

A Support Vector Machine for Regression in Complex Field

Rongling LANG¹, Fei ZHAO¹, Yongtang SHI^{2*}

¹*School of Electronics and Information Engineering, Beihang University, Beijing 100191, China*

²*Center for Combinatorics and LPMC, Nankai University, Tianjin 300071, China*

e-mail: ronglinglang@163.com, zhaofei9307@163.com, shi@nankai.edu.cn

Received: September 2016; accepted: November 2017

Abstract. In this paper, one method for training the Support Vector Regression (SVR) machine in the complex data field is presented, which takes into account all the information of both the real and imaginary parts simultaneously. Comparing to the existing methods, it not only considers the geometric information of the complex-valued data, but also can be trained with the same amount of computation as the original SVR in the real data field. The accuracy of the proposed method is analysed by the simulation experiments. This also can be applied to the field of anti-interference for satellite navigation successfully, which shows its effectiveness in practical application.

Key words: support vector machine for regression, complex field, kernel function.

1. Introduction

Support Vector Machine (SVM) is a high-performance machine learning algorithm, which was introduced by Vapnik in the early 1990s (Ramón and Christodoulou, 2006). SVM has been proved to have excellent performances in addressing non-linear regression problems due to the robustness against noise and interferences. In the application of SVM, there is a kind of important model that is commonly known as SVR (Ramón and Christodoulou, 2006).

The original SVR is targeted to treat real numbers. The SVRs in complex data field are also desired in some field, such as the beam forming (Ramón *et al.*, 2005; Gaudes *et al.*, 2007) and pattern recognition (Scholkopf and Smola, 2002; Chen and Xie, 2005). The main method for processing the complex-valued SVR model is using two separate processes for the real and imaginary parts (Bouboulis and Theodoridis, 2011) respectively, which misses the cross-information of the real and imaginary parts of a complex number.

Currently, Kernel-based methods have been employed to solve nonlinear tasks for SVRs. The key method is Reproducing Kernel Hilbert Spaces (RKHS). Wintinger's calculus on complex RKHS was developed in dissertations (Bouboulis *et al.*, 2013, 2015). It was also proved that a complex SVM/SVR task is equivalent to solving two real SVM/SVR

* Corresponding author.

tasks by exploiting a specific real kernel. In Bouboulis *et al.* (2012), the authors introduced how to use RHKS to solve the problem in complex data field. Different from the above-mentioned methods, we propose a model, which not only considers the geometric information of the complex-valued data, but also can be trained with the same amount of computation as the original SVR in the real data field. The proposed method also has no more additional requests for kernel function than the original SVR in real data field.

The paper is organized as follows. In Section 2, the main contributions of the paper are presented, where the model of complex SVR and the training algorithms for the model are proposed in details. In Section 3, experiments are carried out to demonstrate the performance of the proposed method, and the method is applied to anti-jamming field. Finally, the paper is concluded in Section 4.

2. Complex-Valued SVR Model

In this section, a complex SVR model and the corresponding solving algorithms are proposed. Firstly, the complex-valued original problem of SVR is introduced. Secondly, the dual problem is derived. Thirdly, Karush–Kuhn–Tucker (KKT) conditions are obtained. Finally, the solution of the original problem is achieved by solving the dual problem.

The following notations are used in this paper. The superscripts H and T denote hermitian transposition and transposition. $\|\cdot\|$ is the Euclidean vector norm. $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ denote the real and imaginary part of a complex number, respectively. $|\cdot|$ denotes modulus of a complex data. In particular, when the imaginary part of the complex data is equal to zero, $|\cdot|$ denotes absolute value.

2.1. Model Building

The real-valued SVR can be obtained by solving the following optimization problem (Ramón and Christodoulou, 2006):

$$\min_w \frac{1}{2} \|w\|^2 + P \sum_{i=1}^l (\xi_i + \xi_i'). \quad (1)$$

Subject to the constraints

$$\begin{aligned} y_i - (w^T x_i + b) &\leq \varepsilon + \xi_i, \\ (w^T x_i + b) - y_i &\leq \varepsilon + \xi_i', \\ \xi_i, \xi_i' &\geq 0 \end{aligned} \quad (2)$$

for $i = 1, 2, \dots, l$, where $\{(x_1, y_1), \dots, (x_l, y_l)\}$ is the training set and $x_i \in R^n$, $w \in R^n$, $y_i \in R$. P is a penalty factor. ε is the insensitive loss factor. ξ_i and ξ_i' are the slack variables of the output. P , ε , ξ_i and ξ_i' are all real numbers. In the formula (1), the optimal value can be obtained by finding the optimal value of its dual problem. According to the principle of

real-valued SVR, an SVR model in complex field is built and the corresponding solution method is proposed. In order to simplify the expression, ‘Subject to the constraints’ will be abbreviated to ‘s.t.’. The corresponding complex SVR task is formulated and expressed as

$$\begin{aligned}
 \min \quad & \frac{1}{2} \|w\|^2 + C_1 \sum_{i=1}^l (\xi_i + \xi_i') + C_2 \sum_{i=1}^l (\eta_i + \eta_i'), \\
 \text{s.t.} \quad & \text{Re}(y_i - (w^T x_i + b)) \leq \varepsilon + \xi_i, \\
 & \text{Re}((w^T x_i + b) - y_i) \leq \varepsilon + \xi_i', \\
 & \text{Im}(y_i - (w^T x_i + b)) \leq \varepsilon + \eta_i, \quad i = 1, 2, \dots, l, \\
 & \text{Im}((w^T x_i + b) - y_i) \leq \varepsilon + \eta_i', \\
 & \xi_i, \xi_i', \eta_i, \eta_i' \geq 0
 \end{aligned} \tag{3}$$

where $x_i \in C^n$, $w \in C^n$, $y_i \in C$. C_1 and C_2 are the penalty factor of the real and imaginary parts. ε is the insensitive loss factor. ξ_i and ξ_i' are the slack variables in the real part of the output, and the corresponding η_i and η_i' are in the imaginary part. $C_1, C_2, \varepsilon, \xi_i, \xi_i', \eta_i$ and η_i' are all real numbers. The formula (3) is a complex-valued quadratic programming problem, which also can be solved by solving its dual problem.

2.2. Dual Problem

The Lagrange function of the equation (3) can be expressed as

$$\begin{aligned}
 Lp = \quad & \frac{1}{2} \|w\|^2 + C_1 \sum_{i=1}^l (\xi_i + \xi_i') + C_2 \sum_{i=1}^l (\eta_i + \eta_i') \\
 & - \sum_{i=1}^l (\gamma_i \xi_i + \gamma_i' \xi_i') - \sum_{i=1}^l (\kappa_i \eta_i + \kappa_i' \eta_i') \\
 & + \sum_{i=1}^l \alpha_i [\text{Re}(y_i - w^T x_i - b) - \varepsilon - \xi_i] \\
 & + \sum_{i=1}^l \alpha_i^* [\text{Re}(-y_i + w^T x_i + b) - \varepsilon - \xi_i'] \\
 & + \sum_{i=1}^l \beta_i [\text{Im}(y_i - w^T x_i - b) - \varepsilon - \eta_i] \\
 & + \sum_{i=1}^l \beta_i^* [\text{Im}(-y_i + w^T x_i + b) - \varepsilon - \eta_i'],
 \end{aligned} \tag{4}$$

where $\alpha_i, \alpha_i^*, \beta_i, \beta_i^*, \gamma_i, \gamma_i', \kappa_i$ and κ_i' are the Lagrange multipliers. According to the dual definition of Wolfe (Brandwood, 1983; Huang and Zhang, 2007), the partial derivatives are:

$$\begin{aligned}\nabla_w(Lp) &= w - \sum_{i=1}^l \phi_i x, \\ \nabla_b(Lp) &= - \sum_{i=1}^l \alpha_i + \sum_{i=1}^l \alpha_i^* - j \sum_{i=1}^l \beta_i + j \sum_{i=1}^l \beta_i^*.\end{aligned}$$

For real variables, the gradients can be obtained in the traditional way:

$$\begin{aligned}\nabla_{\xi_i}(Lp) &= C_1 - \gamma_i - \alpha_i, & \nabla_{\xi_i'}(Lp) &= C_1 - \gamma_i' - \alpha_i^*, \\ \nabla_{\eta_i}(Lp) &= C_2 - \kappa_i - \beta_i, & \nabla_{\eta_i'}(Lp) &= C_2 - \kappa_i' - \beta_i^*,\end{aligned}$$

where $\phi_i = (\alpha_i - \alpha_i^*) + j(\beta_i - \beta_i^*)$, for all $i = 1, 2, \dots, l$. As all partial derivatives are equal to zero for saddle point conditions, the following parameters can be obtained:

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) + j(\beta_i - \beta_i^*)x_i, \quad (5)$$

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = \sum_{i=1}^l (\beta_i - \beta_i^*) = 0, \quad (6)$$

$$C_1 - \gamma_i - \alpha_i = 0, \quad C_1 - \gamma_i' - \alpha_i^* = 0, \quad (7)$$

$$C_2 - \kappa_i - \beta_i = 0, \quad C_2 - \kappa_i' - \beta_i^* = 0, \quad (8)$$

for $i = 1, 2, \dots, l$. Let $\lambda_i = \alpha_i - \alpha_i^*$, $\varphi_i = \beta_i - \beta_i^*$, then

$$\phi_i = \lambda_i + j\varphi_i. \quad (9)$$

Accordingly to the equations (5)–(9), the dual problem of the equation (4) can be solved. The solution can be presented as:

$$\begin{aligned}\min & \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \phi_i^H \phi_j (x_i, x_j) + \varepsilon \sum_{i=1}^l (|\lambda_i| + |\varphi_i|) - \operatorname{Re} \left[\sum_{i=1}^l \phi_i^H y_i \right], \\ \text{s.t.} & \begin{cases} \sum_{i=1}^l \phi_i = 0, \\ \alpha_i, \alpha_i^*, \beta_i, \beta_i^* \geq 0, \end{cases} \quad i = 1, 2, \dots, l. \quad (10)\end{aligned}$$

By introducing a kernel function $K(x_i, x_j): C \times C \rightarrow R$, the dual problem can be expressed as:

$$\begin{aligned} \min & \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \phi_i^H \phi_j K(x_i, x_j) + \varepsilon \sum_{i=1}^l (|\lambda_i| + |\varphi_i|) - \operatorname{Re} \left[\sum_{i=1}^l \phi_i^H y_i \right], \\ \text{s.t.} & \begin{cases} \sum_{i=1}^l \phi_i = 0 \\ \alpha_i, \alpha_i^*, \beta_i, \beta_i^* \geq 0, \end{cases} \quad i = 1, 2, \dots, l. \end{aligned} \tag{11}$$

The objective function can be expressed by λ_i and φ_i , and the formula as follows:

$$\begin{aligned} & \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \phi_i^H \phi_j K(x_i, x_j) + \varepsilon \sum_{i=1}^l (|\lambda_i| + |\varphi_i|) - \operatorname{Re} \left[\sum_{i=1}^l \phi_i^H y_i \right] \\ & = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\lambda_i \lambda_j + \varphi_i \varphi_j) K(x_i, x_j) + \varepsilon \sum_{i=1}^l (|\lambda_i| + |\varphi_i|) - \operatorname{Re} \left[\sum_{i=1}^l \lambda_i y_i \right] \\ & \quad - \operatorname{Im} \left[\sum_{i=1}^l \varphi_i y_i \right]. \end{aligned} \tag{12}$$

Assume that the optimal solution of the dual problem (11) is $\bar{\alpha}^{(*)} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \dots, \bar{\alpha}_l, \bar{\alpha}_l^*)^T$ and $\bar{\beta}^{(*)} = (\bar{\beta}_1, \bar{\beta}_1^*, \dots, \bar{\beta}_l, \bar{\beta}_l^*)^T$. The optimal solution of the primal problem can be obtained:

$$w = \sum_{i=1}^l (\bar{\alpha}_i - \bar{\alpha}_i^*) + j(\bar{\beta}_i - \bar{\beta}_i^*)x_i. \tag{13}$$

If the training point is support vector on the boundary, then the threshold \bar{b} can be computed:

$$\bar{b} = y_j - \sum_{i=1}^l \phi_i K(x_i, x_j) + (\varepsilon + j\varepsilon). \tag{14}$$

Finally, a support vector regression function in complex data field can be obtained:

$$f(x) = \sum_{i=1}^l \phi_i K(x_i, x) + \bar{b}. \tag{15}$$

2.3. KKT Conditions

Assume that the solution of the formula (11) is $\bar{\alpha}^{(*)}$ and $\bar{\beta}^{(*)}$, then the KKT condition of (3) can be given in formula (16).

$$\left\{ \begin{array}{l} \operatorname{Re} \left(y_i - \left(\sum_{j=1}^l \phi_j K(x_j, x_i) + b \right) \right) \leq \varepsilon + \xi_i, \\ \operatorname{Re} \left(\left(\sum_{j=1}^l \phi_j K(x_j, x_i) + b \right) - y_i \right) \leq \varepsilon + \xi_i', \\ \operatorname{Im} \left(y_i - \left(\sum_{j=1}^l \phi_j K(x_j, x_i) + b \right) \right) \leq \varepsilon + \eta_i, \\ \operatorname{Im} \left(\left(\sum_{j=1}^l \phi_j K(x_j, x_i) + b \right) - y_i \right) \leq \varepsilon + \eta_i', \\ \alpha_i \left\{ \varepsilon + \xi_i - \operatorname{Re} \left[y_i - \sum_{j=1}^l \phi_j K(x_j, x_i) - b \right] \right\} = 0, \\ \alpha_i^* \left\{ \varepsilon + \xi_i' - \operatorname{Re} \left[-y_i + \sum_{j=1}^l \phi_j K(x_j, x_i) + b \right] \right\} = 0, \\ \beta_i \left\{ \varepsilon + \eta_i - \operatorname{Im} \left[y_i - \sum_{j=1}^l \phi_j K(x_j, x_i) - b \right] \right\} = 0, \\ \beta_i^* \left\{ \varepsilon + \eta_i' - \operatorname{Im} \left[-y_i + \sum_{j=1}^l \phi_j K(x_j, x_i) + b \right] \right\} = 0, \\ (C_1 - \alpha_i) \xi_i = 0, \quad (C_1 - \alpha_i^*) \xi_i' = 0, \\ (C_2 - \beta_i) \eta_i = 0, \quad (C_2 - \beta_i^*) \eta_i' = 0, \\ -\xi_i, -\xi_i', -\eta_i, -\eta_i' \leq 0, \\ -\alpha_i, -\alpha_i^*, -\beta_i, -\beta_i^* \leq 0. \end{array} \right. \quad (16)$$

for all $i = 1, 2, \dots, l$. Here, the inherent relationship among $\alpha_i, \alpha_i^*, \beta_i$ and β_i^* will be considered firstly.

Corollary 1. Let $\bar{\alpha}^{(*)}$ and $\bar{\beta}^{(*)}$ be the solution of formula (11), and then we have

$$\left\{ \begin{array}{l} \alpha_i \alpha_i^* = 0, \\ \beta_i \beta_i^* = 0. \end{array} \right. \quad (17)$$

Proof. The proof procedure is composed of two parts.

In part 1, suppose $\alpha_i \in (0, C_1)$. Based on KKT conditions in (16), the following equations can be achieved:

$$\begin{aligned} \alpha_i \left(\varepsilon + \xi_i - \operatorname{Re} \left(y_i - \sum_{j=1}^l \phi_j K(x_j, x_i) - b \right) \right) &= 0, \\ (C_1 - \alpha_i) \xi_i &= 0. \end{aligned} \tag{18}$$

By calculation of equation (18), we have

$$\begin{aligned} \operatorname{Re} \left(y_i - \sum_{j=1}^l \phi_j K(x_j, x_i) - b \right) &= \varepsilon, \\ \xi_i &= 0. \end{aligned} \tag{19}$$

Through combination of formulas (16) and (19), the following expression can be obtained:

$$\alpha_i^* (\varepsilon + \xi_i' + \operatorname{Re}(y_i - (w \cdot x_i) - b)) = 0, \quad \xi_i' \geq 0. \tag{20}$$

Obviously, $\alpha_i^* = 0$ can be deferred from equation (20).

In part 2, we suppose $\alpha_i = C_1$. According to KKT conditions in (16), it is known that ξ_i is an arbitrary value. And there are some cases as follows.

Part 2.1. Suppose $\xi_i = 0$ and according to part 1, it is easy to get $\alpha_i^* = 0$.

Part 2.2. Suppose $\xi_i \neq 0$ ($\xi_i > 0$), then

$$\begin{cases} \operatorname{Re} \left(-y_i + \sum_{j=1}^l \phi_j K(x_j, x_i) + b \right) = -\varepsilon - \xi_i, \\ \alpha_i^* \left(\varepsilon + \xi_i' + \operatorname{Re} \left(y_i - \sum_{j=1}^l \phi_j K(x_j, x_i) - b \right) \right) = 0, \\ \xi_i' \geq 0. \end{cases} \tag{21}$$

By solving equation (21), $\alpha_i^* = 0$ can also be obtained.

In summary, $\alpha_i^* \alpha_i = 0$ is proved to be true. With the similar method, $\beta_i^* \beta_i = 0$ can be proved as well. \square

According to the Corollary, we can analyse the distribution of the real and imaginary parts of the sample points on the boundary. At the same time, in order to facilitate the expression of the formula, a simple notation is introduced:

$$E_i = E(x_i) = f(x_i) - y_i. \tag{22}$$

Therefore the complex domain KKT conditions can be expressed as:

$$\begin{cases} |\operatorname{Re}(E_i)| \leq \varepsilon & (|\lambda_i| = 0), \\ |\operatorname{Re}(E_i)| = \varepsilon & (0 < |\lambda_i| < C_1), \\ |\operatorname{Re}(E_i)| \geq \varepsilon & (|\lambda_i| = C_1), \\ |\operatorname{Im}(E_i)| \leq \varepsilon & (|\varphi_i| = 0), \\ |\operatorname{Im}(E_i)| = \varepsilon & (0 < |\varphi_i| < C_2), \\ |\operatorname{Im}(E_i)| \geq \varepsilon & (|\varphi_i| = C_2). \end{cases} \quad (23)$$

The formula (11) can be solved by using SMO (Cristianini and Shawo-Taylor, 2005; Yang and Tian, 2004) method.

2.4. SMO Algorithm

SMO algorithm is a special case of decomposing algorithm. The core of the algorithm is to update two Lagrangian multipliers corresponding to two sample points at a time, while others remain unchanged. In the complex domain, the Lagrangian multiplier variables corresponding to sample points (x_i, y_i) and (x_j, y_j) are (λ_i, ϕ_i) and (λ_j, ϕ_j) . It is important for SMO algorithm to choose the two appropriate training points and solve the corresponding optimal problem with two Lagrangian multiplier variables. In the following, these two problems are solved.

Firstly, suppose that there are already two training points (x_1, y_1) and (x_2, y_2) . In order to simplify the formula, let $K_{ij} = K(x_i, x_j)$. Therefore the objective function (12) can be expressed as

$$\begin{aligned} W(\lambda_1, \lambda_2, \phi_1, \phi_2) &= \frac{1}{2}\lambda_1^2 K_{11} + \frac{1}{2}\lambda_2^2 K_{22} + \lambda_1 \lambda_2 K_{12} + \varepsilon[|\lambda_1| + |\lambda_2|] - [\lambda_1 \operatorname{Re}(y_1) + \lambda_2 \operatorname{Re}(y_2)] \\ &\quad + \frac{1}{2}\phi_1^2 K_{11} + \frac{1}{2}\phi_2^2 K_{22} + \phi_1 \phi_2 K_{12} + \varepsilon[|\phi_1| + |\phi_2|] - [\phi_1 \operatorname{Im}(y_1) + \phi_2 \operatorname{Im}(y_2)] \\ &\quad + \lambda_1 \sum_{i=3}^l \lambda_i K_{1i} + \lambda_2 \sum_{i=3}^l \lambda_i K_{2i} + \phi_1 \sum_{i=3}^l \phi_i K_{1i} + \phi_2 \sum_{i=3}^l \phi_i K_{2i} + W(\text{const}), \end{aligned} \quad (24)$$

where $W(\text{const})$ has no relationship with (x_1, y_1) and (x_2, y_2) . Let the initial feasible solution of this problem is $(\lambda_1^{\text{old}}, \phi_1^{\text{old}})$ and $(\lambda_2^{\text{old}}, \phi_2^{\text{old}})$, and the optimal solution of the problem is the new values (λ_1, ϕ_1) and (λ_2, ϕ_2) of the two variables. In order to satisfy the constraint, the new values must satisfy

$$\begin{cases} \lambda_1^{\text{old}} + \lambda_2^{\text{old}} = s_1^{\text{old}} = \lambda_1 + \lambda_2, \\ \phi_1^{\text{old}} + \phi_2^{\text{old}} = s_2^{\text{old}} = \phi_1 + \phi_2, \end{cases} \quad (25)$$

where s_1^{old} and s_2^{old} are constant. The formula (25) is substituted into the objective function (24) and takes partial derivation of W with respect to λ_2 . By simplifying, we finally get

$$\lambda_2 = \lambda_2^{old} + \frac{1}{\kappa} (\text{Re}(E_1) - \text{Re}(E_2) - \varepsilon [\text{sgn}(\lambda_2) - \text{sgn}(\lambda_1)]). \quad (26)$$

Similarly, we can update ϕ_2 :

$$\phi_2 = \phi_2^{old} + \frac{1}{\kappa} (\text{Im}(E_1) - \text{Im}(E_2) - \varepsilon [\text{sgn}(\phi_2) - \text{sgn}(\phi_1)]), \quad (27)$$

where $\kappa = k_{11} + k_{22} - 2k_{12}$ and $\text{sgn}(\cdot)$ is signum function.

Second, we should solve the problem of selecting the two training points in the iteration. According to the definition of gradient, we choose first training point which not only violates KKT conditions but also can maximize the module of gradient.

$$\frac{\partial W}{\partial \lambda_i} = \sum_{j=1}^l \lambda_j K_{ij} + 0 + \varepsilon \text{sgn}(\lambda_i) - \text{Re}(y_i) = \text{Re}(E_i) - \text{Re}(b) + \varepsilon \text{sgn}(\lambda_i), \quad (28)$$

$$\frac{\partial W}{\partial \phi_i} = \sum_{j=1}^l \phi_j K_{ij} + 0 + \varepsilon \text{sgn}(\phi_i) - \text{Im}(y_i) = \text{Im}(E_i) - \text{Im}(b) + \varepsilon \text{sgn}(\phi_i), \quad (29)$$

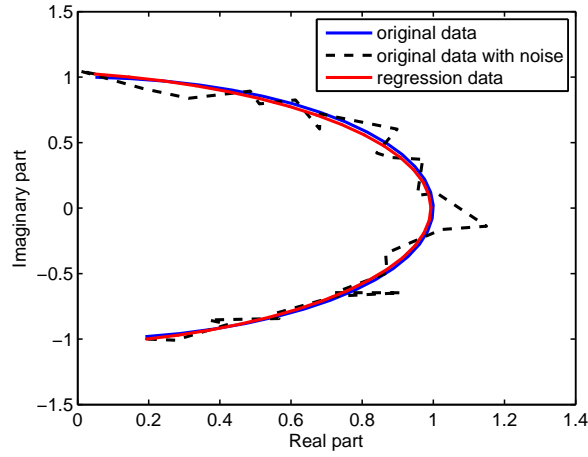
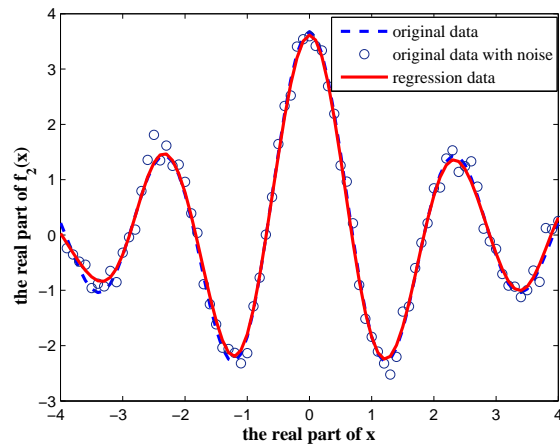
for $i = 1, 2$. Since the ultimate aim of training is to find the training points that make objective function minimize, we choose the second training point which can make greatest change of formula (24).

3. Experimental Results

In this section, the effectiveness of the proposed method in Section 2 will be demonstrated by experiments. Then it will be applied to the anti-interference field for further verification, which is carried out by comparison of our proposed solution method with the existing Power Inversion (PI) algorithm. All the simulation experiments conducted in the paper are implemented in MATLAB.

3.1. SVR-Function Estimation

In this subsection, a regression test is performed to verify the ability of the support vector regression model to handle noise by using two function $f_1(x) = e^{jx/20} + N$, $x \in R$ and $f_2(x) = \frac{\sin(x)}{x} + N$, $x \in C$. For $f_1(x)$, $x \in [0, 60]$. For $f_2(x)$, $x = x_r + jx_i$, x_r and x_i are real random variables, and $x_r \in [-4, 4]$, $x_i \in [-1, 1]$, j as the imaginary unit. N is the white Gaussian noise with zero mean and variances $\sigma = 0.2$. The regression curves are shown in Figs. 1 and 2, which indicate that with the proposed solution method, the regression data can agree well with the original data.

Fig. 1. Regression result of $f_1(x)$.Fig. 2. Regression result of $f_2(x)$.

At the same time, a Monte Carlo experiment for 50 times are conducted to demonstrate the stability of the proposed solution, based on the root mean-square-error (RMSE) of complex-data defined as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N |x_t - \hat{x}_t|^2}, \quad (30)$$

where \hat{x}_t is the estimated value of x_t .

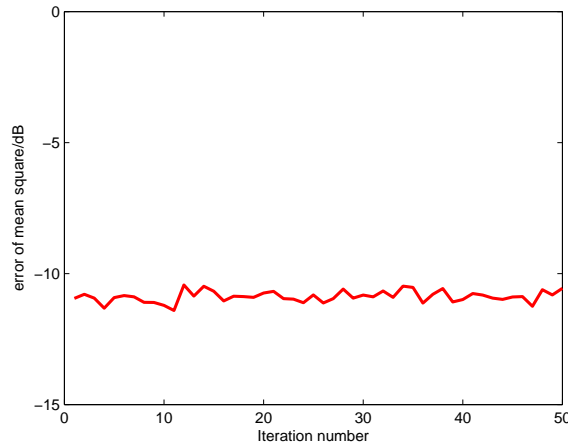


Fig. 3. Resultant RMSE of complex-data.

The resultant RMSE of the complex-data is depicted in Fig. 3, which shows that the square error is small and has a high stability.

In all of the above experiments, a real Gaussian Radial Basis Kernel (RBF) function is selected for SMO algorithm and the Gaussian RBF is expressed as

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{r^2}\right), \tag{31}$$

where r is the Gaussian kernel parameter. For $f_1(x)$, $C_1 = C_2 = 2$, $\varepsilon = 1E-5$, $r = 50$. For $f_2(x)$, $C_1 = C_2 = 10$, $\varepsilon = 0.1$, $r = 0.8$.

Next, we compare the performance of the proposed method with CSV (Bouboulis, 2015). The CSV method exploits a Xcomplex kernel function and transforms the complex problem into two real SVR tasks. In order to compare the performance of the two algorithms under the same standard, the Gaussian Radial Basis kernel function is selected and $C_1 = C_2$. The training samples are generated by $f_2(x)$, and $x_r \in [-1, 1]$, $x_i \in [-1, 1]$. In this experiments, $r = 0.6$ and $\varepsilon = 0.1$ for both proposed solution and CSV. Table 1 shows the difference in RMSE between the two algorithms under 50 Monte Carlo experiments. It can be concluded from the Table 1 that the precision of the proposed solution is roughly the same as that of CSV.

Table 1
RMSE under different penalty factors.

Penalty	Proposed (RMSE)	CSV (RMSE)
$C_1 = C_2 = 2$	-12.89 dB	-12.79 dB
$C_1 = C_2 = 10$	-15.17 dB	-14.93 dB
$C_1 = C_2 = 20$	-15.23 dB	-14.77 dB

3.2. Application to Anti-Interference Problem

In this section, the proposed complex-SVR will be applied to the field of anti-interference to verify its effect in practical application. The anti-interference technology is an important part of array signal processing. According to the literature (Ramón and Christodoulou, 2006), the output vector of an L-element array can be described in matrix notation as:

$$X = A_s S + A_j J + N, \quad (32)$$

where S is desired signals, J is interfere signals, N is noise vector, and A_s is the response vector of the desired signals, A_j is the response vector of the interfere signals. The purpose of the anti-interference is to obtain desired output S and suppress interfere signal. Therefore, the output of the array processor is

$$y = W^H X = d + \varepsilon, \quad (33)$$

where W is the weight vector of the array, d is the estimated value of the desired signal S , ε is the estimation error. In practical applications, we only know the output vector of the array. Therefore, in order to achieve interference suppression, it is important to know how to use the output vector X of an L-element array to get the optimal weight vector W .

Firstly, it is important to generate the training data. For an L-elements Unit Circular Array (UCA), a dataset $X(k) = [x_1(k), x_2(k), \dots, x_l(k)]^T$ can be obtained, where $1 < k < N$ and N is the snapshot number and $x_i(k)$ is the received value of the i -th element at moment k . The autocorrelation matrix of all samples can be computed by

$$R = \frac{1}{N} \sum_{k=1}^N X(k)X(k)^H. \quad (34)$$

According to the PI algorithm (Zahm, 1973; Compton, 1979), the optimal weight is

$$W = \frac{R^{-1} \text{Const}}{\text{Const}^T R^{-1} \text{Const}}, \quad (35)$$

where $\text{Const} = [1, 0, \dots, 0]^T$. And then a set of pairs $\{R, W\}$ can be chosen as training samples.

Secondly, we will show how to use Complex-SVR to achieve anti-interference. The narrowband signal is chosen as the jammer, whose direction is $\theta = 120^\circ$, $\varphi = 50^\circ$, where θ is the azimuth, φ is the elevation angle. In this experiment, a 4-element UCA is chosen and the data X is divided into 100 sections (where $N = 12800$), so as to have 128 snapshots in each section. Then, based on the equations (34) and (35), 100 sample data can be obtained. In the experiment, 70 sample data are chosen as training samples, and the remaining 30 points as the testing samples. In this experiment, $C_1 = C_2 = 20$, $\varepsilon = 1\text{E}-4$, $r = 0.8$. The arrays patterns of PI and the proposed complex-valued SVR are illustrated in Fig. 4. where Fig. 4(a) is a pattern of the azimuth angle when the elevation angle is fixed. and Fig. 4(b)

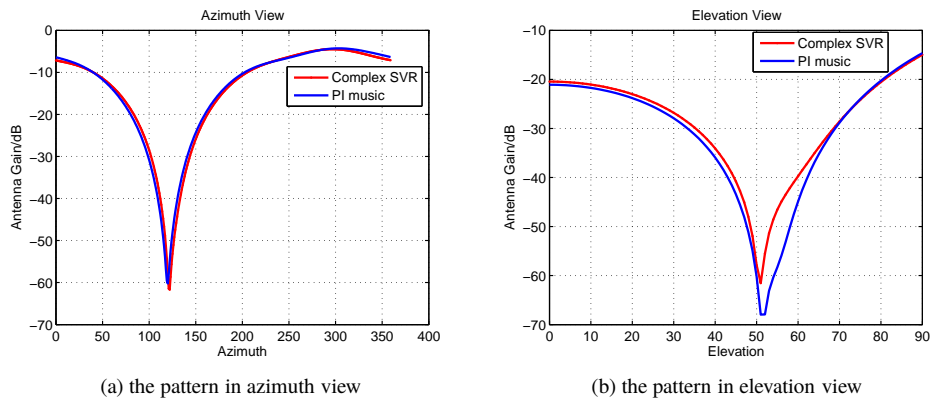


Fig. 4. The arrays pattern of the original PI algorithm and complex SVR.

is a pattern of the elevation angle when the azimuth angle is fixed. The antenna gain is defined as

$$G = 10 \log (\|W^H A_j\|^2). \quad (36)$$

It can be seen from Fig. 4 that complex-valued SVR can get deeper nulls in the direction of the interfering signals than PI.

4. Conclusion

A complex-valued model of SVR was proposed in the paper. The proposed method considers all the information of both the real and imaginary parts of the SVR model. Compared to conventional SVR model, the great feature is that although the method has both the real and imaginary parts information, it does not require extra amount of computation. The accuracy and stability of the proposed model of SVR are finally demonstrated by experiments. In addition, it was also applied to the anti-interference of satellite navigation system for verification of its effect in reality. The results indicate that the proposed method achieved much better performance the normal PI algorithm.

Acknowledgements. The authors would like to thank the editor and referees for their valuable suggestions and comments. Yongtang Shi is partially supported by the Natural Science Foundation of Tianjin (No. 17JCQNJC00300) and The National Natural Science Foundation of China.

References

- Bouboulis, P. (2015). Complex support vector machines for regression and quaternary classification. *IEEE Transaction on Neural Networks and Learning Systems*, 26(6), 1260–1274.
- Bouboulis, P., Theodoridis, S. (2011). Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS. *IEEE Transactions on Signal Processing*, 59(3), 964–978.
- Bouboulis, P., Theodoridis, S., Mavroforakis, C. (2012). Complex support vector regression. In: *IEEE Conference Publications*, pp. 1–6.

- Bouboulis, P., Theodoridou, E., Theodoridis S. (2013). Complex support vector machines for quaternary classification. In: *IEEE International Workshop on Machine Learning for Signal Processing*, Southampton, UK, pp. 22–25.
- Bouboulis, P., Theodoridis, S., Mavroforakis, C., Evaggelatou-Dalla, L. (2015). Complex support vector machines for regression and quaternary classification. *IEEE Transaction on Neural Networks and Learning Systems*, 26(6), 1260–1274.
- Brandwood, D.H. (1983). A Complex gradient operator and its application in adaptive array theory. *IEE Proceedings F – Communications, Radar and Signal Processing*, 130, 11–16.
- Chen, G.Y., Xie, W.F. (2005). Pattern recognition using dual-tree complex wavelets features and SVM. In: *IEEE Conference Publications*, Saskatoon, pp. 2053–2056.
- Compton, R.J. (1979). The power-inversion adaptive array—concept and performance. *IEEE Transaction on Aerospace and Electronic Systems*, 15(6), 803–814.
- Cristianini, N., Shawo-Taylor, J. (2005). *Support Vector Machines and Other Kernel-Based Learning Methods*. China Machine Press.
- Gaudes, C.C., Santamaria, I., Via, J., Gomez, E.M., Paules, T.S. (2007). Robust array beamforming with sidelobe control using support vector machines. *IEEE Transactions on Signal Processing*, 55(2), 574–584.
- Huang, Y., Zhang, S. (2007). Complex matrix decomposition and quadratic programming. *Mathematics of Operations Research*, 32(3), 758–768.
- Ramón, M.M., Christodoulou, C. (2006). *Support Vector Machines for Antenna Array Processing and Electromagnetics*. Morgan & Claypool Publishers.
- Ramón, M.M., Xu N., Christodoulou, C.G. (2005). Beamforming using support vector machines. *IEEE Antennas and Wireless Propagation Letters*, 4(1), 439–442.
- Scholkopf, B., Smola, A. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, USA.
- Yang, D., Tian, Y. (2004). *A New Method of Data Mining-Support Vector Machine*. Science Press, Beijing.
- Zahm, C.L. (1973). Application of adaptive arrays to suppress strong jammers in the presence of weak signals. *IEEE Transaction on Aerospace and Electronic Systems*, 9(3), 260–271.

R. Lang is currently an associate professor in the school of Electronic and Information Engineering in Beihang University. She got BSc Degree and MSc Degree in applied mathematics, Northwestern Polytechnical University, P.R. China, in 2002. She received PhD degree in automatic control, Northwestern Polytechnical University, Xi'an, P.R. China, in 2005. Currently, her research area are data driven GNSS signal monitoring, fault diagnosis and fault prognosis.

F. Zhao is currently pursuing his MS degree in communication and information engineering in Beihang University, Beijing, China. He received his MS degree in communication engineering from China University of Geosciences, Beijing, China. He is now doing the research on jamming suppression in satellite navigation system, blind signal processing and spatial spectrum estimation.

Y. Shi studied mathematics at Northwest University (Xi'an, China) and received his PhD in applied mathematics from Nankai University (Tianjin, China). He visited Technische Universität Bergakademie Freiberg (Germany), UMIT (Austria), Simon Fraser University (Canada) and University of Mississippi (USA). Currently, he is an associate professor at the Center for Combinatorics of Nankai University. His research interests are in graph theory and its applications, especially the applications of graph theory in computer science and information theory. He has written over 60 publications in graph theory and its applications.