learning with kernels. *IEEE Transactions on Signal Processing*, 52:2165–2176.

Leite, S. C. and Fonseca Neto, R. (2008). Incremental margin algorithm for large margin classifiers. *Neurocomputing*, 71:1550–1560.

Luong, H. Q., Goossens, B., Pizurica, A., and Philips, W. (2012). Total least square kernel regression. *Journal of Visual Communication and Image Representation*, 23:94–99.

Markovsky, I. and Huffel, S. V. (2007). Overview of total least-square methods. *Signal Processing*, 87:2283–2303.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

Smola, A. and Schölkopf, B. (2002). *Learning with Kernels.* MIT Press.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer.