Complex Support Vector Regression

Pantelis Bouboulis
Department of Informatics
and Telecommunications
University of Athens
Athens, Greece, 157 84
Email: panbouboulis@gmail.com

Sergios Theodoridis
Department of Informatics
and Telecommunications
University of Athens
Athens, Greece, 157 84
Email: stheodor@di.uoa.gr

Charalampos Mavroforakis
Department of Computer Science
Data Management Lab,
Boston University
Boston, MA 02215, USA
Email: cmav@bu.edu

***Abstract*—We present a support vector regression (SVR) rationale for treating complex data, exploiting the notions of widely linear estimation and pure complex kernels. To compute the Lagrangian and derive the dual problem, we employ the recently presented Wirtinger's calculus on complex RKHS. We prove that this approach is equivalent with solving two real SVR problems exploiting a specific real kernel, which it is induced by the chosen complex kernel.**

I. Introduction

The support vector machines (SVM) framework has become a popular toolbox for addressing classification, regression and time series prediction tasks. Their excellent performance was firmly grounded in the context of statistical learning theory (or VC theory as it is also called, giving credit to Vapnik and Chervonenkis, the authors who developed it), which ensures their supreme generalization properties. Thus, support vector classifiers are amongst the most efficient algorithms for treating real world applications such as optical character recognition, and object recognition problems. In the context of regression, this toolbox is usually called as Support Vector Regression (SVR).

The original support vector machines algorithm is a non-linear generalization of the *Generalized Portrait* algorithm developed in the former USSR in the sixties. The introduction of non-linearity was carried out via a computationally elegant way known today to the machine learning community as the *kernel trick* [1]:

> "Given an algorithm which is formulated in terms of dot products, one can construct an alternative algorithm by replacing each one of the dot products with a positive definite kernel κ."

This has sparkled a new breed of techniques for addressing non linear tasks, the so called *kernel-based methods*. Currently, kernel-based algorithms are a popular tool employed in a variety of scientific domains, ranging from adaptive filtering and image processing [2], [3], [4], [5] to biology and nuclear physics [6], [7].

In kernel-based methods, the notion of the Reproducing Kernel Hilbert Space (RKHS) plays a crucial role. The original data are transformed into a higher dimensional RKHS $\mathcal{H}$ (possibly of infinite dimension) and linear tools are applied to the transformed data in the feature space $\mathcal{H}$. This is equivalent to solving a non-linear problem in the original space. Furthermore, inner products in $\mathcal{H}$ can be computed via the specific kernel function κ associated to the RKHS $\mathcal{H}$, disregarding the actual structure of the space.

Although, the theory of RKHS has been developed by the mathematicians for general complex spaces, most kernel-based methods (as they are targeted to treat real data) employ real kernels. However, complex data arise frequently in applications as diverse as communications, biomedicine, radar, etc. The complex domain not only provides a convenient and elegant representation for such signals, but also a natural way to preserve their characteristics and to handle transformations that need to be performed. In this context, [5] introduced the necessary toolbox for addressing complex tasks in general (of even infinite dimensionality) kernel spaces, while [8], [9] paved the road for applying the (increasingly popular in complex signal processing) widely linear estimation filters.

In this work, exploiting [5], [8], [9], we present the complex support vector regression algorithm to treat complex valued training data. We prove that working in a complex RKHS $\mathbb{H}$, with a pure complex kernel $\kappa_{\mathbb{C}}$, is equivalent to solving two problems in a real RKHS $\mathcal{H}$, with a specific real kernel $\kappa_{\mathbb{R}}$ induced by $\kappa_{\mathbb{C}}$. Our emphasis in this paper is to outline the theoretical development and to verify the validity of our results via a simulation example. The comparative performance of the method in more practical applications will be reported elsewhere, due to lack of space. The paper is organized as follows. In Section II the main mathematical background is outlined and the differences between real RKHS's and complex RKHS's are highlighted. Section III describes the standard real SVR algorithm. The main contributions of the paper can be found in Section IV, where the theory and the respective algorithms are developed. Experiments are presented in Section V. Finally, section VI contains some concluding remarks.

II. Complex RKHS

Throughout the paper, we will denote the set of all integers, real and complex numbers by $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{C}$ respectively. Vector or matrix valued quantities appear in boldfaced symbols. A RKHS [10] is a Hilbert space $\mathcal{H}$ over a field $\mathbb{F}$ for which there exists a positive definite function $\kappa : X \times X \to \mathbb{F}$ with the following two important properties: a) For every $x \in X$,

$\kappa(\cdot, x)$ belongs to $\mathcal{H}$ and b) κ has the so called *reproducing property*, i.e., $f(x) = \langle f, \kappa(\cdot, x)\rangle_{\mathcal{H}}$, for all $f \in \mathcal{H}$, in particular $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x)\rangle_{\mathcal{H}}$. The map $\Phi : X \to \mathcal{H} : \Phi(x) = \kappa(\cdot, x)$ is called the *feature map* of $\mathcal{H}$. Recall, that in the case of complex Hilbert spaces (i.e., $\mathbb{F} = \mathbb{C}$) the inner product is sesqui-linear (i.e., linear in one argument and antilinear in the other) and Hermitian. In the real case, the symmetry condition implies $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x)\rangle_{\mathcal{H}} = \langle \kappa(\cdot, x), \kappa(\cdot, y)\rangle_{\mathcal{H}}$. However, since in the complex case the inner product is Hermitian, the aforementioned condition is equivalent to $\kappa(x, y) = \left(\langle \kappa(\cdot, x), \kappa(\cdot, y)\rangle_{\mathcal{H}}\right)^* = \kappa^*(y, x)$. In the following, we will denote by $\mathbb{H}$ a complex RKHS and by $\mathcal{H}$ a real one.

Definitely, the most popular real kernel in the literature is the *Gaussian radial basis function*, i.e., $\kappa_{t,\mathbb{R}^\nu}(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-t \sum_{k=1}^{\nu}(x_k - y_k)^2\right)$, defined for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^\nu$, where t is a free positive parameter. Many more can be found in [1], [11], [12], [13]. Correspondingly, an important complex kernel is the *complex Gaussian kernel*, which is defined as: $\kappa_{t,\mathbb{C}^\nu}(\boldsymbol{z}, \boldsymbol{w}) := \exp\left(-t \sum_{k=1}^{\nu}(z_k - w_k^*)^2\right)$, where $\boldsymbol{z}, \boldsymbol{w} \in \mathbb{C}^\nu$, z_k denotes the k-th component of the complex vector $\boldsymbol{z} \in \mathbb{C}^\nu$ and $\exp(\cdot)$ is the extended exponential function in the complex domain.

In order to compute the gradients of real valued cost functions, which are defined on complex domains, we adopt the rationale of Wirtinger's calculus [14]. This was brought into light recently [15], [16], [17], [18], [19], as a means to compute, in an efficient and elegant way, gradients of real valued cost functions that are defined on complex domains ($\mathbb{C}^\nu$). It is based on simple rules and principles, which bear a great resemblance to the rules of the standard complex derivative, and it greatly simplifies the calculations of the respective derivatives. The difficulty with real valued cost functions is that they do not obey the Cauchy-Riemann conditions and are not differentiable in the complex domain. The alternative to Wirtinger's calculus would be to consider the complex variables as pairs of two real ones and employ the common real partial derivatives. However, this approach, usually, is more time consuming and leads to more cumbersome expressions. In [5], the notion of Wirtinger's calculus was extended to general complex Hilbert spaces, providing the tool to compute the gradients that are needed to develop kernel-based algorithms for treating complex data.

III. Real valued Support Vector Regression

Suppose we are given training data of the form $\{(\boldsymbol{x}_n, y_n);\ n = 1, \dots, N\} \subset \mathcal{X} \times \mathbb{R}$, where $\mathcal{X} = \mathbb{R}^\nu$ denotes the space of input patterns. Furthermore, let $\mathcal{H}$ be a real RKHS with kernel $\kappa_{\mathbb{R}}$. We transform the input data from $\mathcal{X}$ to $\mathcal{H}$, via the feature map Φ, to obtain the data $\{(\Phi(\boldsymbol{x}_n), y_n);\ n = 1, \dots, N\}$. In support vector regression, the goal is to find an affine function $T : \mathcal{H} \to \mathbb{R} : T(g) = \langle w, g\rangle_{\mathcal{H}} + b$, for some $w \in \mathcal{H}$, $b \in \mathbb{R}$, which is as flat as possible and has at most ϵ deviation from the actually obtained values y_n, for all $n = 1, \dots, N$. Observe that at the training points $\Phi(\boldsymbol{x}_n)$, T takes the values $T(\Phi(\boldsymbol{x}_n))$. Thus, this is equivalent with finding a non-linear function f defined on $\mathcal{X}$ such that

$$f(\boldsymbol{x}) = T \circ \Phi(\boldsymbol{x}) = \langle w, \Phi(\boldsymbol{x})\rangle_{\mathcal{H}} + b, \tag{1}$$

for some $w \in \mathcal{H}$, $b \in \mathbb{R}$, which satisfies the aforementioned properties. The usual formulation of this problem as an optimization task is the following:

$$\begin{aligned} &\underset{w,b}{\text{minimize}} && \tfrac{1}{2}\|w\|^2_{\mathcal{H}} + \tfrac{C}{N}\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n) \\ &\text{subject to} && \begin{cases} \langle w, \Phi(\boldsymbol{x}_n)\rangle_{\mathcal{H}} + b - y_n & \leq \quad \epsilon + \xi_n \\ y_n - \langle w, \Phi(\boldsymbol{x}_n)\rangle_{\mathcal{H}} - b & \leq \quad \epsilon + \hat{\xi}_n \\ \qquad \xi_n, \hat{\xi}_n & \geq \quad 0 \end{cases} \end{aligned} \tag{2}$$

for $n = 1, \dots, N$. The constant C determines a tradeoff between the tolerance of the estimation (i.e., how many larger than ϵ deviations are tolerated) and the flatness of the solution (i.e., T). This corresponds to the so called ϵ-insensitive loss function $|\xi|_\epsilon = \max\{0, |\xi| - \epsilon\}$, [20], [11].

To solve (2), one considers the dual problem derived by the Lagrangian:

$$\begin{aligned} &\underset{\boldsymbol{a},\hat{\boldsymbol{a}}}{\text{maximize}} && \begin{cases} -\frac{1}{2}\displaystyle\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa(x_n, x_m) \\ -\epsilon\displaystyle\sum_{n=1}^{N}(\hat{a}_n + a_n) + \sum_{n=1}^{N} y_n(\hat{a}_n - a_n) \end{cases} \\ &\text{subject to} && \sum_{n=1}^{N}(\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N]. \end{aligned} \tag{3}$$

Note that a_n and $\hat{a}_n$ are the Lagrange multipliers corresponding to to the first two inequalities of problem (6), for $n = 1, 2, \dots, N$. Exploiting the saddle point conditions, it can be proved that $w = \sum_{n=1}^{N}(a_n - \hat{a}_n)\Phi(\boldsymbol{x}_n)$ and thus the solution becomes

$$f(\boldsymbol{x}) = \sum_{n=1}^{N}(\hat{a}_n - a_n)\kappa_{\mathbb{R}}(\boldsymbol{x}_n, \boldsymbol{x}) + b. \tag{4}$$

Furthermore, exploiting the Karush-Khun-Tuker (KKT) conditions one may compute the parameter b.

Several algorithms have been proposed for solving this problem, amongst which are Platt's celebrated Sequential Minimal Optimization (SMO) algorithm [21], interior point methods [22], and geometric algorithms [23], [24]. A more detailed description of the SVR machinery can be found in [25].

IV. Complex Support Vector Regression

Suppose we are given training data of the form $\{(\boldsymbol{z}_n, d_n);\ n = 1, \dots, N\} \subset \mathcal{X} \times \mathbb{C}$, where $\mathcal{X} = \mathbb{C}^\nu$ denotes the space of input patterns. As $\boldsymbol{z}_n$ is complex, we denote by $\boldsymbol{x}_n$ its real part and by $\boldsymbol{y}_n$ its imaginary part respectively, i.e., $\boldsymbol{z}_n = \boldsymbol{x}_n + i\boldsymbol{y}_n$, $n = 1, \dots, N$. Similarly, we denote by d_n^r and d_n^i the real and the imaginary part of d_n, i.e., $d_n = d_n^r + i * d_n^i$, $n = 1, \dots, N$.

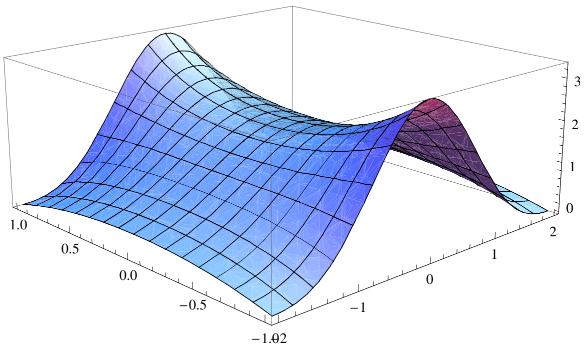

Fig. 1. The element $\Phi((0,0)^T) = 2\kappa_1(\cdot,(0,0)^T)$ of the induced real feature space of the pure complex kernel.

A. Dual Channel SVR

A straightforward approach for addressing this problem (as well as any problem related with complex data) is by considering two different problems in the real domain (this technique is usually referred to as the dual channel approach). That is, split the training data into two sets $\{((\boldsymbol{x}_n,\boldsymbol{y}_n)^T, d_n^r);\ n=1,\dots,N\} \subset \mathbb{R}^{2\nu}\times\mathbb{R}$ and $\{((\boldsymbol{x}_n,\boldsymbol{y}_n)^T, d_n^i);\ n=1,\dots,N\} \subset \mathbb{R}^{2\nu}\times\mathbb{R}$, and perform support vector regression to each set of data using a real kernel $\kappa_{\mathbb{R}}$ and its corresponding RKHS. This is equivalent to the complexification procedure described in [5]. We emphasize that the approach considered here is different from the dual channel rationale. We will develop a framework for solving such a problem on the complex domain employing pure complex kernels, instead of real ones. Nevertheless, we will show that using complex kernels for SVR is equivalent with solving two real problems using a real kernel. This kernel, however, is induced by the selected complex kernel and *it is not one of the standard kernels* appearing in machine learning literature. For example, the use of the complex Gaussian kernel induces a real kernel, which is not the standard real Gaussian rbf (see figure 1). It has to be pointed out that, the dual channel approach and the pure complex approach considered here give different results (see [9], [8]). Depending on the case, the pure complex approach might show increased performance over the dual channel approach and vice versa.

B. Pure Complex SVR

Let $\mathbb{H}$ be a complex RKHS with kernel $\kappa_{\mathbb{C}}$. We transform the input data from $\mathcal{X}$ to $\mathbb{H}$, via the feature map Φ, to obtain the data $\{(\Phi(\boldsymbol{z}_n), y_n);\ n=1,\dots,N\}$. In analogy with the real case and following the principles of widely linear estimation, in complex support vector regression the goal is to find a function $T:\mathbb{H}\to\mathbb{C}: T(g) = \langle g,u\rangle_{\mathbb{H}} + \langle g^*,v\rangle_{\mathbb{H}} + c$, for some $u,v\in\mathbb{H}$, $c\in\mathbb{C}$, which is as flat as possible and has at most ϵ deviation from both the real and imaginary parts of the actually obtained values d_n, for all $n=1,\dots,N$. We emphasize that we employ the widely linear estimation function $S_1:\mathbb{H}\to\mathbb{C}: S_1(g) = \langle g,u\rangle_{\mathbb{H}} + \langle g^*,v\rangle_{\mathbb{H}}$ instead of the usual complex linear function $S_2:\mathbb{H}\to\mathbb{C}: S_1(g) = \langle g,u\rangle_{\mathbb{H}}$ following the ideas of [17], which are becoming quite popular in complex signal processing [26], [27], [28], [29] and have been generalized for the case of complex RKHS in [8], [9].

Observe that at the training points $\Phi(\boldsymbol{z}_n)$, T takes the values $T(\Phi(\boldsymbol{z}_n))$. Following similar arguments as with the real case, this is equivalent with finding a complex non-linear function f defined on $\mathcal{X}$ such that

$$f(\boldsymbol{z}) = T\circ\Phi(\boldsymbol{z}) = \langle\Phi(\boldsymbol{z}),u\rangle_{\mathbb{H}} + \langle\Phi^*(\boldsymbol{z}),v\rangle_{\mathbb{H}} + c, \tag{5}$$

for some $u,v\in\mathbb{H}$, $b\in\mathbb{C}$, which satisfies the aforementioned properties. We formulate the complex support vector regression task as follows:

$$\begin{aligned}
&\min_{u,v,b} \quad \tfrac{1}{2}\|u\|^2_{\mathbb{H}} + \tfrac{1}{2}\|v\|^2_{\mathbb{H}} + \tfrac{C}{N}\sum_{n=1}^{N}(\xi_n^r + \hat{\xi}_n^r + \xi_n^i + \hat{\xi}_n^i)\\
&\text{s. t.} \begin{cases}
\mathrm{Re}(\langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} + \langle\Phi(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} + b - d_n) & \leq \epsilon + \xi_n^r\\
\mathrm{Re}(d_n - \langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} - \langle\Phi^*(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} - b) & \leq \epsilon + \hat{\xi}_n^r\\
\mathrm{Im}(\langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} + \langle\Phi(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} + b - d_n) & \leq \epsilon + \xi_n^i\\
\mathrm{Im}(d_n - \langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} - \langle\Phi^*(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} - b) & \leq \epsilon + \hat{\xi}_n^i\\
\xi_n^r, \hat{\xi}_n^r, \xi_n^i, \hat{\xi}_n^i & \geq 0
\end{cases}
\end{aligned} \tag{6}$$

To solve 6, we derive the Lagrangian and the KKT conditions to obtain the dual problem. Thus we take:

$$\begin{aligned}
\mathcal{L}(u,v,\boldsymbol{a},\hat{\boldsymbol{a}},\boldsymbol{b},\hat{\boldsymbol{b}}) &= \tfrac{1}{2}\|u\|^2 + \tfrac{1}{2}\|v\|^2 + \tfrac{C}{N}\sum_{n=1}^{N}(\xi_n^r + \hat{\xi}_n^r + \xi_n^i + \hat{\xi}_n^i)\\
&+ \sum_{n=1}^{N} a_n(\mathrm{Re}(\langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} + \langle\Phi(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} + c - d_n) - \epsilon - \xi_n^r)\\
&+ \sum_{n=1}^{N} \hat{a}_n(\mathrm{Re}(d_n - \langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} - \langle\Phi^*(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} - c) - \epsilon - \hat{\xi}_n^r)\\
&+ \sum_{n=1}^{N} b_n(\mathrm{Im}(\langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} + \langle\Phi(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} + c - d_n) - \epsilon - \xi_n^i)\\
&+ \sum_{n=1}^{N} \hat{b}_n(\mathrm{Im}(d_n - \langle\Phi(\boldsymbol{z}_n),u\rangle_{\mathbb{H}} - \langle\Phi^*(\boldsymbol{z}_n),v\rangle_{\mathbb{H}} - c) - \epsilon + \hat{\xi}_n^i)\\
&- \sum_{n=1}^{N}\eta_n\xi_n^r - \sum_{n=1}^{N}\hat{\eta}_n\hat{\xi}_n^r - \sum_{n=1}^{N}\theta_n\xi_n^i - \sum_{n=1}^{N}\hat{\theta}_n\hat{\xi}_n^i,
\end{aligned} \tag{7}$$

where a_n, $\hat{a}_n$, b_n, $\hat{b}_n$, η_n, $\hat{\eta}_n$, θ_n, $\hat{\theta}_n$ are the Lagrange multipliers. To exploit the saddle point conditions, we employ the rules of Wirtinger's Calculus for the complex variables on complex RKHS's as described in [5] and deduce that

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial u^*} =& \frac{1}{2}u + \frac{1}{2}\sum_{n=1}^{N}a_n\Phi(\boldsymbol{z}_n) - \frac{1}{2}\sum_{n=1}^{N}\hat{a}_n\Phi(\boldsymbol{z}_n)\\
&- \frac{i}{2}\sum_{n=1}^{N}b_n\Phi(\boldsymbol{z}_n) + \frac{i}{2}\sum_{n=1}^{N}\hat{b}_n\Phi(\boldsymbol{z}_n),
\end{aligned}$$

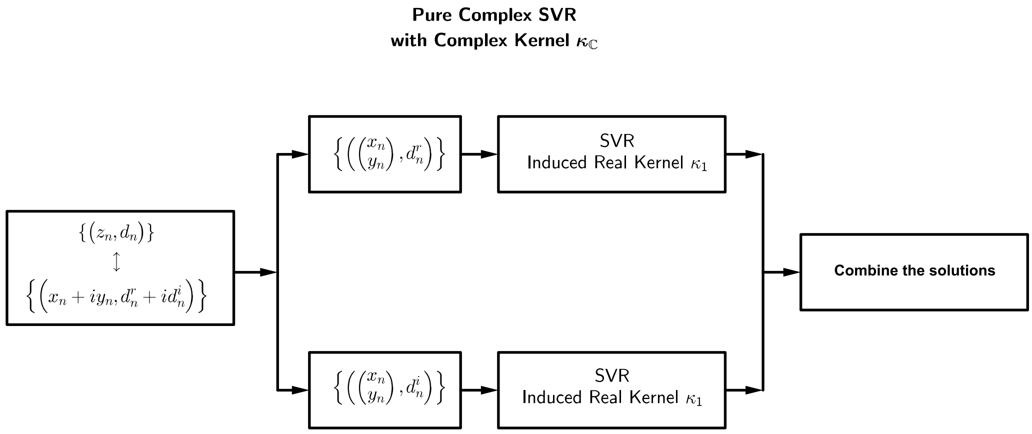

Fig. 2. Pure Complex Support Vector Regression. The difference with the dual channel approach is due to the incorporation of the induced kernel κ_1, which depends on the selection of the complex kernel $\kappa_{\mathbb{C}}$. In this context one exploits the complex structure of the space, which is lost in the dual channel approach.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial v^*} =& \frac{1}{2}v + \frac{1}{2}\sum_{n=1}^{N} a_n \Phi^*(\boldsymbol{z}_n) - \frac{1}{2}\sum_{n=1}^{N} \hat{a}_n \Phi^*(\boldsymbol{z}_n)\\ &- \frac{i}{2}\sum_{n=1}^{N} b_n \Phi^*(\boldsymbol{z}_n) + \frac{i}{2}\sum_{n=1}^{N} \hat{b}_n \Phi^*(\boldsymbol{z}_n),\end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial b^*} = \frac{1}{2}\sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N} \hat{a}_n + \frac{i}{2}\sum_{n=1}^{N} b_n - \frac{i}{2}\sum_{n=1}^{N} \hat{b}_n.$$

For the real variables we compute the gradients in the traditional way:

$$\begin{aligned}\tfrac{\partial \mathcal{L}}{\partial \xi_n^r} &= \tfrac{C}{N} - a_n - \eta_n, & \tfrac{\partial \mathcal{L}}{\partial \hat{\xi}_n^r} &= \tfrac{C}{N} - \hat{a}_n - \hat{\eta}_n,\\ \tfrac{\partial \mathcal{L}}{\partial \xi_n^i} &= \tfrac{C}{N} - b_n - \theta_n, & \tfrac{\partial \mathcal{L}}{\partial \hat{\xi}_n^i} &= \tfrac{C}{N} - \hat{b}_n - \theta_n.\end{aligned}$$

for all $n = 1, \ldots, N$.

As all gradients have to vanish for the saddle point conditions, we finally take that

$$u = \sum_{n=1}^{N} (\hat{a}_n - a_n)\Phi(\boldsymbol{z}_n) - i\sum_{n=1}^{N} (\hat{b}_n - b_n)\Phi(\boldsymbol{z}_n), \tag{8}$$

$$v = \sum_{n=1}^{N} (\hat{a}_n - a_n)\Phi^*(\boldsymbol{z}_n) - i\sum_{n=1}^{N} (\hat{b}_n - b_n)\Phi^*(\boldsymbol{z}_n), \tag{9}$$

$$\sum_{n=1}^{N} (\hat{a}_n - a_n) = \sum_{n=1}^{N} (\hat{b}_n - b_n) = 0, \tag{10}$$

$$\begin{aligned}\eta_n &= \tfrac{C}{N} - a_n, & \hat{\eta}_n &= \tfrac{C}{N} - \hat{a}_n,\\ \theta_n &= \tfrac{C}{N} - b_n, & \hat{\theta}_n &= \tfrac{C}{N} - \hat{b}_n,\end{aligned} \tag{11}$$

for $n = 1, \ldots, N$. In addition, the complex kernel $\kappa_{\mathbb{C}}$ can be written as

$$\kappa_{\mathbb{C}}(\boldsymbol{z}, \boldsymbol{w}) = \kappa_1\left(\begin{pmatrix}\boldsymbol{z}^r\\ \boldsymbol{z}^i\end{pmatrix}, \begin{pmatrix}\boldsymbol{w}^r\\ \boldsymbol{w}^i\end{pmatrix}\right) + i\kappa_2\left(\begin{pmatrix}\boldsymbol{z}^r\\ \boldsymbol{z}^i\end{pmatrix}, \begin{pmatrix}\boldsymbol{w}^r\\ \boldsymbol{w}^i\end{pmatrix}\right), \tag{12}$$

where κ_1, κ_2 are real functions defined on $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$ and $c = c^r + ic^i$. Note that κ_1, κ_2 can be seen either as functions defined on $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$, or as functions defined on $\mathbb{C}^{\nu} \times \mathbb{C}^{\nu}$. Furthermore, the following Lemma holds:

Lemma IV.1. *If $\kappa_{\mathbb{C}}(\boldsymbol{z}, \boldsymbol{w})$ is a complex kernel, then*

$$\kappa_r\left(\begin{pmatrix}\boldsymbol{z}^r\\ \boldsymbol{z}^i\end{pmatrix}, \begin{pmatrix}\boldsymbol{w}^r\\ \boldsymbol{w}^i\end{pmatrix}\right) = \mathrm{Re}(\kappa_{\mathbb{C}}(\boldsymbol{z}, \boldsymbol{w})), \tag{13}$$

where $\boldsymbol{z}^r$, $\boldsymbol{z}^i$, $\boldsymbol{w}^r$, $\boldsymbol{w}^i$ are the real and imaginary parts of $\boldsymbol{z}$ and $\boldsymbol{w}$ respectively, is a real kernel.

Proof: Consider $c_1, \ldots, c_N \in \mathbb{R}$. Then, as $\kappa_{\mathbb{C}}$ is a complex positive definite kernel, for every $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \in \mathbb{C}$ we have that $\sum_{n,m=1}^{N} c_n c_m \kappa_{\mathbb{C}}(\boldsymbol{z}_n, \boldsymbol{z}_m) \geq 0$. Considering that $\kappa_{\mathbb{C}}(\boldsymbol{z}, \boldsymbol{w}) = \kappa_1(\boldsymbol{z}, \boldsymbol{w}) + i\kappa_2(\boldsymbol{z}, \boldsymbol{w})$ and that $\sum_{n,m=1}^{N} c_n c_m \kappa_2(\boldsymbol{z}_n, \boldsymbol{z}_m) = 0$, we finally obtain that $\sum_{n,m=1}^{N} c_n c_m \kappa_1(\boldsymbol{z}_n, \boldsymbol{z}_m) \geq 0$, which means that κ_1 is a real kernel. ∎

Eliminating η_n, $\hat{\eta}_n$, θ_n, $\hat{\theta}_n$ via (11) and u, v via (8-9) we obtain the final form of the Lagrangian:

$$\begin{aligned}\mathcal{L} = \; & -\sum_{n,m=1}^{N} (\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_1(\boldsymbol{z}_n, \boldsymbol{z}_k)\\ & -\sum_{n,m=1}^{N} (\hat{b}_n - b_n)(\hat{b}_m - b_m)\kappa_1(\boldsymbol{z}_n, \boldsymbol{z}_k)\\ & -\epsilon\sum_{n=1}^{N} (a_n + \hat{a}_n + b_n + \hat{b}_n)\\ & +\sum_{n=1}^{N} \boldsymbol{x}_n(\hat{a}_n - a_n) + \sum_{n=1}^{N} \boldsymbol{y}_n(\hat{b}_n - b_n),\end{aligned} \tag{14}$$

where $\boldsymbol{x}_n$, $\boldsymbol{y}_n$ are the real and imaginary parts of $\boldsymbol{z}_n$, $n = 1, \dots, N$. This means that we can split the dual problem into two maximization tasks:

$$\begin{aligned} \underset{\boldsymbol{a},\hat{\boldsymbol{a}}}{\text{maximize}} \quad & \begin{cases} -\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_1(x_n, x_m) \\ -\epsilon\sum_{n=1}^{N}(\hat{a}_n + a_n) + \sum_{n=1}^{N}\boldsymbol{x}_n(\hat{a}_n - a_n) \end{cases} \\ \text{subject to} \quad & \sum_{n=1}^{N}(\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N], \end{aligned} \tag{15}$$

and

$$\begin{aligned} \underset{\boldsymbol{b},\hat{\boldsymbol{b}}}{\text{maximize}} \quad & \begin{cases} -\sum_{n,m=1}^{N}(\hat{b}_n - b_n)(\hat{b}_m - b_m)\kappa_1(x_n, x_m) \\ -\epsilon\sum_{n=1}^{N}(\hat{b}_n + b_n) + \sum_{n=1}^{N}\boldsymbol{y}_n(\hat{b}_n - b_n) \end{cases} \\ \text{subject to} \quad & \sum_{n=1}^{N}(\hat{b}_n - b_n) = 0 \text{ and } b_n, \hat{b}_n \in [0, C/N]. \end{aligned} \tag{16}$$

Observe that the two maximization tasks, (15) and (16), are equivalent with the dual problem of a standard real support vector regression task with kernel $2\kappa_1$. This is a real kernel, as Lemma IV.1 establishes. Therefore (figure 2), one solves the two real SVR tasks for a_n, $\hat{a}_n$, c^r, b_n, $\hat{b}_n$, c^i and combines them to find the solution of the complex problem as

$$\begin{aligned} f(\boldsymbol{z}) =& \langle\Phi(\boldsymbol{z}), u\rangle_{\mathbb{H}} + \langle\Phi^*(\boldsymbol{z}), v\rangle_{\mathbb{H}} + c \\ =& 2\sum_{n=1}^{N}(\hat{a}_n - a_n)\kappa_1(\boldsymbol{z}_n, \boldsymbol{z}) \\ &+ 2i\sum_{n=1}^{N}(\hat{b}_n - b_n)\kappa_1(\boldsymbol{z}_n, \boldsymbol{z}) + c. \end{aligned} \tag{17}$$

In this paper we are focusing mainly in the complex Gaussian kernel. It is important to emphasize that, in this case, the induced kernel $2\kappa_1$ is not the real Gaussian rbf. Figure 1 shows the element $\Phi((0,0)^T) = 2\kappa_1(\cdot, (0,0)^T)$ of the induced real feature space.

V. Experiments

To demonstrate the performance of the proposed algorithmic scheme, we perform a regression test on the complex function $\operatorname{sinc}(z)$. Figures 3 and 4 show the real and the imaginary parts of the estimated $sinc$ function from the performed complex SVR task. For the training data we used an orthogonal grid of 33×9 actual points of the $sinc$ function. Figures 5 and 6 show a similar regression task on the complex $sinc$ function corrupted by a mixture of white Gaussian noise together with some impulses. Note the excellent visual results obtained by the corrupted training data in the second case.

In all the performed experiments, the geometric SVM algorithm was employed using the complex Gaussian kernel (see [23]). The parameters of the SVR task were chosen as $t = 0.25$, $C = 1000$, $\epsilon = 0.1$.

VI. Conclusions

We presented a framework of support vector regression for complex data using pure complex kernels, exploiting the recently developed Wirtinger's calculus for complex RKHS's and the notions of widely linear estimation. We showed that this problem is equivalent to solving two separate real SVR tasks employing an induced real kernel (figure 2). The induced kernel depends on the choice of the complex kernel and it is not one of the usual kernels used in the literature. Although the machinery presented here might seem similar to the dual channel approach, they have important differences. The most important one is due to the incorporation of the induced kernel κ_1, which allows us to exploit the complex structure of the space, which is lost in the dual channel approach. As an example we studied the complex Gaussian kernel and showed by example that the induced kernel is not the real Gaussian rbf. To the best of our knowledge this kernel has not appeared before in the literature. Hence, treating complex tasks directly in the complex plane, opens the way of employing novel kernels.

References

[1] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[2] K. Slavakis, S. Theodoridis, and I. Yamada, "Adaptive constrained learning in reproducing kernel Hilbert spaces: the robust beamforming case," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4744–4764, 2009.

[3] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Adaptive multiregression in reproducing kernel Hilbert spaces: the multiaccess MIMO channel case," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 260 – 276, 2012.

[4] P. Bouboulis, K. Slavakis, and S. Theodoridis, "Adaptive kernel-based image denoising employing semi-parametric regularization," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1465–1479, 2010.

[5] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's calculus to reproducing kernel Hilbert spaces and the complex kernel LMS," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 964–978, 2011.

[6] B. Schlkopf, K. Tsuda, and J.-P. Vert, *Kernel methods in computational biology*. MIT Press, 2004.

[7] A. V. Gribok, J. W. Hines, and R. E. Uhrig, "Use of kernel based techniques for sensor validation in nuclear power plants," in *International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies, Washington, DC, November, 2000.*

[8] P. Bouboulis, S. Theodoridis, and M. Mavroforakis, "Widely linear kernel-based adaptive filters," in *Proc. EUSIPCO*, 2011.

[9] ——, "The augmented complex kernel LMS," *to appear in IEEE Transactions on Signal Processing*, 2012.

[10] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, May 1950.

[11] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4^{th} ed. Academic Press, Nov. 2008.

[12] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[13] P. Bouboulis and M. Mavroforakis, "Reproducing kernel Hilbert spaces and fractal interpolation," *Journal of Computational and Applied Mathematics*, vol. 235, no. 12, pp. 3425–3434, April 2011.

[14] W. Wirtinger, "Zur formalen theorie der functionen von mehr complexen veranderlichen," *Mathematische Annalen*, vol. 97, pp. 357–375, 1927.

[15] D. Mandic and V. Goh, *Complex Valued nonlinear Adaptive Filters*. Wiley, 2009.

[16] T. Adali and H. Li, *Adaptive signal processing: next generation solutions*. Wiley, NJ, 2010, ch. Complex-valued Adaptive Signal Processing, pp. 1–74.

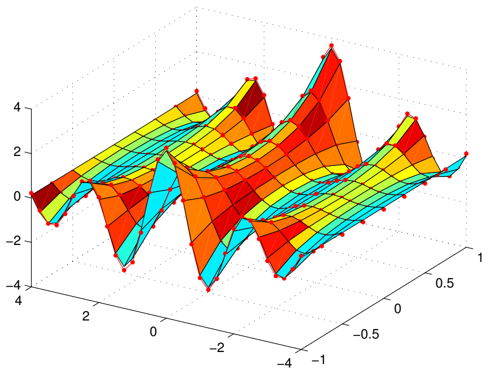

Fig. 3. The real part $(\mathrm{Re}(\mathrm{sinc}(z)))$ of the estimated sinc function from the complex SV regression. The points shown in the figure are the real parts of the training data used in the simulation.

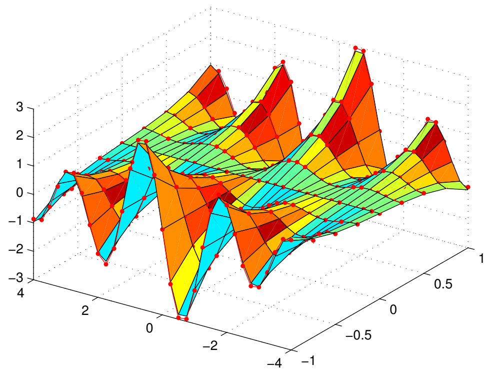

Fig. 4. The imaginary part $(\mathrm{Im}(\mathrm{sinc}(z)))$ of the estimated sinc function from the complex SV regression. The points shown in the figure are the imaginary parts of the training data used in the simulation.

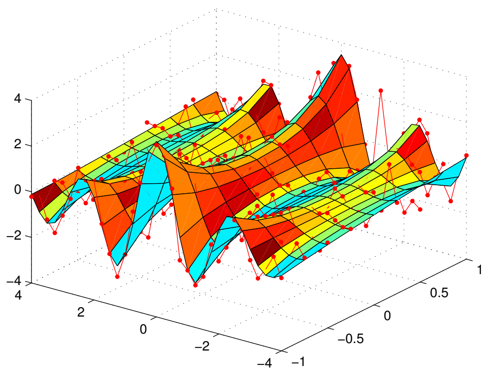

Fig. 5. The real part $(\mathrm{Re}(\mathrm{sinc}(z)))$ of the estimated $sinc$ function from the complex SV regression. The points shown in the figure are the real parts of the noisy training data used in the simulation.

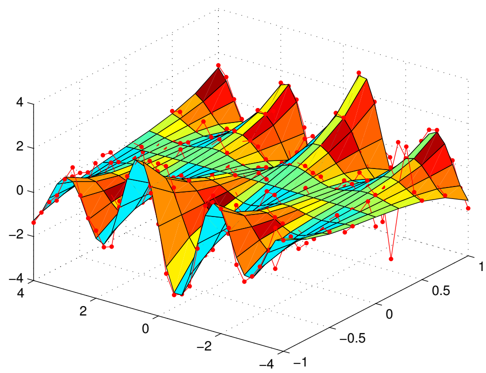

Fig. 6. The imaginary part $(\mathrm{Im}(\mathrm{sinc}(z)))$ of the estimated $sinc$ function from the complex SV regression. The points shown in the figure are the imaginary parts of the noisy training data used in the simulation.

[17] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 2030–2033, 1995.

[18] M. Novey and T. Adali, "On extending the complex ICA algorithm to noncircular sources," *IEEE Transanctions on Signal Processing*, vol. 56, no. 5, pp. 2148–2154, 2008.

[19] H. Li, "Complex-valued adaptive signal processing using Wirtinger calculus and its application to Independent Component Analysis," Ph.D. dissertation, University of Maryland Baltimore County, 2008.

[20] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1999.

[21] J. Platt, "Sequential minimal optimization: A fast algorithm for training Support Vector Machines," [Online] ftp://ftp.research.microsoft.com/pub/tr/tr-98-14.doc, Microsoft Research, Tech. Rep. MSR-TR-98-14, April 1998.

[22] J. Vanderbei, "Loqo: an interior point code for quadratic programming," Statistics and Operations Research, Princeton University, NJ, Tech. Rep., 1994.

[23] M. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 671–682, May 2006.

[24] M. Mavroforakis, M. Sdralis, and S. Theodoridis, "A geometric nearest point algorithm for the efficient solution of the SVM classification task," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1545–1549, 2007.

[25] A. Smola and B. Schölkopf, "A tutorial on support vector regression," Royal Holloway Coll., Univ. London, London, UK Tech, Tech. Rep. NC-TR-98-030, 1998.

[26] C. C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4427–4431, 2010.

[27] P. Chevalier and F. Pipon, "New insights into optimal widely linear receivers for the demodulation of BPSK, MSK and GMSK signals corrupted by noncircular interferences - application to SAIC," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 870–883, 2006.

[28] J. J. Jeon, J. G. Andrews, and K. M. Sung, "The blind widely linear output energy algorithm for DS-CDMA systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1926–1931, 2006.

[29] A. Cacciapuoti, G. Gelli, L. Paura, and F. Verde, "Finite-sample performance analysis of widely linear multiuser receivers for DS-CDMA systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1572 – 1588, 2008.