Large-Scale Sonar Target Detection with ℓ_1-Norm SV Regression based on Unfeasible Interior Point Methods

Pablo Rivas-Perea, Jose G. Rosiles

Department of Electrical and Computer Engineering
The University of Texas El Paso
500 W. University Ave. El Paso, TX 79902

Abstract

Support Vector Machines (SVMs) have become one of the most popular supervised learning-machines in the statistical pattern recognition area. They are used for classification (*i.e.* SVM) and regression analysis (*i.e.* Support Vector Regression, SVR). However, when the number of samples available to model an SVM/SVR problem supersedes the computational resources (*i.e.* large-scale problems where the number of dimensions or samples are in the order of millions) the traditional methods fail in finding the optimal solution to a classification problem based on regression models. The reason for the typical failures is that the solution finding process involves high-dimensional vector operations. The aim of this research is to overcome the natural limitation of large-scale problems particular to SVR using an efficient convex linear programming framework. We propose a sequential decomposition method based on a linear programming support vector regression (SLP-SVR) approach. The proposed scheme uses an interior point method (IPM) to solve a sequence of smaller LP optimization sub-problems given by the proposed decomposition strategy. Then we perform chunking (preserve the the support vectors) of the sub-problem at each iterate. We take advantage of the quadratic rate of convergence of IPM on the proposed LP-SVR method to finds the global solution to the regression/classification problem in few iterates. Experiments were performed to solve the large-scale sonar mine-rocks detection problem show fast rate of convergence, and very good performance when compared to other approaches such as neural networks and PCA-based methods. The proposed linear programming formulation is efficient from the mathematical point of view and the sequential decomposition strategy using IPM poses fast rate of convergence to make the proposed model also efficient from the computational point of view.

Keywords: Support vector machines, support vector regression, linear programming, interior point methods, sequential optimization.
2010 MSC: 62J02, 62H30, 68T05, 68T10, 90C05, 90C51.

1. Introduction

Real-life problems are difficult to model by finding a set of rules or a linear relationship. Such is the case of sonar-based rescue or military missions as illustrated in Figure 1. The problem we address here is the large-scale automatic detection of mines based on millions of sonar frequency-based data. The mines subject of this study are typically cylindrically shaped as shown in Figure 2. Typically researchers and engineers can design methods to produce an image of sea surface and a qualified technician may observe and detect possible threats. Figure 3 shows an example of this kind of technology. In our study we use frequency-based information at sonar range. The dataset of information we use comes from Gorman's *et.al.* [1] previous work using neural networks to detect mines. However, in our research we use a novel concept in the statistical and machine learning community.

Statistical learning theory concepts introduced by Vapnik *et.al.* [2, 3] leaded to Support Vector Machines for Regression (SVR), derived from the concepts of *structural risk minimization* and *empirical risk minimization*. The training (setup) of an SVR is equivalent to solve a Quadratic Program (QP) with linear constraints [4]. The number of data points (instrument readings, calibration data) is equivalent to the number of variables for the QP problem. Hence, problems arise as the training data set size grows to millions, as is the case of real-life engineering applications, such as instrument calibration, military equipment testing and certification, etc.

Current approaches [4, 5, 6, 7, 8, 9] assume that the following conditions are satisfied: (i) the number of data points available to train the SVR is manageable; (ii) the total number of support vectors (relevant information) is manageable. However, these conditions do not hold in some engineering applications. Particularly, we aim to address condition (i) where the number of samples is in the order of millions.

We propose a training algorithm that takes advantage of Linear Programming (LP) efficiency. We pose the SVR optimization problem as an LP problem rather than a QP problem. We use an interior point methods (IPM) to solve a sequence of LP problems that converge quadratically to the solution. Then, we

Email addresses: privas@miners.utep.edu (Pablo Rivas-Perea), grosiles@utep.edu (Jose G. Rosiles)

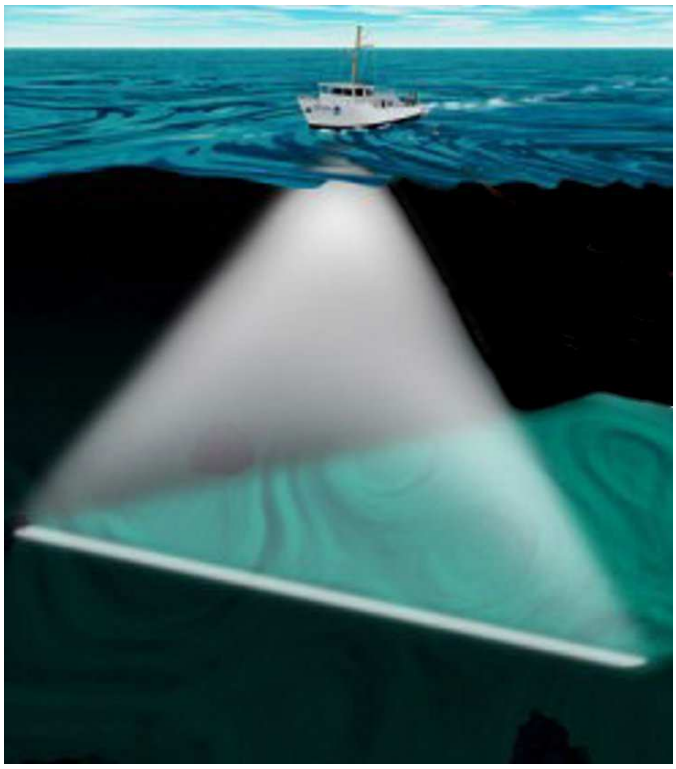

Figure 1: Sonar instrument on-board a ship sensing the seafloor.

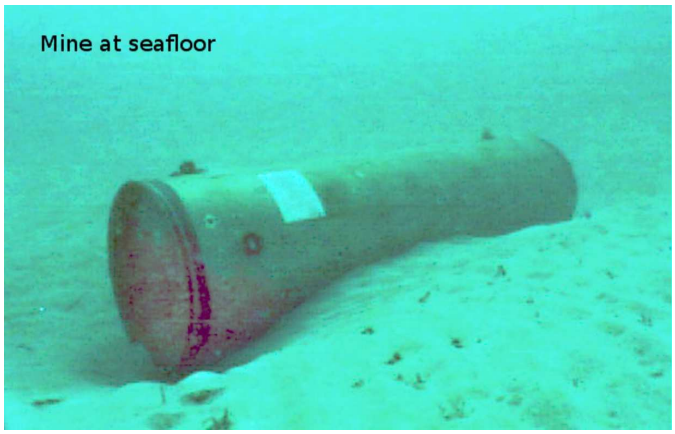

Figure 2: Real picture of a cylindrical mine at a sandy seafloor.

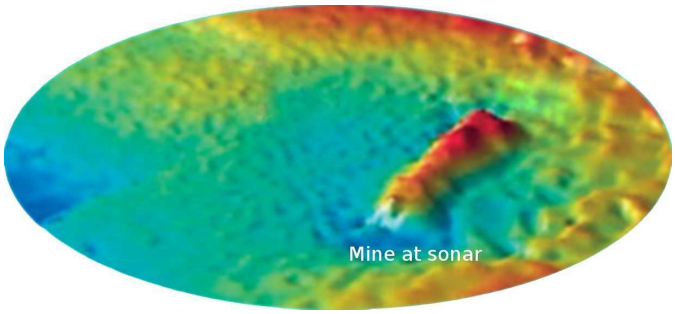

Figure 3: A typical display or image produced by sonar-based algorithms.

address the problem of sonar target detection using millions of sonar instrument readings.

This paper is organized as follows: the fundamentals of SVR are explained in Section 2. The linear programming formulation for SVR is explained in Section 3. In Section 4 we describe the proposed strategy for sonar large-scale target detection. Finally, we draw conclusions in Section 5.

2. Support Vector Regression Theoretical Basis

In 1995, Vapnik *et al.* [2, 3] and later Smola *et al.* [10, 11] explored and developed the SVM approach for regression problems. This approach is commonly known as Support Vector Regression (SVR), which increased the application range of SVMs since SVRs can also perform multi-class pattern recognition. This type of machine is typically formulated using quadratic optimization under the umbrella of convex optimization.

Let's consider the linear regression case where the dependency of a scalar observable d on a regressor $\mathbf{x}$ is denoted as

$$d = \mathbf{w}^T \mathbf{x} + b, \tag{1}$$

where the parameter vector $\mathbf{w}$ and the bias b are the unknowns. The problem is to estimate $\mathbf{w}$ and b given the training samples $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$ where the vector elements $\mathbf{x}_i$ are assumed to be statistically independent and identically distributed (iid). The problem formulated by Vapnik is aimed to minimize the *risk functional*

$$\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N} |y_i - d_i|_\epsilon \tag{2}$$

which can be expressed as an optimization problem in its primal form as follows [12]:

$$\begin{array}{lll} \min_{\mathbf{w},L_\epsilon} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N} |y_i - d_i|_\epsilon & \\ \text{s.t.} & \left\{ \begin{array}{rcl} d_i - y_i & \leq & \epsilon + \xi_i \\ y_i - d_i & \leq & \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq & 0 \end{array} \right. \\ \text{for} & i = 1, 2, \cdots, N. & \end{array}$$

where the summation in the cost function accounts for the ϵ-insensitive training error, which forms a tube where the solution is allowed to be defined without penalization, as shown in Figure 4. $C > 0$ is a constant describing the trade off between

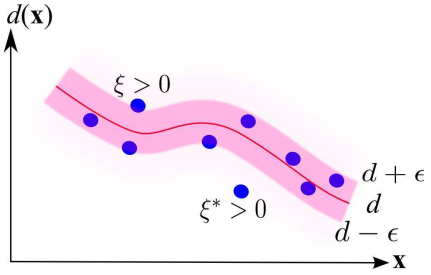

Figure 4: The epsilon tube where the solution is is allowed to be defined without penalization.

the training error and the penalizing term $\|\mathbf{w}\|_2^2$. The variable y_i is the estimator output produced in response to the input $\mathbf{x}_i$, that is $y_i = \mathbf{w}^T\mathbf{x}_i + b$. The variables ξ_i and ξ_i^* are two sets of nonnegative slack variables that describe the ϵ-insensitive loss function denoted as

$$L_\epsilon(d,y) = \begin{cases} |d-y| - \epsilon & \text{for } |d-y| \geq \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where ϵ is a prescribed parameter. This loss function is illustrated in Figure 5 in relation with the data shown in Figure 4.

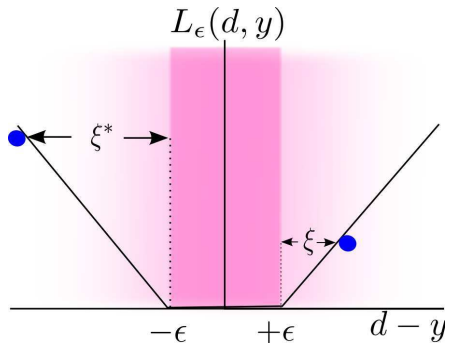

Figure 5: The ϵ-insensitive loss function. Solution data-points are penalized as they go apart from the limits specified by the parameter ϵ.

The objective function in the primal can be rewritten in terms of the slack variables ξ and ξ^*, by observing the restrictions of the primal and the definition of the ϵ-insensitive function, and thus defining $\xi = d_i - y_i - \epsilon$ and $\xi^* = y_i - d_i - \epsilon$. Then one obtain another common version of the primal problem as follows:

$$\begin{array}{lll} \min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^N \left(\xi_i + \xi_i^*\right) & \\ \text{s.t.} & \left\{ \begin{array}{rcl} \mathbf{w}^T\mathbf{x}_i + b - d_i & \leq & \epsilon + \xi_i \\ d_i - \mathbf{w}^T\mathbf{x}_i - b & \leq & \epsilon + \xi_i^* \\ \boldsymbol{\xi}, \boldsymbol{\xi}^* & \geq & \mathbf{0} \end{array} \right. \\ \text{for} & i = 1, 2, \cdots, N. & \end{array} \tag{4}$$

The SVR problem in the dual is obtained by the Lagrange multipliers method. The result is the maximization problem

$$\begin{array}{lll} \max_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\gamma},\boldsymbol{\gamma}^*} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^N \left(\xi_i + \xi_i^*\right) & \\ & -\sum_{i=1}^N \left(\gamma_i\xi_i + \gamma_i^*\xi_i^*\right) & \\ & -\sum_{i=1}^N \alpha_i \left(\mathbf{w}^T\mathbf{x}_i + b - d_i + \epsilon + \xi_i\right) & \\ & -\sum_{i=1}^N \alpha_i^* \left(d_i - \mathbf{w}^T\mathbf{x}_i - b + \epsilon + \xi_i^*\right) & \\ \text{s.t.} & \left\{ \begin{array}{rcl} \alpha_i\left(\epsilon + \xi_i + d_i - y_i\right) & = & 0 \\ \alpha_i^*\left(\epsilon + \xi_i^* + d_i - y_i\right) & = & 0 \\ (\alpha_i - C)\xi_i & = & 0 \\ \left(\alpha_i^* - C\right)\xi_i^* & = & 0 \\ \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\gamma}, \boldsymbol{\gamma}^* & \geq & \mathbf{0} \end{array} \right. \\ \text{for} & i = 1, 2, \cdots, N. & \end{array} \tag{5}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^*$, $\boldsymbol{\gamma}$, and $\boldsymbol{\gamma}^*$ are the nonnegative Lagrange multipliers.

Problems (4) and (5) solve the linear regression problems, and for the non-linear regression case it simply follows to introduce a kernel function $k(\cdot,\cdot)$ mapping in the following way:

$$\mathbf{x}_i^T\mathbf{x}_j = k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j), \tag{6}$$

$$\mathbf{x}_i = k\left(\mathbf{x}_i, \cdot\right) = \phi(\mathbf{x}_i), \tag{7}$$

where the map

$$\phi : \mathcal{X} \mapsto \mathcal{H} \tag{8}$$

is known as feature map from the data space $\mathcal{X}$ into the feature space $\mathcal{H}$. The feature space is assumed to be a Hilbert space of real valued functions defined on $\mathcal{X}$. The typical kernel functions are as follows:

$$\text{Polynomial} \quad : \quad (\mathbf{x}_i^T\mathbf{x}_j + 1)^p, \tag{9}$$

$$\text{Radial} \quad : \quad e^{-\frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}, \tag{10}$$

$$\text{Sigmoidal} \quad : \quad \tanh(\kappa_1\mathbf{x}_i^T\mathbf{x}_j + \kappa_2). \tag{11}$$

The kernel $k\left(\mathbf{x}_i, \mathbf{x}_j\right)$ may also be referred as the ij-th element of the symmetric $N \times N$ matrix

$$\mathbf{K} = \left\{k\left(\mathbf{x}_i, \mathbf{x}_j\right)\right\}_{i,j=1}^N. \tag{12}$$

The matrix $\mathbf{K}$ is called "kernel matrix." It is positive definite since it satisfies the condition $\mathbf{a}^T\mathbf{K}\mathbf{a} \geq 0$ for any real valued vector $\mathbf{a}$ of dimension compatible with $\mathbf{K}$. The matrix notation of $\mathbf{K}$ is specially useful when posing the optimization problem in matrix-vector form.

Then, introducing introducing kernel functions for the primal case, the sole modification is on the restrictions which are redefined as

$$\mathbf{w}^T k(\mathbf{x}_i, \cdot) + b - d_i \leq \epsilon + \xi_i \tag{13a}$$

$$d_i - \mathbf{w}^T k(\mathbf{x}_i, \cdot) - b \leq \epsilon + \xi_i^* \tag{13b}$$

and for the dual problem the objective function is redefined as

$$\begin{array}{ll} \max\limits_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\gamma},\boldsymbol{\gamma}^*} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^N \left(\xi_i + \xi_i^*\right) - \sum_{i=1}^N \left(\gamma_i\xi_i + \gamma_i^*\xi_i^*\right) \\ & -\sum_{i=1}^N \alpha_i \left(\mathbf{w}^T k(\mathbf{x}_i, \cdot) + b - d_i + \epsilon + \xi_i\right) \\ & -\sum_{i=1}^N \alpha_i^* \left(d_i - \mathbf{w}^T k(\mathbf{x}_i, \cdot) - b + \epsilon + \xi_i^*\right) \end{array} \tag{14}$$

for $i, j = 1, 2, \cdots, N$.

3. Formulation of a Linear Programming SVR

Linear Programs are those problems that can be stated in the *canonical* form as:

$$\begin{array}{lll} \min_{\mathbf{z}} & \mathbf{c}^T\mathbf{z} & \\ \text{s.t.} & \left\{ \begin{array}{rcl} \mathbf{A}\mathbf{z} & = & \mathbf{b} \\ \mathbf{z} & \geq & \mathbf{0}. \end{array} \right. \end{array} \tag{15}$$

where $\mathbf{z} \in \Re^n$ is a vector containing the unknowns, $\mathbf{c} \in \Re^n$ and $\mathbf{b} \in \Re^m$ are vectors of known parameters, and $\mathbf{A} \in \Re^{m \times n}$ is a matrix of known coefficients associated to $\mathbf{z}$ in a linear relationship.

Let us re-state the original support vector regression problem (4) using the ℓ_1-norm. The decision function can be expressed by means of $\mathbf{x}$ as $d(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + b$. Thus, using Mercer's theorem[13, 14], we can replace the expression $\mathbf{w}^T\mathbf{x}_i + b$, by the following kernel expansion

$$d(\mathbf{x}_j) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_j, \mathbf{x}_i) + b, \tag{16}$$

where $\boldsymbol{\alpha}, b$ can take on any real value, $k(\cdot,\cdot)$ is a valid kernel function such as (9)-(11). Then, taking advantage of the geometric analysis, which states that $\xi_i\xi_i^* = 0$ in regression problems, it is sufficient to just introduce ξ_i, then the problem becomes

$$\begin{array}{lll} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} & \|\alpha\|_1 + 2C\sum_{i=1}^N \xi_i & \\ \text{s.t.} & \left\{ \begin{array}{rcl} d_j - \sum_{i=1}^N \alpha_i k(\mathbf{x}_j,\mathbf{x}_i) - b & \leq & \epsilon + \xi_j \\ \sum_{i=1}^N \alpha_i k(\mathbf{x}_j,\mathbf{x}_i) + b - d_j & \leq & \epsilon + \xi_j \\ \boldsymbol{\xi} & \geq & \mathbf{0} \end{array} \right. \\ \text{for} & j = 1, 2, \cdots, N. & \end{array} \tag{17}$$

Then, since the requirement of a linear program is to have the unknowns greater than zero, we must then decompose such variables in their positive and negative part. Therefore, we denote $\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-$, and $b = b^+ - b^-$. Then, in order to pose the problem as a linear program in its canonical form and in order to use an interior point method solver, problem (17) must have no inequalities; thus, we need to add a slack variable $\mathbf{u}$, which results on the following problem

$$\begin{array}{lll} \min\limits_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-,b^+,b^-,\boldsymbol{\xi},\mathbf{u}} & \sum_{i=1}^N \left(\alpha_i^+ + \alpha_i^- + 2C\xi_i\right) & \\ \text{s.t.} & \left\{ \begin{array}{rcl} -\sum_{i=1}^N(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j,\mathbf{x}_i) & & \\ \quad -b^+ + b^- - \xi_j + u_j & = & \epsilon - d_j \\ \sum_{i=1}^N(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j,\mathbf{x}_i) & & \\ \quad +b^+ - b^- - \xi_j + u_j & = & \epsilon + d_j \\ \alpha_j^+, \alpha_j^-, b^+, b^-, \xi_j, u_j & \geq & 0 \end{array} \right. \\ \text{for} & j = 1, 2, \cdots, N. & \end{array} \tag{18}$$

which is finally an acceptable problem to be solved. Note that (18) allows us to define the following equalities:

$$\mathbf{A} = \begin{pmatrix} -\mathbf{K} & \mathbf{K} & -1 & 1 & -\mathbf{I} & \mathbf{I} \\ \mathbf{K} & -\mathbf{K} & 1 & -1 & -\mathbf{I} & \mathbf{I} \end{pmatrix}, \tag{19}$$

$$\mathbf{b} = \begin{pmatrix} \mathbf{1}\epsilon - \mathbf{d} \\ \mathbf{1}\epsilon + \mathbf{d} \end{pmatrix}, \tag{20}$$

$$\mathbf{z} = \begin{pmatrix} \boldsymbol{\alpha}^+ & \boldsymbol{\alpha}^- & b^+ & b^- & \boldsymbol{\xi} & \mathbf{u} \end{pmatrix}^T, \tag{21}$$

$$\mathbf{c} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{2C} & \mathbf{0} \end{pmatrix}^T. \tag{22}$$

Then, we have posed the problem in the LP canonical form (15). Note that the problem has $(4N+2)$ variables and $2N$ constraints.

This is a definition more accurate than the one described by Lu *et.al.* in late 2009 [15], and also it is a generalized and extended version of the LP-SVM work presented by Torii *et.al.* in early 2009 [16] and by Zhang in early 2010 [17].

By introducing the Lagrange multipliers $(\boldsymbol{\lambda}, \mathbf{s})$ into the primal problem above, we obtain the following dual:

$$\begin{array}{lll} \max_{\boldsymbol{\lambda}} & \mathbf{b}^T\boldsymbol{\lambda} & \\ \text{s.t.} & \left\{ \begin{array}{rcl} \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{s} & = & \mathbf{c} \\ \mathbf{s} & \geq & \mathbf{0}. \end{array} \right. \end{array} \tag{23}$$

where $\boldsymbol{\lambda}$ is a vector of dual variables defined over $\Re^{2N}$, and $\mathbf{s}$ is a slack variable vector in $\Re^{4N+2}$.

The solution to the primal problem is denoted as $\mathbf{z}^*$, and the solution to the dual problem is denoted as $(\boldsymbol{\lambda}^*, \mathbf{s}^*)$. The duality theorem states that $\mathbf{c}^T\mathbf{z}^* = \mathbf{b}^T\boldsymbol{\lambda}^*$, which means that the solution $\mathbf{z}^*$ also solves the dual, and the solution of the dual $(\boldsymbol{\lambda}^*, \mathbf{s}^*)$ also solves the primal.

The Karush-Kuhn-Tucker (KKT) conditions, are a set of equalities and inequalities that for this problem are necessary and sufficient conditions to establish optimality of the model. The KKT conditions for the LP problem are the following:

$$\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, \tag{24a}$$

$$\mathbf{A}\mathbf{z} = \mathbf{b}, \tag{24b}$$

$$z_i s_i = 0, \tag{24c}$$

$$(\mathbf{z}, \mathbf{s}) \geq \mathbf{0}, \tag{24d}$$

$$i = 1, 2, ..., 4N + 2,$$

where the equality $z_i s_i$ implies that one of both variables must be zero. This equality will be referred to as the *complementarity condition*. Note that the problem depends on the variables $(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s})$, and if the set of solutions $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*)$ satisfy all the conditions, the problem is said to be solved. The vector $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*)$ is known as a *primal-dual solution*.

The dual problem (23) in its extended form is

$$\begin{array}{lll} \min\limits_{\boldsymbol{\lambda},\mathbf{s}} & \sum_{i=1}^N \lambda_i(\epsilon - d_i) + \sum_{i=1}^N \lambda_{i+N}(\epsilon + d_i) & \\ \text{s.t.} & \left\{ \begin{array}{rcl} \sum_{i=1}^N \lambda_{j+N}k(\mathbf{x}_j,\mathbf{x}_i) - \sum_{i=1}^N \lambda_j k(\mathbf{x}_j,\mathbf{x}_i) + s_j & = & 1_j \\ \sum_{i=1}^N \lambda_j k(\mathbf{x}_j,\mathbf{x}_i) - \sum_{i=1}^N \lambda_{j+N}k(\mathbf{x}_j,\mathbf{x}_i) + s_{j+N} & = & 1_j \\ \sum_{i=1}^N \lambda_{i+N} - \lambda_i + s_{2N+1} & = & 0 \\ \sum_{i=1}^N \lambda_i - \lambda_{i+N} + s_{2N+2} & = & 0 \\ -\lambda_j - \lambda_{j+N} + s_{j+2N+2} & = & 2C \\ \lambda_j + \lambda_{j+N} + s_{j+3N+2} & = & 0 \\ \mathbf{s} & \geq & \mathbf{0} \end{array} \right. \\ \text{for} & j = 1, 2, \cdots, N. & \end{array} \tag{25}$$

The KKT conditions (24) in their extended form are

$$\sum_{i=1}^{N} \lambda_{j+N}k(\mathbf{x}_j,\mathbf{x}_i) - \sum_{i=1}^{N} \lambda_j k(\mathbf{x}_j,\mathbf{x}_i) + s_j = 1_j \tag{26a}$$

$$\sum_{i=1}^{N} \lambda_j k(\mathbf{x}_j,\mathbf{x}_i) - \sum_{i=1}^{N} \lambda_{j+N}k(\mathbf{x}_j,\mathbf{x}_i) + s_{j+N} = 1_j \tag{26b}$$

$$\sum_{i=1}^{N} \lambda_{i+N} - \lambda_i + s_{2N+1} = 0 \tag{26c}$$

$$\sum_{i=1}^{N} \lambda_i - \lambda_{i+N} + s_{2N+2} = 0 \tag{26d}$$

$$-\lambda_j - \lambda_{j+N} + s_{j+2N+2} = 2C \quad (26e)$$
$$\lambda_j + \lambda_{j+N} + s_{j+3N+2} = 0_j \quad (26f)$$
$$-\sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) - b^+ + b^- - \xi_j + u_j = \epsilon - d_j \quad (26g)$$
$$\sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) + b^+ - b^- - \xi_j + u_j = \epsilon + d_j \quad (26h)$$
$$s_j\alpha_j^+ + s_{j+N}\alpha_j^- + s_{2N+1}b^+ + s_{2N+2}b^- + s_{j+2N+2}\xi_j + s_{j+3N+2}u_j = 0 \quad (26i)$$
$$s_i, \alpha_j^+, \alpha_j^-, b^+, b^-, \xi_j, u_j \geq 0 \quad (26j)$$

for all $j = 1, 2, \cdots, N$, and $i = 1, 2, \cdots, 4N + 2$

where the equalities (26a)-(26f) come from the expansion of (24a), the equalities (26g)-(26h) come from the expansion of (24b), the complementarity condition (24c) results in the expansion (26i), and the inequalities in (24d) are the ones that force positivity in (26j).

4. Sonar Large-Scale Target Detection using LP-SVR

In this section we explain the proposed solution of (18), (25), and (26a)-(26f) for the large scale sonar-based mines detection problem. The idea is to divide the training set $\{\mathbf{x}_i, d_i\}_{i\in T}$, for $T = \{1, 2, ..., N\}$ in two subsets. The first is the *working set* $\{\mathbf{x}_i, d_i\}_{i\in B}$, for $B \subset T$, $|B| \ll |T|$. The second set is the remaining of the data $\{\mathbf{x}_i, d_i\}_{i\in M}$, for $M \subset T$, $|B| \ll |M| < |T|$, and $B + M = T$. Then, it follows to define

$$\boldsymbol{\alpha}_B^+, \boldsymbol{\alpha}_B^- = \alpha_i^+, \alpha_i^-, \text{ for all } i \in B,$$
$$\boldsymbol{\alpha}_M^+, \boldsymbol{\alpha}_M^- = \alpha_j^+, \alpha_j^-, \text{ for all } j \in M,$$

respectively. In this proposed formulation, the set of remaining variables are fixed $\alpha_j^+, \alpha_j^- = 0$, for all $j \in M$, This follows the idea of the well known QP-SVM methods, such as [4, 5, 6, 7, 8, 9]. Under this formulation, the subproblem to solve becomes

$$\min_{\boldsymbol{\alpha}_B^+, \boldsymbol{\alpha}_B^-, b^+, b^-, \boldsymbol{\xi}_B, \mathbf{u}_B} \quad \sum_{i\in B}\left(\alpha_i^+ - \alpha_i^- + 2C\xi_i\right)$$
$$\text{s.t.} \quad \begin{cases} -\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) \\ \quad -b^+ + b^- - \xi_j + u_j &= \epsilon - d_j \\ \sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) \\ \quad +b^+ - b^- - \xi_j + u_j &= \epsilon + d_j \\ \alpha_j^+, \alpha_j^-, b^+, b^-, \xi_j, u_j &\geq 0 \\ \text{for all } j \in B \end{cases} \quad (27)$$

Then, we can solve this problem by finding a feasible $\mathbf{z}^* = [\boldsymbol{\alpha}_B^{+*}, \boldsymbol{\alpha}_B^{-*}, b^{+*}, b^{-*}, \boldsymbol{\xi}_B^*, \mathbf{u}_B^*]$. Using the same argument, we find the corresponding dual problem to be

$$\min_{\boldsymbol{\lambda}, \mathbf{s}} \quad \sum_{i\in B}\lambda_i(\epsilon - d_i) + \sum_{i\in B}\lambda_{i+|B|}(\epsilon + d_i)$$
$$\text{s.t.} \quad \begin{cases} \sum_{i\in B}\lambda_{i+|B|}k(\mathbf{x}_j, \mathbf{x}_i) - \sum_{i\in B}\lambda_i k(\mathbf{x}_j, \mathbf{x}_i) + s_j &= 1_j \\ \sum_{i\in B}\lambda_i k(\mathbf{x}_j, \mathbf{x}_i) - \sum_{i\in B}\lambda_{i+|B|}k(\mathbf{x}_j, \mathbf{x}_i) \\ \quad +s_{j+|B|} &= -1_j \\ \sum_{i\in B}\lambda_{i+|B|} - \lambda_i + s_{2|B|+1} &= 1 \\ \sum_{i\in B}\lambda_i - \lambda_{i+|B|} + s_{2|B|+2} &= -1 \\ -\lambda_i - \lambda_{i+|B|} + s_{j+2|B|+2} &= 2C_i \\ \lambda_i + \lambda_{i+|B|} + s_{j+3|B|+2} &= 1_i \\ \mathbf{s} &\geq \mathbf{0} \\ \text{for all } j \in B \end{cases} \quad (28)$$

and the KKT conditions are rewritten as follows

$$\sum_{i\in B}\lambda_{i+|B|}k(\mathbf{x}_j, \mathbf{x}_i) - \sum_{i\in B}\lambda_i k(\mathbf{x}_j, \mathbf{x}_i) + s_j = 1_j \quad (29a)$$
$$\sum_{i\in B}\lambda_i k(\mathbf{x}_j, \mathbf{x}_i) - \sum_{i\in B}\lambda_{i+|B|}k(\mathbf{x}_j, \mathbf{x}_i) + s_{j+|B|} = -1_j \quad (29b)$$
$$\sum_{i\in B}\lambda_{i+|B|} - \lambda_i + s_{2|B|+1} = 1 \quad (29c)$$
$$\sum_{i\in B}\lambda_i - \lambda_{i+|B|} + s_{2|B|+2} = -1 \quad (29d)$$
$$-\lambda_j - \lambda_{j+|B|} + s_{j+2|B|+2} = 2C_j \quad (29e)$$
$$\lambda_j + \lambda_{j+|B|} + s_{j+3|B|+2} = 1_j \quad (29f)$$
$$-\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) - b^+ + b^- - \xi_j + u_j = \epsilon - d_j \quad (29g)$$
$$\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) + b^+ - b^- - \xi_j + u_j = \epsilon + d_j \quad (29h)$$
$$s_j\alpha_j^+ + s_{j+|B|}\alpha_j^- + s_{2|B|+1}b^+ + s_{2|B|+2}b^- \quad (29i)$$
$$+s_{j+2|B|+2}\xi_i + s_{j+3|B|+2}u_j = 0 \quad (29j)$$
$$s_i, \alpha_j^+, \alpha_j^-, b^+, b^-, \xi_j, u_j \geq 0 \quad (29k)$$
$$\text{for all } i, j \in B.$$

The initial working set B is assumed to be smaller than the original problem since we kept only $|B|$ data points and $3|B|$ constraints for the primal. For the dual, we have $2|B|$ out of $2N$ data points and $4|B| + 2$ out of $4N + 2$ constraints. And, also it is assumed that the set of parameters $(T, |B|, C, \epsilon, \sigma, t^*, \mu)$ are given. The first parameter C is the trade-off between overfitting or relaxing the solution. The parameter σ controls the characteristics of the kernel function like the RBF in (10). The parameter t^* is the maximum number of iterations allowed.

Under this formulation, we start by fixing $\alpha_i^\pm = 0, \forall i \in M$, and proceed to solve the subproblem (27),(28) satisfying the KKT conditions (29a)-(29k) of the subproblem. If the subproblem is solved, then check if the KKT conditions are satisfied for all $i \in M$. To do this we have to restate the KKT conditions, such that the missing values in the objective function and constraints are estimated according to the subproblem solution.

First, assume that the primal subproblem inequalities are satisfied, then remove from the set B those indexes that are not associated with support vectors (*i.e.* those indexes $i \in B$ such that $\alpha_i = 0$). Then, the values for the primal variables ξ_i and u_i

for $i \in M$ can be estimated according to the following cases:

Case 1: When the following inequalities holds true for $j \in M$

$$-\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) - b^+ + b^- - \epsilon + d_j \geq 0 \tag{30a}$$

$$\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) + b^+ - b^- - \epsilon - d_j \geq 0, \tag{30b}$$

then the values for the j-th index are computed as follows

$$u_j = 2\left(\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) + b^+ - b^- - d_j\right), \tag{30c}$$

$$\xi_j = 0. \tag{30d}$$

Case 2: When the inequalities (30) holds false, then the values for the j-th index are computed as follows:

$$u_j = 0, \tag{30e}$$

$$\xi_j = -2\left(\sum_{i\in B}(\alpha_i^+ - \alpha_i^-)k(\mathbf{x}_j, \mathbf{x}_i) + b^+ - b^- - d_j\right). \tag{30f}$$

Here the values for the dual variable $\lambda_i = 0$ for $i \in M$, and the values for the dual slack s_i for $i = 1, 2, ..., 4|M| + 2$ can be estimated as follows:

$$s_j = 1_j - \sum_{i\in B}\lambda_{i+|B|}k(\mathbf{x}_j, \mathbf{x}_i) + \sum_{i\in B}\lambda_i k(\mathbf{x}_j, \mathbf{x}_i) \tag{30g}$$

$$s_{j+|B|} = -1_j - \sum_{i\in B}\lambda_i k(\mathbf{x}_j, \mathbf{x}_i) + \sum_{i\in B}\lambda_{i+|B|}k(\mathbf{x}_j, \mathbf{x}_i) \tag{30h}$$

$$s_{2|B|+1} = 1 - \sum_{i\in B}\lambda_{i+|B|} + \lambda_i \tag{30i}$$

$$s_{2|B|+2} = -1 - \sum_{i\in B}\lambda_i + \lambda_{i+|B|} \tag{30j}$$

$$s_{j+2|B|+2} = 2C_j + \lambda_j + \lambda_{j+|B|} \tag{30k}$$

$$s_{j+3|B|+2} = 1_j - \lambda_j - \lambda_{j+|B|} \tag{30l}$$

$$\text{for all } j \in M.$$

Once we have estimated all the values for the set M using (30), we must verify the KKT conditions; particularly the following conditions

$$\begin{aligned} z_i s_i &= 0 \\ (\mathbf{z}, \mathbf{s}) &\geq \mathbf{0} \\ i &= 1, 2, ..., 4|M| + 2 \end{aligned}$$

because they ensure that both the primal and dual problems are being solved simultaneously, and that the solutions are strictly non-negative. If there were no violations, the problem has been solved for both sets B, M, and the method has converged for the set of parameters given. However, if there were violations in M, we look for those inactive constraints in B (*i.e.* those whose $\alpha_i = 0$) and move them into M and replace those indexes with the indexes that violate the complementarity conditions from M into B. We maintain record of which indexes has been moved from M into B. The number of indexes moved from M to B is proportional to the number of indexes moved out from B to M. In the case that all the constraints in B are active (*i.e.* those whose $\alpha_i \neq 0$), then the size of B is incremented by a scaling exponent as follows

$$|B|_{t+1} = |B|_t + 1 + \lceil \log |B|_t \rceil, \text{ such that: } |B|_{t+1} \leq |M|_t \tag{31}$$

allowing the consideration of more indexes. This ends the first iteration, $t = 0$. And it should be repeated until convergence. After $t > 10$ iterations, we check if there exist any sample indexes that have been moving from M into B for at least five times. If this condition is true, the indexes will be held in B without being removed until the algorithm converges. The complete process can be summarized in the following steps:

Algorithm 1 Decomposition Strategy for Large-Scale LP-SVR Training

1: Set B with the first $|B|$ indexes from the data set.
2: For all indexes in B **Solve LP** sub-problem.
3: Verify if the current solution satisfy the KKT conditions for the indexes in M. If so, then **Stop**.
4: Those indexes that have been at least k times in and out of the working set B are moved permanently into B.
5: If the number of variables violating the KKT conditions increased, then, *roll-back* the indexes that were moved out from B to M at iteration $t - 1$, and go to Step 2.
6: Move the **worst** $1 + \lceil \log |B| \rceil$ violating indexes from M to B. Go to Step 2.

The fact that our algorithm stops when the KKT conditions are satisfied, guarantees the convergence to an optimal solution. Furthermore, our algorithm avoids a possible infinite loop by limiting indexes from going in and out of the set B for a limited number of times. This guarantees that the algorithm will converge in finite number of iterations. Of course, the solution will be sub-optimal if the algorithm stops when the maximum number of iterations t^* is reached.

In the following subsection, we explain how the optimality is reached, and why the function is decreasing at each iteration.

4.1. Convergence and Optimality Conditions

Problem (27) is solved using Interior Point Methods (IPM). A more extensive reference for IPM can be found in [18]. For this case we use an *infeasible* IPM within the *path-following* framework, which means that the algorithm will follow the path to the solution instead of looking at the vertex of each constraint *e.g.* the simplex method. For promoting a fast rate of convergence a *predictor-corrector* strategy in computing the Newton step was chosen.

First, let us consider the KKT conditions (26a)-(26j) established for our problem (18). Let us recall that the problem (15) is equivalent to (18), and that the KKT conditions (24a)-(24d) are also equivalent to (26a)-(26j). IPM considers the KKT con-

ditions as the following function

$$F(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s}) = \begin{pmatrix} \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c} \\ \mathbf{Az} - \mathbf{b} \\ \mathbf{XS1} \end{pmatrix} = 0, \tag{32a}$$

$$(\mathbf{z}, \mathbf{s}) \geq \mathbf{0} \tag{32b}$$

where $\mathbf{X} = \text{diag}(z_1, z_2, ..., z_n)$, and $\mathbf{S} = \text{diag}(s_1, s_2, ..., s_n)$. The IPM generates a set of solutions $(\mathbf{z}^t, \boldsymbol{\lambda}^t, \mathbf{s}^t)$ at each iteration t. The key idea is to find solutions $(\mathbf{z}^t, \boldsymbol{\lambda}^t, \mathbf{s}^t)$ that satisfy $F(\mathbf{z}^t, \boldsymbol{\lambda}^t, \mathbf{s}^t) = 0$ and more importantly $(\mathbf{z}^t, \mathbf{s}^t)$ being strictly positive, except at the solution where $\mathbf{z}$ or s may be equal to zero.

Then, IMP uses a quasi-Newton's method to approach the solution of problem (32a). The most remarkable difference between Newton's method and IPM, is that the former does not care of keeping $(\mathbf{z}, \mathbf{s}) \geq \mathbf{0}$, while the latter does. IPM surrounds the current point in a linear model in order to obtain the step direction $(\Delta\mathbf{z}, \Delta\boldsymbol{\lambda}, \Delta\mathbf{s})$ as follows:

$$J(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s}) \begin{pmatrix} \Delta\mathbf{z} \\ \Delta\boldsymbol{\lambda} \\ \Delta\mathbf{s} \end{pmatrix} = -F(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s}), \tag{33}$$

where $J(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s})$ is the Jacobian of $F(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s})$. Then the step direction (using a predictor-corrector strategy) becomes

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{z} \\ \Delta\boldsymbol{\lambda} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\mathbf{XS1} - \Delta\mathbf{X}^{\text{aff}}\Delta\mathbf{S}^{\text{aff}}\mathbf{1} + \sigma\mu\mathbf{1} \end{pmatrix}, \tag{34}$$

where $\mathbf{r}_c = \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c}$ and $\mathbf{r}_b = \mathbf{Az} - \mathbf{b}$ are residuals, $\Delta\mathbf{X}^{\text{aff}}, \Delta\mathbf{S}^{\text{aff}}$ are the affine-scaling direction, μ is the duality gap, and σ is an adaptive line-search parameter depending on μ. The new iterate is therefore

$$(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s}) + \alpha(\Delta\mathbf{z}, \Delta\boldsymbol{\lambda}, \Delta\mathbf{s}), \tag{35}$$

where $\alpha \in (0, 1]$ is appropriately chosen in order to maintain $(\mathbf{z}, \mathbf{s})$ strictly positive. As mentioned before, the predictor-corrector strategy promotes a very fast rate of convergence, which is desirable. In fact, theoretical studies demonstrate that IPM is *q-quadratically* convergent to a feasible solution, *i.e.* it is equivalent to the Newton method. Even if $J(\mathbf{z}, \boldsymbol{\lambda}, \mathbf{s})$ is degenerate, the IPM is *q-superlinearly* convergent. In contrast, the simplex method which is typically used in most decomposition strategies in large-scale SVM, is of exponential complexity. Figure 6 shows how the IPM iteratively minimizes the KKT conditions at an arbitrarily three-class non-separable classification problem.

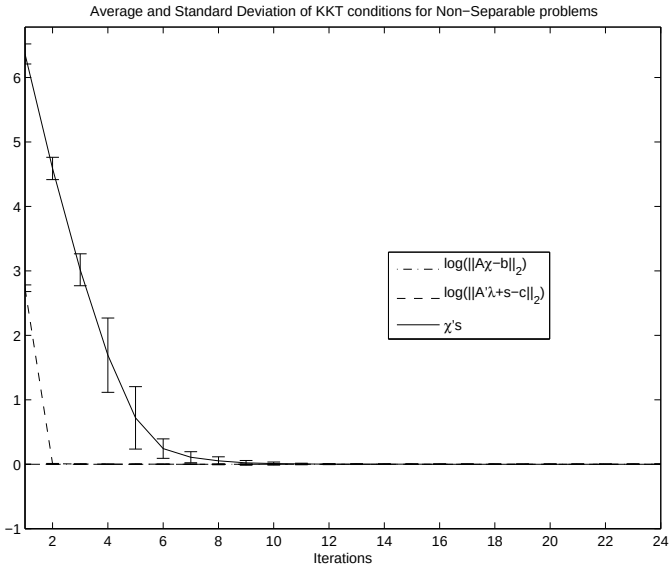

Figure 6: Behavior of the KKT conditions as the number of iterations progress. The primal, dual, and complementarity condition must converge to zero. The results shown represent the average value and the standard deviation over several experiments.

4.1.1. Sonar Large-Scale Target Detection Results

The problem of sonar target classification was introduced initially by Gorman [1]. In his study, a neural network approach was applied to a sonar target classification problem. The problem was to classify readings of sonar returns from an undersea metal cylinder and a rock with a very similar shape and size. He obtained an 82.7% of accuracy, and compared this results to trained human listeners that achieved an accuracy of 88%.

The sonar data used for Gorman experiments were sonar measurements collected from a metal cylinder and a cylindrically shaped rock positioned on a sandy ocean floor. Both targets were approximately 5 ft in length. Readings were taken at a range of 10 meters and obtained from the cylinder and rock at aspect angles spanning 90 and 180 degrees respectively.

The preprocessing of the raw signal was based on experiments with human listeners. A set of sampling apertures superimposed over the spectrogram of the sonar measurement. The dataset was composed of 60 spectral samples, normalized to take on values between 0.0 and 1.0.

There have been more recent approaches to solve this problem. Reid [19] reported an 82% of accuracy with his Random Forest algorithm.

In 2009 [20], Siddiqui *et.al.* attempted to use sparse transformations and dimensionality reduction techniques, however, the maximum accuracy obtained was 83%. Altough they obtain slighly better results than Gorman, they have achieved this with only 25% of the dataset.

Malzahn1 *et.al.* [21] achieved in 88% accuracy in average as they incremented (via bootsrapping) the original dataset.

Now, to improve existing methos, we propose a sequential decomposition model for sonar target detection problem using SVR, the approach shown in Algorithm 1, and the interior point method with predictor-corrector in the form of (34).

The classification problem is defined by a large-scale sonar dataset that consist of two classes non-separable, with a million of samples each. For training we have two million, and for testing we have another two million readings. This dataset defines the set $\{\mathbf{x}_i, d_i\}_{i=1}^{N=4,000,000}$, with $\mathbf{x} \in \Re^2$ and $d \in \{0, 1, 2\}$.

The reader must be aware of the high degree of non-separability of the data. To illustrate this, we have used PCA to make the dataset iid. A few readings projected in PCA and the resulting hyperplane using only two dimensions is shown in

Figure 7. The decision boundaries found using PCA reduction to two dimensions and SVR illustrate the complexity in the separation of the two classes, even using optimal classifiers. When

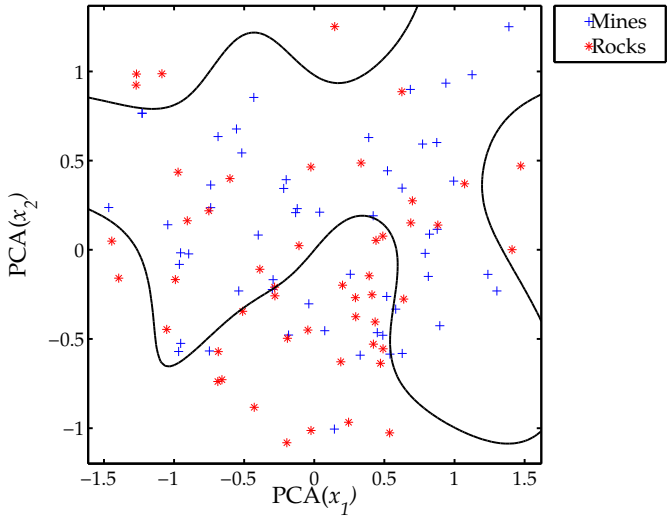

Figure 7: PCA-based decision hyperplane found by the LP-SVR approach. Note the non-separability of the data in two dimensions with PCA

we performed an analysis of the area under Receiver Operating Characteristic (ROC) curve, by varying the dimensionality reduction rate, we can appreciate how intuitive is that by preserving most of the information the detection rate (in terms of the area under the ROC curve) improves and hence, better results are produced. This behavior is illustrated in Figure 8.

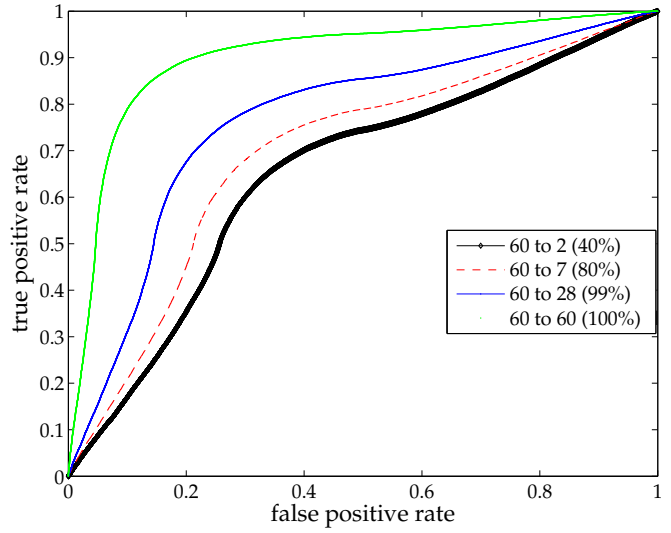

Figure 8: Receiving Operating Characteristic curve for different reduction amounts using PCA. The more information is kept the more robust the detection is. Note that the dimensional increase is not linear with respect to the ROC behavior.

When we used no PCA projections, and kept the proposed large-scale algorithm we observed an accuracy of 93.6% and the area under the ROC curve was 0.9509 (see Figure 9). These results suggest that the iterative optimization that preserves support vectors that promote the solution of the global optimization problem is comparable and in most cases improves previous efforts to solve the problem of sonar mines-rock detection.

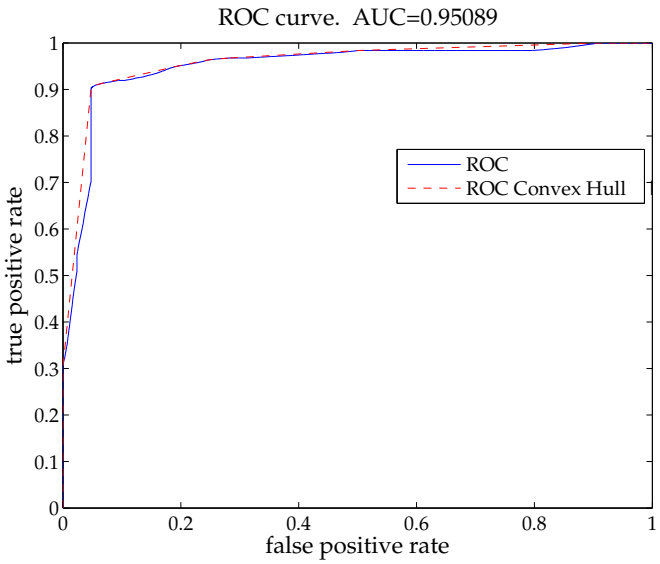

Figure 9: ROC curve using no PCA dimensionality reduction. The ROC is not as smooth as when PCA is used; however, the performance and the area under the ROC curve is greater.

In Table 1 we present the summary of the classification performances for the large-scale sonar mines-rocks detection problem. The variables shown are: dimensionality of the problem, number of samples, accuracy, in percentage; accuracy in training (in parenthesis); and average time, in seconds; its standard deviation; and the total number of iterations (in parenthesis).

Table 1: Classification performance over large-scale sonar mines-rocks detection. The variables are: dimensionality of the problem, number of samples, accuracy (in testing), number of SVs, and the average iteration time, and number of t iterations between parenthesis.

PCA	Dim.	N	Accuracy	SVs	Time (t)
Yes	2	4×10^6	58.1	22	0.54 (17)
Yes	7	4×10^6	62.3	28	0.60 (15)
Yes	28	4×10^6	71.5	39	0.90 (16)
Yes	60	4×10^6	88.8	39	1.09 (18)
No	60	4×10^6	93.6	39	1.08 (18)

From Table 1 we can observe that the number of iterations required to reach a solution is very small for this problem. The expectation of the algorithm from preliminary results are promising from current evaluation of our algorithm. In comparison, a typical simplex method will take several thousands of iterations.

5. Conclusion

In this paper we propose to approach the problem of nonlinear classification of sonar targets (mines versus rocks) when the very large number of samples available make the typical

SVR model computationally intractable since the solution finding process involves highly-dimensional vector operations. The aim of this research is to overcome the natural limitation of a large-scale problems particular to SVR with a real-life application. We propose a sequential linear programming support vector regression (SLP-SVR) approach that uses an interior point method (IPM) to solve the LP optimization problem. The proposed idea consists of a sequential selection of data points and then solve the sub-problem associated to these data points. Then we preserve the *support vectors* at each iterate (chunking). We use the proposed IPM-based LP-SVR approach to find the optimal solution to the classification problem iteratively by by taking advantage of the quadratic rate of convergence of IPM. To improve the rate of convergence, we use a predictor corrector scheme for IPM. Experiments demonstrate the ability to perform classification over non-trivial problems; while at the same time it demonstrates a very fast rate of convergence. The proposed research promises to be as robust and as efficient as the best state of the art SVM training approaches, and also to have many applications in engineering problems.

References

[1] R. Gorman, T. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets, Neural networks 1 (1988) 75–89.

[2] V. N. Vapnik, The nature of statistical learning theory, Springer, New York, NY, 1995.

[3] V. Vapnik, S. Golowich, A. Smola, Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing, Neural Information Processing Systems, MIT Press, Cambridge, MA, 1997.

[4] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop, pp. 276 –285.

[5] J. Platt, Using analytic qp and sparseness to speed training of support vector machines, Advances in Neural Information Processing Systems (1999) 557–563.

[6] T. Joachims, Making large scale svm learning practical (1999).

[7] R. Rifkin, Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[8] P. Drineas, M. W. Mahoney, On the nystrm method for approximating a gram matrix for improved kernel-based learning, Journal of Machine Learning Research 6 (2005) 2153–2175.

[9] D. Hush, P. Kelly, C. Scovel, I. Steinwart, Qp algorithms with guaranteed accuracy and run time for support vector machines, The Journal of Machine Learning Research 7 (2006) 769.

[10] A. J. Smola, B. Scholkopf, A tutorial on support vector regression, Statistics and Computing 14 (2004) 199–222.

[11] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, V. Vapnik, Support vector regression machines (1996) 155–161.

[12] S. S. Haykin, Neural networks and learning machines, Prentice Hall, 2009.

[13] J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 209 (1909) 415–446.

[14] R. Courant, D. Hilbert, Methods of mathematical physics, Interscience New York, 1966.

[15] Z. Lu, J. Sun, K. R. Butts, Linear programming support vector regression with wavelet kernel: A new approach to nonlinear dynamical systems identification, Mathematics and Computers in Simulation 79 (2009) 2051–2063.

[16] Y. Torii, S. Abe, Decomposition techniques for training linear programming support vector machines, Neurocomputing 72 (2009) 973–984.

[17] L. Zhang, W. Zhou, On the sparseness of 1-norm support vector machines, Neural Networks 23 (2010) 373 – 385.

[18] S. Wright, Primal-dual interior-point methods, Society for Industrial Mathematics, 1997.

[19] S. Reid, Decreasing the Randomness of Random Forests (2004).

[20] S. Siddiqui, S. Robila, J. Peng, D. Wang, Sparse Representations for Hyperspectral Data Classification, in: Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, volume 2, IEEE.

[21] D. Malzahn, M. Opper, Approximate analytical bootstrap averages for support vector classifiers, Advances in Neural Information Processing Systems 16 (2003).