Quaternionic and complex-valued Support Vector Regression for Equalization and Function Approximation

Alistair Shilton and Daniel T. H. Lai

Abstract—Support Vector Regressors (SVRs) are a class of
nonlinear regressor inspired by Vapnik’s Support Vector (SV)
method for pattern classification. The standard SVR has been
successfully applied to real number regression problems such as
financial prediction and weather forecasting. However in
some applications the domain of the function to be estimated
may be more naturally and efficiently expressed using complex
numbers (e.g. communications channels) or quaternions (e.g. 3-
dimensional geometrical problems). Since SVRs have previously
been proven to be efficient and accurate regressors, the exten-
tion of this method to complex numbers and quaternions is
of great interest. In the present paper the standard SVR
method is extended to cover regression in complex numbers
and quaternions. Our method differs from existing approaches
in-so-far as the cost function applied in the output space is
rotationally invariant, which is important as in most cases it is
the magnitude of the error in the output which is important,
not the angle. We demonstrate the practical usefulness of this
new formulation by considering the problem of communications
channel equalization.

I. INTRODUCTION

The problem of function estimation can be solved using
supervised learning where the unknown function is assumed
to model some generating system such as a chemical plant.
This is commonly treated as a regression problem, where
given a set of training data consisting of inputs and corre-
sponding system outputs, the unknown generating function
is estimated by regressing on the available data.

Support Vector regressors (SVRs) [1], [2], [3] are a class
of nonlinear regressor inspired by Vapnik’s SV formulation
for classification [4], [5]. Like Vapnik’s support vector classi-
fiers, SVRs achieve nonlinearity by first implicitly mapping
all data into a (usually) higher dimensional feature space.
In this feature space, a linear function is constructed by
minimizing a regularized cost function. The regularization
term in the cost function is present to bias the estimated
function toward functions with smaller gradient in feature
space [6], [7] and hence minimise noise sensitivity.

The standard SVR formulation assumes an unknown gen-
erating function with real outputs. However, in many other
areas such as telecommunications and geometrical problems,
the target outputs are either complex numbers (e.g. com-
unication channels, 2-dimensional geometric problems) or
quaternions (e.g. 3-dimensional geometric problems), where
quaternions are an extension of complex numbers and have
4 axes which represent 1 real and 3 imaginary components.

While it is possible to use either 2 or 4 independent SVRs
to separately treat each component (axis) of the complex
number/quaternion output, this may not be desirable for a
number of reasons. Firstly, the complex/quaternion system
outputs may be coupled and treating them independently
may degrade regressor performance. Secondly, as each of
the independent SVRs are constructed using minimization
of a risk function along a single axis, the overall risk
function will not be rotationally symmetric. This means that
changes in angles between the axes will affect the estimated
function when only the magnitude of the error is considered
to be of primal importance. Some previous work has been
done toward the extension of the SVR framework to cover
complex-valued and quaternion-valued regression [8], [9].
However, while this work has addressed the first point (ie.
the problem of interconnectedness) it has not addressed the
second. In this paper we demonstrate a new way of extending
the SVR framework so that both points are addressed.

We begin by first reviewing the standard \( \epsilon \)-SVR method
in section II. This is important to set the background for
our proposed extension to complex/quaternionic systems
in section III. We start with the primal form and systematically
work through the derivation to obtain the dual formulation.
While the derivation of the dual is more complex than the
standard \( \epsilon \)-SVR dual which means that it can be
solved using ideas from existing SVR training algorithms.
To demonstrate the practical applicability of our regressor,
we apply it to a 4-QAM (Quadrature Amplitude Modulation)
channel equalization problem [10] which requires a highly
non-linear decision boundary.

A. Notation

Throughout this paper the quaternionic division algebra
[11], [12], [13] will be denoted \( \mathbb{H} \), the field of complex
numbers \( \mathbb{C} \), the completely ordered field of reals \( \mathbb{R} \), the
positive reals \( \mathbb{R}^+ \) and the negative reals \( \mathbb{R}^- \). We use \( \mathbb{N} \)
to denote the natural numbers (including zero), \( \mathbb{Z} \) the integers,
and \( \mathbb{Z}_n \) to represent the integers modulo \( n \in \mathbb{N} \) (so \( \mathbb{Z}_n = \{0, 1, \ldots, n - 1\} \)). For generality we let \( \mathbb{X} \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\} \). An
\( n \)-sphere of radius \( \epsilon \) will be denoted \( S^n_\epsilon (\mathbb{X}) \), ie.:

\[
S^n_\epsilon (\mathbb{X}) = \{ z \in \mathbb{X}^{n+1} | \| z \| = \epsilon \}
\]

For any \( x \in \mathbb{H}, x_R \in \mathbb{R} \) is defined to be the real part
of \( x \) and \( x_I, x_J, x_K \in \mathbb{R} \) the imaginary parts of \( x \), so in
componentwise notation the quaternion \( x \) may be written
\( x = x_R + ix_I + jx_J + kx_K \) using the standard \( i, j, k \) notation.
The conjugate and norm of \( x \) will be denoted \( \bar{x} \) and \( |x| \) respectively. As per [11], for \( x \in H \):

\[
\bar{x} = x_R - ix_I - jx_J - kx_K
\]

\[
|x| = \sqrt{\bar{x}^*x} = \sqrt{x_R^2 + x_I^2 + x_J^2 + x_K^2} \in \mathbb{R}\setminus\mathbb{R}^- \]

\[
\text{Re} (x) = \frac{1}{2} (x + \bar{x}) = x_R \in \mathbb{R}.
\]

\[
\text{Pu} (x) = \frac{1}{2} (x - \bar{x}) = ix_I + jx_J + kx_K \in \mathbb{H}\setminus(\mathbb{R}^2 \cup \mathbb{R}^-)
\]

\[
\text{Un} (x) = \frac{\bar{x}}{\bar{x}} \in \mathbb{H}_{(1,0)} \forall x \neq 0
\]

\[
\text{sgn} (x) = \text{sgn} (x_R) + j\text{sgn} (x_I) + j\text{sgn} (x_J) + k\text{sgn} (x_K)
\]

\[
\nabla x = \nabla x_R + j\nabla x_I + j\nabla x_J + k\nabla x_K
\]

and for convenience we define \( \text{sgn} (0) = \text{Un} (0) = 0 \). This notation extends to \( \mathbb{C} \) via the removal of all \( j, k \) terms.

The ranges of most indices \( x, y, m \) etc. are implicit and follow the convention \( i, j, k, l \in \mathbb{Z}_N, m \in \mathbb{Z}_{d_H}, p \in \mathbb{Z}_{d_L} \) and \( q, r, s, t \in \{1, J, K\} \); where \( N \) is the size of the training set, \( d_L \) the dimension of input space and \( d_H \) the dimension of the feature space. Ordering of indices of a dissimilar type (eg. \( i \) and \( q \)) is arbitrary, so for example \( \alpha_{i,q} = \alpha_{q,i} \). Ordering of similar indices, however, is not arbitrary, so for example \( Q_{i,j} \neq Q_{j,i} \) in general. Unless otherwise stated summation ranges are implicit, so for example:

\[
\sum_{i \in \mathbb{Z}_N} f_i = \sum_{i \in \mathbb{Z}_N} f_i
\]

Column vectors will be written in lower case bold (eg. \( a, c, \alpha \)) and matrices in upper case bold (eg. \( A, K \)). Transposition will be indicated by a superscript \( T \) (eg. \( a^T \)), and conjugate transposition by a superscript \( \dagger \) (so \( a^\dagger = a^T \)). We denote the identity matrix \( I \), and a vector whose every element is \( 1 \) by \( 1 \). For two vectors \( a, b \in \mathbb{K}^n \), \( ab^T \) is the outer product, \( ab \) the elementwise product and \( a^T b \) the inner product. We also define the elementwise norm \( \|a\|_1 = \|b\|_1 \), the elementwise sigmoid \( \text{sgn}(a) = \text{sgn}(b) \) and the elementwise unit \( (a = \text{Un}(b)) \). The standard vector 2-norm is denoted \( \| \alpha \| = \sqrt{\alpha^T \alpha} \). 

Vapnik's \( \epsilon \)-insensitive risk function [1] (where \( \epsilon \in \mathbb{R}\setminus\mathbb{R}^- \)) is defined to be:

\[
|\theta|_\epsilon = \begin{cases} 
|\theta| - \epsilon & \text{if } |\theta| \geq \epsilon \\
0 & \text{otherwise}
\end{cases}
\]

II. THE STANDARD \( \epsilon \)-SVR REGRESSION PROBLEM

The standard (real valued) \( \epsilon \)-insensitive support vector regression (\( \epsilon \)-SVR) problem is formulated as follows. Given a training set:\footnote{Technically, there is no reason to restrict the input space to \( \mathbb{R}^{d_L} \) (any Lebesgue measurable set could be used). However doing so would make the construction of the primal less intuitive, and moreover once the dual is attained this extension may be directly introduced using standard extensions to Mercer’s theorem.}

\[
\Theta = \{ (x_0, z_0), (x_1, z_1), \ldots, (x_{N-1}, z_{N-1}) \}
\]

\[
x_i \in \mathbb{R}^{d_L} \text{ is the } i^{th} \text{ input vector.}
\]

\[
z_i \in \mathbb{R} \text{ is the system output given input } x_i.
\]

where \( z_i = \hat{g}(x_i) + \text{noise for some } \hat{g} : \mathbb{R}^{d_L} \rightarrow \mathbb{R}, \text{ and } x_i \text{ is drawn from an unknown distribution, construct an }

\text{approximation } g : \mathbb{R}^{d_L} \rightarrow \mathbb{R} \text{ of } \hat{g}. \text{ An approximation } g \text{ constructed for a given training set } \Theta \text{ is called a trained machine, and the construction process training. We assume that all noise sources are smooth, independent and identically distributed (i.i.d.) with zero mean. }

In the standard approach [14], [1], [15] we implicitly define a nonlinear map \( \varphi : \mathbb{R}^{d_L} \rightarrow \mathbb{R}^{d_H} \) from input space to feature space, where often \( d_H \gg d_L \). Using this map, the trained machine is defined to be:

\[
g(x) = \varphi(x)^T w + b
\]

where \( w \in \mathbb{R}^{d_H} \) is the weight vector and \( b \in \mathbb{R} \) is the bias. This is known as the universal approximator form, and is a linear function of position in feature space which is nonlinear in input space by virtue of \( \varphi \) if \( \varphi \) is nonlinear.

The variables \( w \) and \( b \) are chosen by solving the \( \epsilon \)-SVR primal, namely:

\[
\min_{w,b} R(w, b) = \frac{1}{2} w^T w + \frac{C}{N} \sum_{i} [g(x_i) - z_i]_\epsilon
\]

(2)

where \( R \) is known as the regularized risk function. The second term in (2) is a measure of the empirical risk associated with this model when it is applied to the training set \( \Theta \), and the first term is a regularization term included to minimize sensitivity to noise in \( x \) (as \( \nabla_x g(x) \propto \|w\| \)). The constant \( C \in \mathbb{R}^+ \) is used to trade-off empirical risk minimization (and possible overfitting if \( C \) is too large) and regularization (and possible underfitting if \( C \) is too small), while \( \epsilon \in \mathbb{R}\setminus\mathbb{R}^- \) adds a degree of noise insensitivity. The dual form of this optimization problem [3] is usually solved instead of the primal, namely:

\[
\min_{\alpha \in \mathbb{R}^N} Q = \frac{1}{2} \alpha^T K \alpha + \epsilon |\alpha|^T 1 - \alpha^T z
\]

\[
\text{such that: } -\frac{C}{N} 1 \leq \alpha \leq \frac{C}{N} 1
\]

\[
1^T \alpha = 0
\]

where \( K \in \mathbb{R}^{N \times N}, K_{i,j} = K(x_i, x_j) \) and \( K : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \rightarrow \mathbb{R} \) is the kernel function \( K(x,y) = \varphi(x)^T \varphi(y) \), which may be any function satisfying Mercer’s condition [16].

It may be shown that \( K \) is positive semi-definite, and so the dual is a constrained convex quadratic programming problem, and hence all local minima will be global. Note that the exact form of the feature map \( \varphi \) is never required for a given Mercer kernel for either training or use, as the construction of the dual requires only the kernel \( K \) and the trained machine may be expressed solely in terms of \( \alpha, b \) and the kernel function [3]:

\[
g(y) = \sum_{i \in SV} K(y, x_i) \alpha_i + b
\]

III. THE \( \epsilon \)-SVR REGRESSION PROBLEM

We now consider the extension of the standard \( \epsilon \)-SVR to complex and quaternionic \( \epsilon \)-insensitive support vector
regression ($\epsilon_X$-SVR), where $X \in \{R, C, H\}$ is the division algebra of interest. The training set is defined to be:

$$\Theta = \{(x_0, z_0), (x_1, z_1), \ldots, (x_{N-1}, z_{N-1})\}$$

$x_i \in \mathbb{R}^{d_x}$ is the $i^{th}$ input vector.

$z_i \in \mathbb{X}$ is the system output given input $x_i$.

where, as for standard $\epsilon$-SVR, $z_i = \hat{g}(x_i)$ + noise and we aim to construct an approximation of $g$ by $\hat{g}$. In this case, however, $g, \hat{g} : \mathbb{R}^{d_x} \to \mathbb{X}$, and the trained machine is defined by:

$$g(x) = \varphi(x)^\top w + b$$

where $w \in \mathbb{X}^{d_x}$ is the weight vector, $b \in \mathbb{X}$ is the bias, and $\varphi : \mathbb{R}^{d_x} \to \mathbb{X}$ is the map from input space to (non-real in general if $\mathbb{X} \neq \mathbb{R}$) feature space.

By analogy with the standard $\epsilon$-SVR method, $w$ and $b$ are chosen to minimize the regularized cost function:

$$\min_{w,b} R_{\mathbb{X}}(w,b) = \frac{1}{2} w^\top w + \xi^\top \sum_i |g(x_i) - z_i|_\epsilon$$

(4)

where $C \in \mathbb{R}^+$ and $\epsilon \in \mathbb{R}\setminus\mathbb{R}^-$ are constants. This is directly analogous to the standard $\epsilon$-SVR risk, except that:

- The ranges of $w$ and $b$ differ.
- A complex/quaternionic norm is present inside the $\epsilon$-insensitive risk term so that only the magnitude of the error $z_i - g(x_i)$ is penalized.

The regularization term $\frac{1}{2} w^\top w$ is included to minimize the sensitivity of the regressor to noise in $x$. To understand this choice in the current context, let $\tilde{w} = \|w\|^{-1} w$ be the unit vector in the direction of $w$. It may be seen that:

$$\nabla_{x_i} g(x) = \|w\| (\nabla_{x_i} \varphi(x))^\top w$$

is the projection of $\nabla_{x_i} \varphi(x)$ in the direction of $w$, scaled by the magnitude $\|w\|$ of $w$. It follows that $|\nabla_{x_i} g(x)| \propto \|w\|$, and hence minimizing $\frac{1}{2} w^\top w = \frac{1}{2} \|w\|_2^2$ will minimize the sensitivity of the regressor to noise in the input, $x$.

A. The $\epsilon_X$-SVR Primal

To simplify (4) we use the standard $\epsilon$-SVR approach of introducing a set of slack variables $\xi^+ \in \mathbb{R}^N$ into the problem, allowing (4) to be re-expressed as follows:

$$\min_{w,b,\xi^+} R_{\mathbb{X}} = \frac{1}{2} w^\top w + \frac{\epsilon}{N} \sum_i \xi^+$$

such that: $|g(x_i) - z_i| \leq (\epsilon + \xi^+) \forall i$

(5)

Next, to remove the complex/quaternionic norm from the constraint set we introduce a set of unit slack variables $\alpha^\perp \in (S^0(\mathbb{X}))^N$ to project the error terms $g(x_i) - z_i$ onto the real axis, and an additional set of slack variables $\xi \in \mathbb{R}^N$ to counter the ambiguity in this sign of projection. We also re-express the bias using $b = b^\perp b^\parallel$, where $b^\parallel = \|b\|$ in $\mathbb{R}$ and $b^\perp = \text{Un}(b) \in S^0(\mathbb{X}) \cup \{0\}$, to obtain the non-convex quadratic programming problem:

$$\min_{w,b^\parallel,\xi,\alpha^\perp,\xi^+} R_{\mathbb{X}} = \frac{1}{2} w^\top w + \frac{\epsilon}{N} \sum_i (\xi_i + \xi^+_i)$$

such that:

$$\text{Re} (\alpha^\perp_i (g(x_i) - z_i)) \geq - (\epsilon + \xi^+_i) \forall i$$

$$\text{Re} (\alpha^\perp_i (g(x_i) - z_i)) \leq (\epsilon + \xi^+_i) \forall i$$

$$\text{Pu} (\alpha^\perp_i (g(x_i) - z_i)) = 0 \forall i$$

$$\xi, \xi^+ \geq 0$$

(6)

which will refered to as the $\epsilon_X$-SVR primal. Note that, for any solution, $\xi_i = 0$ and $\xi^+_i = |g(x_i) - z_i|$ if

$$\text{Re} (\alpha^\perp_i (g(x_i) - z_i)) = |g(x_i) - z_i|; \text{ and } \xi^+_i = 0$$

and $\xi^+_i = |g(x_i) - z_i|$, otherwise. Hence all solutions $w, b$ to (6) must also be solutions to (4) and vice-versa.

The advantage of this form is that it allows us to construct the $\epsilon_X$-SVR dual, which is directly analogous to the standard $\epsilon$-SVR dual. This is desirable for two reasons. Firstly, it makes the feature map implicit rather than explicit, thereby removing any practical constraints on $d_H$. Secondly, the constraint set of the dual is simpler than the constraint set of the primal, which makes it easier to solve.

B. The $\epsilon_X$-SVR Dual

As the $\epsilon_X$-SVR primal (6) is non-convex it is advantageous to remove this non-convexity before proceeding further. Our method of achieving this is to re-write (6) as a bilevel optimization problem [17]:

$$\min_{\alpha^\perp,\xi^+} R_{UX} = \frac{1}{2} w^\top w + \frac{\epsilon}{N} \sum_i (\xi_i + \xi^+_i)$$

such that: $\text{Pu} (\alpha^\perp_i (g(x_i) - z_i)) = 0 \forall i$

(7)

$\xi, \xi^+ \geq 0$

where $\Psi(\alpha^\perp,\xi^+)$ is the set of solutions to the optimisation problem:

$$\min_{\alpha^\perp,\xi^+} R_{UX} = \frac{1}{2} w^\top w + \frac{\epsilon}{N} \sum_i (\xi_i + \xi^+_i)$$

such that:

$$\text{Re} (\alpha^\perp_i (g(x_i) - z_i)) \geq - (\epsilon + \xi^+_i) \forall i$$

$$\text{Re} (\alpha^\perp_i (g(x_i) - z_i)) \leq (\epsilon + \xi^+_i) \forall i$$

$$\xi, \xi^+ \geq 0$$

(8)

wherein (7) is referred to as the upper-level $\epsilon_X$-SVR primal problem, $R_{UX}$ is referred to as the upper-level objective, (8) is referred to as the lower-level $\epsilon_X$-SVR primal problem, and $R_{UX}$ is referred to as the lower-level objective.

Defining $\psi_i = \varphi(x_i) \alpha^\perp_i$ and re-writing (8) in component-wise form the lower-level $\epsilon_X$-SVR primal problem may be written:

$$\min_{w,\psi_i,\xi_i} R_{UX} = \frac{1}{2} w^\top w + \sum_q w^\top T \psi_q + \frac{\epsilon}{N} \sum_i (\xi_i + \xi^+_i)$$

such that:

$$\psi_i \geq w^\top T \psi_i + (\alpha^\perp_i \psi^\perp \psi_i) \geq \xi^+ + \xi \geq 0$$

(9)

which we note is a convex quadratic optimisation problem in real variables, and hence has a well-defined dual. To construct this dual, for each of the first set of constraints in the lower-level $\epsilon_X$-SVR primal (9) we associate a Lagrange multiplier $\beta_i \geq 0$, and likewise for each constraint in the second, third and fourth constraint sets in (9) we associate the Lagrange multipliers $\beta_i \geq 0, \gamma_i \geq 0$ and $\gamma^+_i \geq 0$. 
respectively. Using this notation the Lagrangian of (9) is:

\[
\mathcal{L}_X = \frac{1}{2} w^T w_R + \frac{1}{2} \sum_i \psi_i^T \psi_i + \frac{C}{N} \xi^T 1 + \frac{C}{N} \xi^T 1 \\
- \sum_i \beta_i \psi_i^T w_R - \sum_i \beta_i \sum_j \psi_i^T w_q - b_i \beta^T \text{Re} (\bar{\alpha}^T \beta) \\
+ \beta^T \text{Re} (\bar{\alpha}^T \gamma) - \epsilon \beta^T b - \xi^T \beta - \xi^T \gamma \\
- \sum_i \bar{\psi}_i \psi_i^T w_R - \sum_i \bar{\psi}_i \sum_j \psi_i^T w_q - b_i \beta^T \text{Re} (\bar{\alpha}^T \beta) \\
+ \beta^T \text{Re} (\bar{\alpha}^T \gamma) + \epsilon \beta^T 1 + \xi^T \beta^* - \xi^T \gamma^*
\]

and the Wolfe dual [18] of (9) is:

\[
\min_{w, b, \parallel} \max_{\xi, \beta, \parallel, \xi^*, \beta^*, \parallel, \gamma, \parallel, \gamma^*, \parallel} \mathcal{L}_X \\
\text{such that:} \quad \nabla_w \mathcal{L}_X = 0, \nabla_{b, \parallel} \mathcal{L}_X = 0 \\
\nabla_\xi \mathcal{L}_X = -\nabla_\xi^* \mathcal{L}_X = 0 \\
\beta, -\beta^* \geq 0 \\
\gamma, \gamma^* \geq 0
\]

which has the KKT optimality conditions:

\[
\gamma (\nabla_\gamma \mathcal{L}_X) = \gamma^* (\nabla_\gamma^* \mathcal{L}_X) = 0 \quad \text{(12)} \\
\beta (\nabla_\beta \mathcal{L}_X) = \beta^* (\nabla_\beta^* \mathcal{L}_X) = 0 \quad \text{(13)} \\
\nabla_w \mathcal{L}_X = 0 \quad \text{(14)} \\
\nabla_{b, \parallel} \mathcal{L}_X = 0 \quad \text{(15)} \\
\nabla_\xi \mathcal{L}_X = \nabla_\xi^* \mathcal{L}_X = 0 \quad \text{(16)} \\
\beta, -\beta^* \geq 0 \quad \text{(17)} \\
\gamma, \gamma^* \geq 0 \quad \text{(18)}
\]

Condition (16) implies that:

\[
\beta = \frac{C}{N} 1 - \gamma \quad \text{(19)} \\
-\beta^* = \frac{C}{N} 1 - \gamma^* \quad \text{(20)}
\]

and hence, using (17) and (18) and defining \( \alpha^\parallel = \beta + \beta^* \) and \( \alpha = \alpha^\parallel \alpha^\perp \) it may be seen that:

\[-\frac{C}{N} 1 \leq \alpha^\parallel \leq \frac{C}{N} 1
\]

It follows from (12) that \( \gamma_i = 0 \) if \( \xi_i > 0 \) and \( \gamma^*_i = 0 \) if \( \xi^*_i > 0 \) for all \( i \). Combined with (13), (19) and (20), this implies that:

\[
\beta_i > 0 \quad \forall i : \text{Re} (\bar{\alpha}_i^T \left(g(x_i) - z_i\right)) = -\epsilon - \xi_i \quad \text{(21)} \\
\beta_i = 0 \quad \forall i : \text{Re} (\bar{\alpha}_i^T \left(g(x_i) - z_i\right)) > -\epsilon \quad \text{(22)} \\
\beta^*_i = \xi^*_i > 0 \quad \forall i : \text{Re} (\bar{\alpha}^*_i^T \left(g(x_i) - z_i\right)) < \epsilon \quad \text{(23)} \\
\beta^*_i < 0 \quad \forall i : \text{Re} (\bar{\alpha}^*_i^T \left(g(x_i) - z_i\right)) = \epsilon + \xi^*_i \quad \text{(24)}
\]

and hence:

\[
\xi_i + \xi^*_i = -\frac{N}{C} (|\alpha_i| \epsilon + \text{Re} (\bar{\alpha}_i \left(g(x_i) - z_i\right))) \quad \text{(25)}
\]

Finally, equations (14) and (15) imply that:

\[
w = \sum_i \alpha_i^\parallel \psi_i = \sum_i \alpha_i \varphi (x_i) \quad \text{(26)} \\
\alpha^\parallel^T \text{Re} (\bar{\alpha}^T \beta) = \text{Re} (\left(\alpha^\parallel 1\right) b^*) = 0 \quad \text{(27)}
\]

Using these results the lower-level \( \epsilon_X \)-SVR dual (11) may be written:

\[
\max_{b, \parallel} \min_{\alpha^\parallel \in \mathbb{R}^N} Q_{UX} = \frac{1}{2} \left(\alpha^\parallel \right)^T \left[ \begin{array}{cc} H & b \end{array} \right] \left[ \begin{array}{c} \alpha^\parallel \end{array} \right] + \epsilon |\alpha^\parallel| 1 - \alpha^\parallel^T \text{Re} (\bar{\alpha}^T \gamma) \quad \text{(28)}
\]

such that:

\[
\frac{C}{N} 1 \leq \alpha^\parallel \leq \frac{C}{N} 1 \\
g^T \alpha^\parallel = 0
\]

where \( g \in \mathbb{R}^N \), \( g = \text{Re} (\bar{\alpha}^T \beta) \), \( G \in \mathbb{R}^{N \times N} \), \( G_{i,j} = \text{Re} (\bar{\alpha}_i^T \bar{K}_{ij} \alpha_j^T) \), \( Q_{UX} = -R_{UX} \), \( K_{i,j} = K (x_i, x_j) \), and \( K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \) is the kernel function:

\[
K (x, y) = \varphi_m (x)^T \varphi_m (y)
\]

which may be any function satisfying a quaternion extension Mercer’s condition [19].

Having expressed the lower-level \( \epsilon_X \)-SVR training problem in dual form we will now rewrite the upper-level \( \epsilon_X \)-SVR training problem in terms of \( \alpha \) and \( b \), which allows us to merge the resulting bilevel optimization problem back into a standard optimization problem in terms of \( \alpha \) and \( b \).

Consider the upper-level \( \epsilon_X \)-SVR primal (7). Re-expressing in terms of \( \alpha \) and \( b \), negating and using (25), (26) and (27) this becomes:

\[
\max \limits_{b, \parallel, \alpha^\parallel} Q_{UX} = \frac{1}{2} \left[ \begin{array}{c} \alpha^\parallel \end{array} \right]^T \left[ \begin{array}{cc} H & b \end{array} \right] \left[ \begin{array}{c} \alpha^\parallel \end{array} \right] + \epsilon |\alpha^\parallel| 1 - \text{Re} (\alpha^\parallel^T (\alpha^\parallel^T \gamma)) \quad \text{(29)}
\]

such that:

\[
\text{Pu} (\alpha^\parallel (K \alpha + 1b - z)) = 0 \quad \text{Pu} (\alpha^\parallel, b^\parallel) \text{ solve (28)}
\]

where \( h \in \mathbb{R}^N \), \( h = \alpha^\parallel b^\parallel \), \( H \in \mathbb{R}^{N \times N} \), \( H_{i,j} = \alpha^\parallel_i K_{i,j} \alpha^\parallel_j \) and \( Q_{UX} = -R_{UX} \). Recombining (29) and the lower-level \( \epsilon_X \)-SVR dual (28) we arrive at the complete, single-level \( \epsilon_X \)-SVR dual optimization problem:

\[
\max \min \limits_{b, \parallel, \alpha^\parallel} Q_X = \frac{1}{2} \left[ \begin{array}{c} \alpha \end{array} \right]^T \left[ \begin{array}{cc} K & 1 \end{array} \right] \left[ \begin{array}{c} \alpha \end{array} \right] + \epsilon |\alpha| 1 - \text{Re} (\alpha^T \gamma) \quad \text{(30)}
\]

such that:

\[
0 \leq |\alpha| \leq \frac{C}{N} 1 \\
\text{Pu} (\alpha^\parallel (K \alpha + 1b - z)) = 0 \\
\text{Re} (b^T (1T \alpha)) = 0
\]

Suppose we neglect the third constraints in (30), leaving \( b \) unconstrained. Under this assumption the first order optimality conditions for \( b \) is \( \nabla_b Q_X = 0 \) or, explicitly, \( 1^T \alpha = 0 \). But this automatically satisfies the third constraint in (30), indicating that this constraint is superfluous. Hence the \( \epsilon_X \)-SVR dual may be written in a form directly analogous to the

\[2\]The actual statement of the quaternion extension of Mercer’s condition is essentially identical to the standard statement of Mercer’s condition, although more care must be taken with the ordering of elements in the various equations.

\[3\]When deriving this form, note that constraint (27) allows us to add and subtract arbitrary multiples of \( \text{Re} (\left(\alpha^\parallel 1\right) b^*) \) to \( R_{UX} \) without modifying the resulting expression.
where for all training pairs $(x, y) \in \mathbb{X}$ we have chosen to use the ε-SVR dual form if $\mathbb{X} = \mathbb{R}$ (the final constraint is trivially true in this case, as there are no imaginary elements in $\mathbb{R}$).

Note that the form of the $\varepsilon$-SVR dual is directly analogous to the standard ε-SVR problem. It may also be seen, using (3) and (26), that the trained $\varepsilon$-SVR machine has the form:

$$
g(y) = \sum_{i \in SV} K(y, x_i) \alpha_i + b$$

and hence, as for standard ε-SVR, training and use of the $\varepsilon$-SVR does not require an explicit feature map, only a kernel function satisfying Mercer’s condition.

### C. Optimality Conditions

For completeness we now give the optimality conditions of the $\varepsilon$-SVR dual. To begin we define:

$$
\begin{bmatrix}
  e \\
  f
\end{bmatrix} =
\begin{bmatrix}
  K & 1^T \\
  1 & 0
\end{bmatrix}
\begin{bmatrix}
  \alpha \\
  b
\end{bmatrix} -
\begin{bmatrix}
  z
\end{bmatrix}
$$

where for all training pairs $(x_i, z_i) \in \Theta$, $e_i \in \mathbb{X}$ is the difference between the measured output of the system $z_i$ given an input $x_i$ and the output of the trained machine $g(x_i)$ given the same input. Using this notation, the lower-level optimality conditions (21)-(24) may be written:

$$
\begin{align*}
|e_i| &\geq \varepsilon &\forall i: |\alpha_i| &< \frac{C}{\sqrt{N}} \\
|e_i| &\leq \varepsilon &\forall i: |\alpha_i| &> \frac{C}{\sqrt{N}} \\
|e_i| &\leq \varepsilon &\forall i: \alpha_i = 0
\end{align*}
$$

Combined with the additional constraints of the $\varepsilon$-SVR dual (30) the optimality conditions may be seen to be:

$$
\begin{align*}
f &= 0 \\
0 &\leq |\alpha| \leq \frac{C}{\sqrt{N}} \\
|e_i| &= \pm (\varepsilon + \chi_i) &\forall i: |\alpha_i| &= \frac{C}{\sqrt{N}} \\
|e_i| &= \pm \varepsilon &\forall i: 0 < |\alpha_i| < \frac{C}{\sqrt{N}} \\
|e_i| &\leq \varepsilon &\forall i: \alpha_i = 0
\end{align*}
$$

where $\chi \in (\mathbb{R}^+ \cup \{0\})^N$.

### IV. EXPERIMENTAL RESULTS

In this section we consider a practical application for the $\varepsilon$-SVR. Specifically, we consider the problem of equalization of a 4-symbol quadrature amplitude modulated (4-QAM) signal over a complex linear communication channel.

All code for this experiment was written in C++, and all simulations were done on a 3.2GHz Pentium D 940 processor based machine with 4GB of RAM running Ubuntu Linux 6.06 (Dapper Drake). The $\varepsilon$-SVR optimizer code was based on a modified version of SVMHeavy [20].
results show that the decision boundary compares favourably with the optimal decision boundary derived using an ideal Bayesian equalizer for this channel over a range of SNRs.

REFERENCES


Author/s:
SHILTON, A; LAI, T

Title:
Quaternionic and complex-valued Support Vector Regression for Equalization and Function Approximation

Date:
2007

Citation:

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/34668