Online Method for Regression with an Incremental Strategy

Roberto C. S. N. P. Souza*†, Saul C. Leite†, Carlos C. H. Borges†, Raul Fonseca Neto†
*Department of Computer Science, Federal University of Minas Gerais,
nalon@dcc.ufmg.br
†Department of Computer Science, Federal University of Juiz de Fora,
{saul.leite, raulfonseca.neto}@ufjf.edu.br, cchb@lncc.br

***Abstract*—In this contribution, we introduce an online method for regression composed of two parts. The first is based on a stochastic gradient descent approach combined with the idea of tube used in support vector regression. This algorithm can be used in primal or in dual variables. The latter formulation allows the introduction of kernels and soft margins. The second part consists of an incremental strategy algorithm which is introduced in order to find *sparse solutions*. Also, when soft margin is not desirable, this incremental strategy may be used to obtain the "minimal tube" containing the data. The algorithm is very simple to implement and avoids quadratic optimization. Numerical results show that the method works well in comparison to a standard implementation of the SV-regression.**

I. 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. 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 [1]. This approach presents the best estimative for empirical risk minimization principle, when the noise related to the real sample of data follows a normal distribution.

In [2], Vapnik adapted the ideas of the support vector machines, originally introduced for binary classification problems [3], to the regression problem. Thus, a formulation based on the ε-insensitive loss function was derived and the concept of tube was introduced in this scenario. The derived method is known as the support vector (SV-) regression. This method has become quite popular due to its flexibility, specially with respect to the use of kernels [4]. In its standard formulation, the SV-regression requires the solution of a quadratic optimization problem, which may be computationally expensive for large problems and some time dependent applications.

For classification problems, considerable attention has been given to finding simple and efficient algorithms to construct large margin classifiers that avoid the complexity of quadratic programming, a few examples of the vast literature includes [5], [6], [7], [8], [9], [10], [11]). Most of them consider classification problems. The regression problem is discussed briefly in [8] and an algorithm for regression is derived using a modified version of the ε-insensitive loss function, although no numerical examples of the proposed method are given there. The idea is to adapt the size of the tube ε as the algorithm iterates through the data. This process, however, may result in a solution which is not sparse, since several points in the training set may contribute to the final solution. In [12], Bi and Bennet proposed a geometric interpretation for the regression problem transforming it in a binary classification problem for a given ε value. In principle, one could use this technique to extend online algorithms for classification to the regression problems. However, this procedure produces a duplicate set points, making it less attractive to practical application.

In this work, we propose an online algorithm which uses the stochastic gradient descent approach, similar to Rosenblatt's Perceptron [13], to regression problems. The 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. In addition, we introduce an incremental strategy algorithm, which can be combined with the method in order to find sparse solutions. Also, when margin flexibility is not desirable this incremental strategy algorithm may be used to obtain an approximation to the "minimal tube" containing the data. It is important to mention that this minimal tube cannot be found using the standard SV-regression approach.

The paper is structured as follows. Section II introduces the regression problem. In section III, the framework for online algorithms is presented and the proposed algorithm is derived in primal variables. Section IV derives the algorithm in dual variables. In section V, we introduce the incremental strategy algorithm. Furthermore, some numerical tests and results are shown in section VI to support the theory. Finally, section VII presents some conclusions.

II. Preliminaries

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 $Z_m := \{(y, x) : y \in Y_m \text{ and } x \in X_m\}$ 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 $\mathcal{P}_Z$. 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*: $\mathrm{E}_Z[\ell(Y, X, f)]$, where the expectation is taken over the distribution $\mathcal{P}_Z$ 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. An approach in this case is to use

Z_m to estimate $\mathcal{P}_Z$, 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 residuals $\delta y_i := y_i - f(x_i)$ in such a way that $y_i = f(x_i) + \delta y_i$.

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. This way, 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.

III. Online Framework for Regression

In the online learning setting, one constructs the candidate functions (usually called *hypothesis*) $f \in \mathcal{C}$ minimizing the empirical risk by examining one training example (y_i, x_i) at a time. This way, one starts 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 [13] and use an stochastic gradient descent approach. In view of 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. This way, 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.

An important aspect of this approach is that if $\ell(\cdot) \geq 0$, which is true for most loss functions, the above updates need only take effect in case $\ell(y_i, x_i, f) > 0$. Otherwise, the current hypothesis f_t already achieves the minimum for the example (y_i, x_i) and no change needs to be made, i.e. $f_{t+1} = f_t$. In this sense, loss functions that are based on the idea of tube are well suited for this scheme, since the example will only affect the current hypothesis if it is outside of the tube.

A. Fixed Radius Perceptron (FRP)

Let us now apply the ideas introduced in the previous section to the loss function ℓ_ε, restricting our class of functions $\mathcal{C}$ to linear functions $f_{(w,b)}(x)$. 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 $\text{sign}(x) := x/|x|$, for $x \in \mathbb{R}\backslash\{0\}$. We call this algorithm *Fixed ε-Radius Perceptron* (εFRP). Similar ideas has been presented in [8], using a similar loss function. The algorithms are similar when the parameter ν, used in [8], is set to zero. The εFRP algorithm is presented in details in Algorithm 1.

Algorithm 1 εFRP in primal variables

Input: Z_m, w_{init}, b_{init}, η, ε, T
Output: (w, b)
$w_0 \leftarrow w_{init}$, $b_0 \leftarrow b_{init}$, $t \leftarrow 0$
repeat
  for $i = 1, \ldots, m$ **do**
    if $|y_i - \langle w_t, x_i\rangle - b| > \varepsilon$ **then**
      $w_{t+1} \leftarrow w_t + \eta\,(\text{sign}(y_i - \langle w_t, x_i\rangle - b_t) x_i)$
      $b_{t+1} \leftarrow b_t + \eta\,(\text{sign}(y_i - \langle w_t, x_i\rangle - b))$
    end if
  end for
  $t \leftarrow t + 1$
until no mistakes were made **or** $t > T$

IV. 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., [4]). 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 for any $f \in \mathcal{H}$ one has that $k(x, \cdot) \in \mathcal{H}$ and $\langle f, k(x, \cdot)\rangle_{\mathcal{H}} = f(x)$ 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 one can write the hypothesis at iteration t as:

$$f_t(x) = \sum_{i=1}^{m} \alpha_{t,i} k(x_i, x) + b_t \tag{4}$$

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 (4), as:

$$f_t(x) = \langle w_t, k(x, \cdot)\rangle_{\mathcal{H}} + b_t, \tag{5}$$

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. [4]), it can be shown that there is a corresponding reproducing kernel Hilbert space $\mathcal{H}$ that has k as it 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, given the above representation of w_t as the linear combination $\sum_{i=1}^{m} \alpha_{t,i}k(x_i, \cdot)$, we can derive the update rule for the εFRP in dual variables α_t by examining the update rule given by (3). For a mistake in the example (y_i, x_i) the update rule for w_t becomes:

$$\sum_{i=1}^{m} \alpha_{t+1,i}k(x_i, \cdot) \longleftarrow \sum_{i=1}^{m} \alpha_{t,i}k(x_i, \cdot) + \eta \text{ sign}(y_i - f_t(x_i))k(x_i, \cdot),$$

which implies the following update rule for the dual variable α_t:

$$\alpha_{t+1,i} \longleftarrow \alpha_{t,i} + \eta \text{ sign}(y_i - f_t(x_i)) \tag{6}$$

The εFRP algorithm in dual variables is given by Algorithm 2.

Algorithm 2 εFRP in dual variables

Input: Z_m, α_{init}, b_{init}, η, ρ, T
Output: α, b
$\alpha_0 \leftarrow \alpha_{init}$, $b_0 \leftarrow b_{init}$, $t \leftarrow 0$
calculate $f_0(x_j)$, for $j = 1, \ldots, m$
repeat
  for $i = 1, \ldots, m$ **do**
    if $|y_i - f_t(x_i)| > \varepsilon$ **then**
      $\alpha_{t+1,i} \leftarrow \alpha_{t,i} + \eta\,(\text{sign}(y_i - f_t(x_i)))$
      $b_{t+1} \leftarrow b_t + \eta\,(\text{sign}(y_i - f_t(x_i)))$
    end if
  end for
  calculate $f_{t+1}(x_j)$, for $j = 1, \ldots, m$
  $t \leftarrow t + 1$
until no mistakes were made **or** $t > T$

A. Margin Flexibility

The complex nature of data frequently leads to problems where noise are present. In such cases, an accurate representation of the training set may result in a hypothesis with a poor generalization performance. It is then important to have some mechanism to weigh the trade-off between an accurate representation of the outlining data and the generalization capability of the hypothesis. In the setting of ε-tube loss function, one common approach is to consider *soft margins*, where one allows some of the most outlining training points to cross the tube boundaries.

An early formulation of soft margin was proposed for a linear programming scheme in [14]. Few years later in [2], Vapnik adapted the concept of soft margin to support vector machines. In this approach, slack variables are added to the problem's constraints to allow margin violation. These slack variables are then penalized in the cost function and a parameter C is introduced as a measure of the trade-off between the number and magnitude of margin violations and the accurate representation of the training data.

Another common approach to soft margins is to simply add a constant $\lambda_{diag} > 0$ to the diagonal of the kernel matrix [4]:

$$\tilde{K} := K + \lambda_{diag} I,$$

where the kernel matrix is defined as $K \in \mathbb{R}^{m\times m}$ with components $K_{ij} := k(x_i, x_j)$, for $x_i, x_j \in X_m$. It can be shown that this approach is equivalent to the introduction of slack variables in SVM formulation when they are penalized quadratically [4]. In fact, it is possible to establish a direct relationship between this constant λ_{diag} and the parameter C, which is given by $\lambda_{diag} = 1/2C$ (see for instance [15]).

In order to consider margin flexibility for the proposed method, we follow a similar idea and add a constant λ_{diag} to the diagonal of the kernel matrix. It is interesting to see the effect of this constant λ_{diag} on the εFRP algorithm. For the modified kernel matrix, notice that the condition for a point (y_i, x_i) to be inside of the tube at iteration t can be written as:

$$-\varepsilon \le y_i - \sum_{j=1}^{m} \alpha_{t,j}\tilde{K}_{ij} - b_t \le \varepsilon,$$

or equivalently, using the definition of $\tilde{K}$:

$$-\varepsilon + \alpha_{t,i}\lambda_{diag} \le y_i - \sum_{j=1}^{m} \alpha_{t,j}K_{ij} - b_t \le \varepsilon + \alpha_{t,i}\lambda_{diag}.$$

Notice that $\alpha_{t,i}$ will most likely have the same sign of $y_i - f_t(x_i)$, by the update rule given by equation (6). This implies that the desired flexibility of the margin constraint is achieved by adding a slack $\alpha_{t,i}\lambda_{diag}$ to the problem's constraints.

V. Incremental Strategy

In this section, we present an incremental strategy algorithm (ISA), similar to the one introduced in [11]. This incremental strategy may be used combined with εFRP in order to find *sparse solutions* and also can be useful to find the *minimal tube* containing the data.

Given a training set Z_m and a fixed constant ε, the εFRP algorithm can be seen as one that finds a point (w, b) inside the version space $\mathcal{V}(Z_m, \varepsilon)$. Suppose that one is able to construct a tube radius $\tilde{\varepsilon}$ such that $\tilde{\varepsilon} < \varepsilon$ from a solution $(w, b) \in \mathcal{V}(Z_m, \varepsilon)$ in such a way that the new version space $\mathcal{V}(Z_m, \tilde{\varepsilon})$ is not empty. Then one can use the εFRP algorithm to find a sequence of strictly decreasing tube radii $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_n$ such that the corresponding version spaces are not empty.

One application of such a strategy is to identify *support vectors*, or points that are the most outlining among the training set. Suppose for instance that a radius ε_f is desired for a given problem. Then one can proceed in the following fashion: first

one chooses a large ε_0 and progressively decreases this radius up to a final radius ε_n such that $\varepsilon_n \leq \varepsilon_f$. 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:

$$\varepsilon^* := \inf\{\varepsilon : \mathcal{V}(Z_m, \varepsilon) \neq \emptyset\}.$$

This can be done by iteratively producing new radii ε_n up to a final iteration N where $\varepsilon_N \approx \varepsilon^*$.

In order to construct a new radius value ε_{n+1} from the previous one ε_n, suppose that the εFRP finds a solution in $\mathcal{V}(Z_m, \varepsilon_n)$, namely (w_n, b_n). Define the corresponding positive and negative radius as: $\varepsilon_n^+ = \max_i \{y_i - \langle w_n, x_i\rangle - b_n\}$ and $\varepsilon_n^- = \max_i \{\langle w_n, x_i\rangle + b_n - y_i\}$.

A desirable feature for the final solution ε^* is to have the positive and negative radius balanced. This way, we can update the radius by setting:

$$\varepsilon_{n+1} = \frac{(\varepsilon_n^+ + \varepsilon_n^-)}{2}. \tag{7}$$

Notice that this new radius always yields a feasible solution, since

$$-\varepsilon_n^- \leq y_i - \langle w_n, x_i\rangle - b_n \leq \varepsilon_n^+ \quad \forall i \tag{8}$$

and by adding $(\varepsilon_n^- - \varepsilon_n^+)/2$ to the inequality, a new solution is obtained by only changing the bias term.

However, in some cases, it is possible to have $\varepsilon_n^+ \approx \varepsilon_n^-$ and the new radius would not be much different from the previous value. In order to cope with this, we use the following update rule for the radius:

$$\varepsilon_{n+1} = \min\left\{\frac{(\varepsilon_n^+ + \varepsilon_n^-)}{2}, (1-\delta/2)\varepsilon_n\right\}.$$

It turns out that by adopting this rule, we might end up with an empty version space and the εFRP will not converge. Hence we stipulate a maximum number of iterations T for the εFRP to converge. If the convergence was not achieved in T iterations the algorithm then returns the solution for the last solved problem. Therefore, the value δ should be chosen carefully in order not to interfere in the incremental process.

In the case that a final radius ε_f is given and we are only looking for a more sparse solution, the choice of δ must be such that $\delta \leq 2(1-\varepsilon_f/\varepsilon_n)$. If an approximation to the minimal tube containing the data is desired, then this parameter can be chosen according to the quality of the expected approximation. That is, if an α-approximation of the minimum radius (i.e. the final radius is less than $(1+\alpha)\varepsilon^*$, $\alpha \in (0,1)$) is desired, then δ should be set to the value of α. In order to see this, suppose that we have a solution $(w_n, b_n) \in \mathcal{V}(Z_m, \varepsilon_n)$, for some $n \geq 1$, and a new radius is constructed $\varepsilon_{n+1} = (1-\alpha/2)\varepsilon_n$. Suppose that this new radius is such that $\varepsilon_{n+1} < \varepsilon^*$. Then, εFRP will not converge and the last feasible solution found in $\mathcal{V}(Z_m, \varepsilon_n)$ is returned. This final solution has radius ε_n, which satisfies: $\varepsilon_n = \frac{\varepsilon_{n+1}}{(1-\alpha/2)} \leq \frac{\varepsilon^*}{(1-\alpha/2)} < (1+\alpha)\varepsilon^*$.

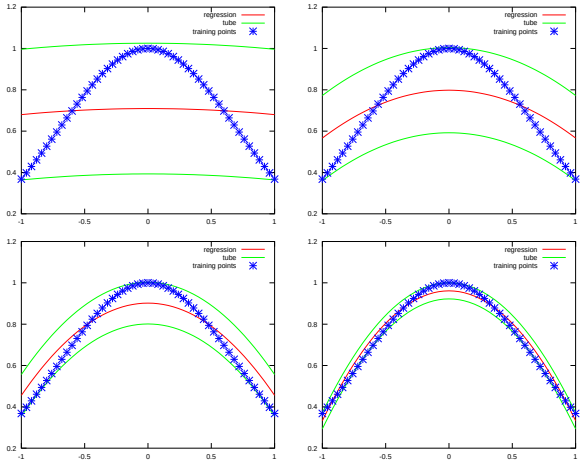


Fig. 1. Incremental strategy algorithm process.

Finally, it is important to mention that each final εFRP solution w is used as a starting point for the next εFRP problem. This strategy has the advantage that εFRP needs to make fewer corrections to satisfy the new radius. In addition, for the first εFRP, we set the initial bias term to $b_0 = \frac{1}{m}\sum_{i=1}^{n} y_i$ in order to have better sparsity results. Algorithm 3 presents the incremental strategy algorithm. We illustrate the incremental strategy process in Figure 1.

Algorithm 3 Incremental strategy algorithm

Input: Z_m, η, δ, ε_0, T
Output: last feasible (w, b) and associated ε_n
$w_0 \leftarrow 0$, $b_0 \leftarrow \frac{1}{m}\sum_{i=1}^{n} y_i$
repeat
  $w_{n+1} \leftarrow \varepsilon$**FRP**$(z_m, w_n, b_n, \eta, \varepsilon_n, T)$
  $\varepsilon_{n+1} = \min\left\{\frac{(\varepsilon_n^+ + \varepsilon_n^-)}{2}, (1-\delta/2)\varepsilon_n\right\}$
until the convergence of εFRP in T iterations is not achieved

VI. Experimental results

In this section, we present some numerical experiments using the proposed method in different artificial datasets. In order to point out different technical aspects of the method, this section was divided in three parts.

A. *Effectiveness in obtaining sparse solutions*

In this group of experiments we tested the εFRP combined with the ISA (called εFRP$_{ISA}$). This is done in order to evaluate the effectiveness in obtaining sparse solutions. Therefore, we also tested the εFRP (without ISA) and SVM-light ([16]) for comparison.

The training sets were generated from a chosen function and polluted with Gaussian noise, $y = f(\cdot) + 0.2\xi$, where $\xi \sim N(0,1)$. For the testing set, we generated new points in the same range using the same function with no noise. This is done in order to observe the ability to recover the true function. The testing set has twice the number of points. Table I presents details of generated data.

Data	m	Function	Range
Exp	101	$y = \exp(-x^2)$	$x \in [-1, 1]$
Sinc	101	$y = \text{sinc}(x)$	$x \in [-\pi, \pi]$

TABLE I. SUMMARY OF GENERATED DATASETS.

All tests in this section were performed in the following way. First, we calculated the range of the target values $r := \max_{i=1,\ldots,m} y_i - \min_{i=1,\ldots,m} y_i$ in the training set. Then, we set the value of ε to $0.1r$ for the εFRP and SVM algorithms. The same value was used as stop criterion for $\varepsilon\text{FRP}_{ISA}$. The learning rate was set to $\eta = 0.01$. The capacity control parameter was set to $C = 100$. For dataset *Exp* dataset we choose a polynomial kernel $k(x_i, x_j) := (s\langle x_i, x_j\rangle + c)^d$, with $d = 2$, $s = 1$ and $c = 0$. For dataset *Sinc* we used a Gaussian kernel $k(x_i, x_j) := exp(-\gamma||x_i - x_j||^2)$ with $\gamma = 1$. In each case, we save the model and use it to perform the tests. The results are presented in Table II.

In order to compare results we present the following criteria: the percentage of support vectors ($\%sv$) and the norm of the solution. Also, for εFRP and $\varepsilon\text{FRP}_{ISA}$, we present the total number of iterations (T) and updates (up). As error criteria, we measure the *root mean squared error* (RMSE) on training and testing sets.

	$\%sv$	norm	T/up	Training RMSE	Testing RMSE
Exp					
εFRP	88.12	0.45083	8082/144055	0.17956	0.02712
$\varepsilon\text{FRP}_{ISA}$	57.42	0.48948	11028/121655	0.17974	0.02681
SVM-light	52.47	0.72483	-/-	0.18008	0.04158
Sinc					
εFRP	73.27	4.10598	8221/70484	0.17321	0.07384
$\varepsilon\text{FRP}_{ISA}$	37.62	3.54023	8283/61339	0.17211	0.05920
SVM-light	35.64	3.70680	-/-	0.17344	0.06727

TABLE II. RESULTS REGARDING THE OBTAINING OF SPARSE SOLUTIONS.

First, notice that the ISA proved to be a good approach to maintain the sparsity of the solution. In all cases, the $\varepsilon\text{FRP}_{ISA}$ obtained fewer number of support vectors than εFRP leading to a more sparse solution. Notice that the percentage of support vectors found by $\varepsilon\text{FRP}_{ISA}$ is slightly greater than what was obtained by SVM-light.

The second observation is that when using ISA the εFRP has a higher number of iterations as expected, once it starts with a larger value for ε, however, notice that it needs fewer number of updates. This indicates that the algorithm updates only the most important points.

Considering the quality of the fit, εFRP and $\varepsilon\text{FRP}_{ISA}$ achieved good results on the training and testing sets in comparison with SVM-light.

B. Effectiveness in obtaining the minimal tube

In this section, we tested the $\varepsilon\text{FRP}_{ISA}$ without margin flexibility in order to find an approximation to the minimal tube containing the data. The tube achieved by $\varepsilon\text{FRP}_{ISA}$ was then used to run SVM-light for comparison. It is important to mention that this minimal tube cannot be found using the standard SV-regression approach. Also, we implemented and tested the NORMA algorithm [8] for regression.

In order to perform the tests, we used the same datasets presented in the previous section. We set the number of iterations to $T = 1000$ and the learning rate to $\eta = 0.01$ for $\varepsilon\text{FRP}_{ISA}$ and NORMA. Also, we set the regularization parameter $\beta = 0.01$ for NORMA. We used two different values for the parameter ν to run NORMA. In order to avoid the margin flexibility we set the parameter C to a large value on SVM-light. The kernel functions were set in a similar fashion of previous section. In order to compare the results we present the following: number of support vectors (sv), the radius of the tube (ε) and the norm of the solution. In addition, we present the RMSE criterion measured on the training and testing sets. Results are presented in Table III.

	sv	ε	norm	Training RMSE	Testing RMSE
Exp					
$\varepsilon\text{FRP}_{ISA}$	8	0.3967	0.2613	0.18959	0.05585
NORMA (0.1)	19 (11)	0.3100	0.2971	0.18675	0.06257
NORMA (0.2)	26 (20)	0.2600	0.3953	0.18396	0.02767
SVM-light	3	0.3967	0.5068	0.19127	0.05841
Sinc					
$\varepsilon\text{FRP}_{ISA}$	16	0.3540	1.9114	0.18993	0.07545
NORMA (0.1)	25 (12)	0.3200	1.1951	0.20303	0.07601
NORMA (0.2)	47 (13)	0.3000	1.3210	0.22478	0.10871
SVM-light	10	0.3540	1.2626	0.19077	0.08690

TABLE III. RESULTS REGARDING THE OBTAINING OF MINIMAL TUBE CONTAINING THE DATA.

The first observation is that NORMA did not yield a sparse solution when compared to $\varepsilon\text{FRP}_{ISA}$ and SVM-light. In addition, when NORMA achieved a smaller radius value, it left points outside of the tube. The values in parenthesis near the support vectors for NORMA represent the number of points left outside of the tube. Considering the quality of the fit, the $\varepsilon\text{FRP}_{ISA}$ shows good results comparing with NORMA and SVM-light.

C. Effectiveness in large datasets

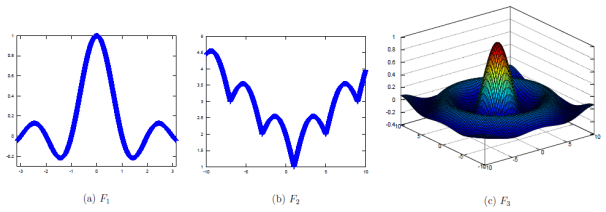
In this section we tested the $\varepsilon\text{FRP}_{ISA}$ in large data sets. This is done in order to compare the run time of $\varepsilon\text{FRP}_{ISA}$ and SVM-light.

The training sets were generated from a chosen mathematical function and the target values were polluted with Gaussian noise in such way that $y = f(\cdot) + 0.1\xi$, where $\xi \sim N(0, 1)$. A summary about the generated data is presented in Table IV. Also, Figure 2 illustrates the relation between training points and target values for used functions.

Data	m	Function	Range
F_1	10006	$y = \text{sinc}(x)$	$x \in [-\pi, \pi]$
F_2	10001	$y = \left\lvert \frac{x-1}{4} \right\rvert + \left\lvert \text{sen}(\pi(1 + \frac{x-1}{4})) \right\rvert + 1$	$x \in [-10, 10]$
F_3	10001	$y = \text{sinc}(\sqrt{x_1^2 + x_2^2})$	$x_i \in [-10, 10]$

TABLE IV. SUMMARY OF GENERATED DATASETS.

The experiments were performed in the following fashion: For the $\varepsilon\text{FRP}_{ISA}$ we defined as stop criterion the radius $\varepsilon_{F_1} =$



(a) F_1 (b) F_2 (c) F_3

Fig. 2. Dependent relations between points and targets for generated functions.

0.1, $\varepsilon_{F_2} = 0.2$ and $\varepsilon_{F_3} = 0.1$. Also, we set the learning rate to $\eta_{F_1} = 0.01$, $\eta_{F_2} = 0.03$ and $\eta_{F_3} = 0.02$. For all datasets we used the final radius obtained with εFRP$_{ISA}$ to run SVM-light, ir order to obtain the run time to achieve the same final radius. The capacity control parameter was set to $C = 10$ for all experiments. Also we set the Gaussian kernel parameter to $\gamma = 1.0$.

In order to compare the results we present the following data obtained from the experiment: percentage of support vectors ($\%sv$); run time (rt); the tube radius (ε); solution norm (norm); for εFRP$_{ISA}$ we also present the total number of iterations (T) and the total number of updates (up). The results are presented in Table V.

	$\%sv$	T/up	ε	norm	rt
F_1					
εFRP$_{ISA}$	0.27	103/138	0.49859	0.71759	8.58
εFRP$_{ISA}$	0.33	163/222	0.39910	0.93835	8.66
εFRP$_{ISA}$	0.65	457/1510	0.29898	1.37356	9.63
εFRP$_{ISA}$	5.99	1470/40446	0.19904	5.19886	37.24
εFRP$_{ISA}$	17.97	2103/138435	0.14974	10.29991	106.36
εFRP$_{ISA}$	38.46	2914/381342	0.09949	18.64224	277.73
SVM-light	33.20	-/-	0.09949	1.45873	791.48
F_2					
εFRP$_{ISA}$	0.69	67/142	0.98938	2.12675	11.26
εFRP$_{ISA}$	0.75	118/205	0.68016	2.54115	11.32
εFRP$_{ISA}$	0.85	167/265	0.49988	2.93399	11.39
εFRP$_{ISA}$	1.73	412/1498	0.29944	3.66530	12.37
εFRP$_{ISA}$	3.91	678/6132	0.24400	4.80350	15.86
εFRP$_{ISA}$	10.34	906/23449	0.19846	7.81095	28.77
SVM-light	5.71	-/-	0.19846	4.54684	32.43
F_3					
εFRP$_{ISA}$	0.27	75/88	0.49038	0.79215	12.90
εFRP$_{ISA}$	0.33	107/127	0.39383	1.03416	12.93
εFRP$_{ISA}$	0.66	264/643	0.29984	1.38644	13.37
εFRP$_{ISA}$	8.23	1286/24495	0.19981	5.57679	31.92
εFRP$_{ISA}$	19.92	1884/82775	0.14817	11.28930	75.07
εFRP$_{ISA}$	44.96	2462/238715	0.09543	21.20861	190.40
SVM-light	35.55	-/-	0.09543	1.73510	1836.48

TABLE V. RESULTS COMPARING THE εFRP$_{ISA}$ AND SVM-LIGHT IN LARGE DATASETS.

We present incremental solutions obtained by εFRP$_{ISA}$ during its running time. This is done in order to show that one has a feasible solution every time a call to FRP algorithm returns. This technical feature might be interesting for time dependent applications, where it might not be important to obtain an exact solution.

In addition, notice that εFRP$_{ISA}$ obtained a solution faster than the SVM-light for every dataset. Also, notice that as the algorithm approaches the final fixed radius value, the FRP algorithm takes more time to converge. One explanation for this behavior is that as the radius shrinks, the version space becomes more restricted and a solution is more difficult to achieve.

VII. CONCLUSIONS

In this paper, we introduced a new online learning method for regression based on the stochastic gradient descent coupled with the incremental strategy algorithm. The method uses the ε-insensitive loss function which enables the use of the support vector approach obtaining sparse solutions. When formulated in dual variables, it allows the introduction of kernels and margin flexibility. The algorithm is entirely based on the Perceptron which makes it simple to understand and implement. The experimental results show that the algorithm works well in comparisons with the standard support vector regression approach.

ACKNOWLEDGMENTS

The authors would like to thank CNPq, CAPES and InWeb for financial support.

REFERENCES

[1] P. J. Huber, "Robust statistics: A review," *The Annals of Mathematical Statistics*, vol. 43, pp. 1041–1067, 1972.

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

[3] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, 1992, pp. 144–152.

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

[5] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, pp. 293–300, 1999.

[6] C. Gentile, "A new approximate maximal margin classification algorithm," *Journal of Machine Learning Research*, vol. 2, pp. 213–242, 2001.

[7] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," *Machine Learning*, vol. 46, pp. 361–387, 2002.

[8] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2165–2176, 2004.

[9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[10] S. S. Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 807–814.

[11] S. C. Leite and R. Fonseca Neto, "Incremental margin algorithm for large margin classifiers," *Neurocomputing*, vol. 71, pp. 1550–1560, 2008.

[12] J. Bi and K. Bennet, "A geometric approach to support vector regression," *Neurocomputing*, vol. 55, pp. 79–108, 2003.

[13] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958.

[14] K. P. Bennett and O. L. Mangasarian, "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[15] C. Campbell, "Kernel methods: A survey of current techniques," *Neurocomputing*, vol. 48, pp. 63–84, 2002.

[16] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in kernel methods*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, 1999, pp. 169–184.