Complex Support Vector Machines for Regression and Quaternary Classification

Pantelis Bouboulis, *Member, IEEE*, Sergios Theodoridis, *Fellow, IEEE*, Charalampos Mavroforakis, and Leoni Evaggelatou-Dalla

Abstract—The paper presents a new framework for complex support vector regression (SVR) as well as Support Vector Machines (SVM) for quaternary classification. The method exploits the notion of widely linear estimation to model the input-out relation for complex-valued data and considers two cases: 1) the complex data are split into their real and imaginary parts and a typical real kernel is employed to map the complex data to a complexified feature space and 2) a pure complex kernel is used to directly map the data to the induced complex feature space. The recently developed Wirtinger's calculus on complex reproducing kernel Hilbert spaces is employed to compute the Lagrangian and derive the dual optimization problem. As one of our major results, we prove that any complex SVM/SVR task is equivalent with solving two real SVM/SVR tasks exploiting a specific real kernel, which is generated by the chosen complex kernel. In particular, the case of pure complex kernels leads to the generation of new kernels, which have not been considered before. In the classification case, the proposed framework inherently splits the complex space into four parts. This leads naturally to solving the four class-task (quaternary classification), instead of the typical two classes of the real SVM. In turn, this rationale can be used in a multiclass problem as a split-class scenario based on four classes, as opposed to the one-versus-all method; this can lead to significant computational savings. Experiments demonstrate the effectiveness of the proposed framework for regression and classification tasks that involve complex data.

Index Terms—Classification, complex kernels, complex valued data, regression, support vector machines, widely linear estimation.

I. INTRODUCTION

THE support vector machine (SVM) framework has become a popular toolbox for addressing real-world applications that involve nonlinear classification and regression tasks. In its original form, the SVM method is a nonlinear generalization of the Generalized Portrait algorithm, which has been developed in the former USSR in the 1960s. The introduction of nonlinearity was carried out via a computationally elegant way known today as the kernel trick [1]. Usually, this trick is applied in a black-box rationale, where one simply replaces dot products with a positive definite kernel function. The successful application of the kernel trick in SVMs has sparked a new breed of techniques for addressing nonlinear tasks, the so-called kernel-based methods. Currently, kernel-based algorithms constitute a popular tool employed in a variety of scientific domains, ranging from adaptive filtering [2], [3] and image processing to biology and nuclear physics [1], [4]–[18]. The key mathematical notion underlying these methods is that of reproducing kernel Hilbert spaces (RKHS). These are inner product spaces in which the point wise evaluation functional is continuous. Through the kernel trick, 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 so-called feature space $\mathcal{H}$. This is equivalent to solving a nonlinear problem in the original space. Furthermore, inner products in $\mathcal{H}$ can efficiently be computed via the specific kernel function κ associated with the RKHS $\mathcal{H}$, disregarding the actual structure of the space. Recently, this rationale has been generalized, so that the task simultaneously learns the so-called kernel in some fashion, instead of selecting it *a priori* in the context of multiple kernel learning (MKL) [19]–[22].

Although the theory of RKHS has been developed by mathematicians for general complex spaces, most kernel-based methods employ real kernels. This is largely due to the fact that many of them originated as variants of the original SVM formulation, which was targeted to treat real data. However, in modern applications, complex data arise frequently in areas as diverse as communications, biomedicine, radar, and so on. Although all the respective algorithms that employ complex data (e.g., in communications) can also be cast in the real domain (disregarding any type of complex algebra), by splitting the complex data into two parts and working separately, this approach usually leads to more intricate expressions and tedious calculations. The complex domain provides not only a convenient and elegant representation for such data, but also a natural way to preserve their characteristics and to handle transformations that need to be performed.

Hence, the design of SVMs suitable for treating problems of complex and/or multidimensional outputs has attracted

P. Bouboulis is with the Department of Informatics and Telecommunications, University of Athens, Athens 15784, Greece (e-mail: panbouboulis@gmail.com).

S. Theodoridis is with the Department of Informatics and Telecommunications, University of Athens, Athens 15784, Greece, and also with the Research Academic Computer Technology Institute, Patra 26500, Greece (e-mail: stheodor@di.uoa.gr).

C. Mavroforakis is with the Data Management Laboratory, Department of Computer Science, Boston University, Boston, MA 02215 USA (e-mail: cmav@bu.edu).

L. Evaggelatou-Dalla is with the department of Mathematics, University of Athens 15784, Greece (e-mail: ldalla@math.uoa.gr).

some attention in the machine learning community. Perhaps the most complete works, which attempt to generalize the SVM rationale in this fashion, are: 1) Clifford SVM [23] and 2) division algebraic SVR [24]–[26]. In Clifford SVM, the authors use Clifford algebras to extend the SVM framework to multidimensional outputs. Clifford algebras belong to a type of associative algebras, which are used in mathematics to generalize the complex numbers, quaternions, and several other hyper complex number systems. On the other hand, in division algebraic SVR, division algebras are employed for the same purpose. These are algebras, closely related to the Clifford algebras, where all nonzero elements have multiplicative inverses. In a nutshell, Clifford algebras are more general, and they can be employed to create a general algebraic framework (i.e., addition and multiplication operations) in any type of vector spaces (e.g., $\mathbb{R}$, $\mathbb{R}^2$, $\mathbb{R}^3$, ...), while the division algebras are only four: the real numbers, the complex numbers ($\mathbb{R}^2$), the quaternions ($\mathbb{R}^4$), and the octonions ($\mathbb{R}^8$). This is due to the need for inverses, which can only be satisfied in these four vector spaces. Although Clifford algebras are more general, their limitations (e.g., the lack of inverses) make them a difficult tool to work with, compared with the division algebras. Another notable attempt that pursues similar goals is the multiregression SVMs of [27], where the outputs are represented simply as vectors, and an ϵ-insensitive loss function is adopted. Unfortunately, this approach does not result in a well-defined dual problem. In contrast to the more general case of hyper complex outputs, where applications are limited [28], complex-valued SVMs have been adopted by a number of authors for the beam forming problem (see [29], [30]), although restricted to the simple linear case.

It is important to emphasize that most of the aforementioned efforts to apply the SVM rationale to complex and hyper complex numbers are limited to the case of the output data.[1] These methods consider a multidimensional output, which can be represented, for example, as a complex number or a quaternion, while the input data are real vectors. In some cases, complex input data are considered as well, but in a rather trivial way, i.e., splitting the data into their real and imaginary parts. Moreover, these methods employ real-valued kernels to model the input-out relationship, breaking it down to its multidimensional components. However, in this way, many of the rich geometric characteristics of complex and hyper complex spaces are lost.

In this paper, we adopt an alternative rationale. To be in line with the current trend in complex signal processing, we employ the so-called widely linear estimation process, which has been shown to perform better than the conventional linear estimation process [31]–[35]. This means that we model the input-out relationship as a sum of two parts. The first is linear with respect to the input vector, while the second is linear with respect to its conjugate. Furthermore, we consider two cases to generalize the SVM framework to complex spaces. In the first one, the data are split into their real and imaginary parts, and typical well-established real kernels are employed to

map the data into a complexified RKHS. This scenario bears certain similarities with other preexisting techniques that also split the output into two parts (see [25]). The difference with our technique is that the widely linear estimation process is employed to model the input-out relationship of the SVM. In the second case, the modeling takes place directly into complex RKHS, which are generated by pure complex kernels,[2] instead of real ones. In that fashion, the geometry of the complex space is preserved. Moreover, we show that in the case of complex SVMs, the widely linear approach is a necessity, as the alternative path would lead to a significantly restricted model. To compute the gradients, which are required by the Karush–Kuhn–Tucker (KKT) conditions and the dual, we employ the generalized Wirtinger Calculus introduced in [16]. As one of our major results, 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 $\mathcal{H}$, albeit with a specific real kernel $\kappa_{\mathbb{R}}$, which is induced by the complex $\kappa_{\mathbb{C}}$. It must be pointed out that these induced kernels are not trivial. For example, the exploitation of the complex Gaussian kernel results in an induced kernel different from the standard real Gaussian Radial Basis Function.

To summarize, the main contribution of our work is the development of a complete mathematical framework suitable for treating any SVR/SVM task that involves complex data, in an elegant and uniform manner. Moreover, we provide a new way of treating a special multiclassification problem (i.e., quaternary classification). Our emphasis in this paper is to outline the theoretical development and to verify the validity of our results via some simulation examples. The paper is organized as follows. In Section II, the main mathematical background regarding RKHS is outlined, and the differences between a real RKHS and a complex RKHS are highlighted. The main contributions of the paper can be found in Sections III and IV, where the theory and the generalized complex SVR and SVM algorithms are developed, respectively. The complex SVR developed there is suitable for general complex-valued function estimation problems defined on complex domains. The proposed complex SVM rationale, on the other hand, is suitable for quaternary (i.e., four class) classification, in contrast to the binary classification carried out by the real SVM approach. The experiments that are presented in Section V demonstrate certain cases where the use of the pure complex Gaussian kernel in the SVR rationale offers significant advantages over the real Gaussian kernel. In the SVM, besides the new case of quaternary classification, experiments also show how the exploitation of complex data improves the classification accuracy. Finally, Section VI contains some concluding remarks.

II. REAL AND COMPLEX RKHS

We devote this section to present the notation that is adopted in the paper and to summarize the basic mathematical background regarding RKHS. Throughout the paper, we will denote the set of all integers, real and complex numbers,

---

[1] In [23], the authors also consider a Gabor kernel function which takes multivector inputs.

[2] The term pure complex kernels refers to complex-valued kernels with complex variables that are complex analytic.

by $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{C}$, respectively. The imaginary unit is denoted as i, while z^* denotes the conjugate of z. Vector or matrix valued quantities appear in boldfaced symbols.

An RKHS [36] is a Hilbert space $\mathcal{H}$ over a field $\mathbb{F}$ for which there exists a positive definite kernel function κ : $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{F}$ with the following two important properties: 1) for every $x \in \mathcal{X}$, $\kappa(\cdot, x)$ belongs to $\mathcal{H}$ and 2) 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 $\Phi : \mathcal{X} \rightarrow \mathcal{H} : \Phi(x) = \kappa(\cdot, x)$ is called the feature map of $\mathcal{H}$. In the case of complex spaces (i.e., $\mathbb{F} = \mathbb{C}$), the inner product is sesquilinear (i.e., linear in one argument and antilinear in the other) and Hermitian, i.e., $\kappa(x, y) = (\langle \kappa(\cdot, x), \kappa(\cdot, y) \rangle_{\mathcal{H}})^* = \kappa^*(y, x)$. In the real case, however, this is simplified to $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x) \rangle_{\mathcal{H}} = \langle \kappa(\cdot, x), \kappa(\cdot, y) \rangle_{\mathcal{H}}$. In the following, we will denote a complex RKHS by $\mathbb{H}$ and a real RKHS by $\mathcal{H}$. Moreover, to distinguish the two cases, we will use the notations $\kappa_{\mathbb{R}}$ and $\Phi_{\mathbb{R}}$ to refer to a real kernel and its corresponding feature map, instead of the notation $\kappa_{\mathbb{C}}$, $\Phi_{\mathbb{C}}$, which is reserved for pure complex kernels.

A variety of kernel functions can be found in the respective literature [1], [4], [6], [37], [38]. In this paper, we will use the popular real Gaussian kernel, i.e., $\kappa_{\mathbb{R}^\nu, t}(x, y) := \exp\left(-t \sum_{k=1}^{\nu} (x_k - y_k)^2\right)$, defined for $x, y \in \mathbb{R}^\nu$, and the complex Gaussian kernel, i.e., $\kappa_{\mathbb{C}^\nu, t}(z, w) := \exp\left(-t \sum_{k=1}^{\nu} (z_k - w_k^*)^2\right)$, where $z, w \in \mathbb{C}^\nu$, z_k denotes the kth component of the complex vector $z \in \mathbb{C}^\nu$ and $\exp(\cdot)$ is the extended exponential function in the complex domain. In both cases, t is a free positive parameter that controls the shape of the kernel.

Besides the complex RKHS produced by the associated complex kernels, such as the aforementioned ones, one may construct a complex RKHS as a Cartesian product of a real RKHS with itself, in a fashion similar to the identification of the field of complex numbers, $\mathbb{C}$, to $\mathbb{R}^2$. This technique is called complexification of a real RKHS and the respective Hilbert space is called complexified RKHS. Let $\mathcal{X} \subseteq \mathbb{R}^\nu$ and define the spaces $\mathcal{X}^2 \equiv \mathcal{X} \times \mathcal{X} \subseteq \mathbb{R}^{2\nu}$ and $\mathbb{X} = \{x + iy; x, y \in \mathcal{X}\} \subseteq \mathbb{C}^\nu$, where the latter is equipped with a complex inner product structure. Let $\mathcal{H}$ be a real RKHS associated with a real kernel $\kappa_{\mathbb{R}}$ defined on $\mathcal{X}^2 \times \mathcal{X}^2$ and let $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ be its corresponding inner product. Then, every $f \in \mathcal{H}$ can be regarded as a function defined on either $\mathcal{X}^2$ or $\mathbb{X}$, i.e., $f(z) = f(x + iy) = f(x, y)$. Moreover, we define the Cartesian product of $\mathcal{H}$ with itself, i.e., $\mathcal{H}^2 = \mathcal{H} \times \mathcal{H}$. It is easy to verify that $\mathcal{H}^2$ is also a Hilbert space with inner product

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\mathcal{H}^2} = \langle f^r, g^r \rangle_{\mathcal{H}} + \langle f^i, g^i \rangle_{\mathcal{H}} \qquad (1)$$

for $\boldsymbol{f} = (f^r, f^i)$, $\boldsymbol{g} = (g^r, g^i)$. Our objective is to enrich $\mathcal{H}^2$ with a complex structure (i.e., with a complex inner product). To this end, we define the space $\mathbb{H} = \{f = f^r + if^i; f^r, f^i \in \mathcal{H}\}$ equipped with the complex inner product

$$\langle f, g \rangle_{\mathbb{H}} = \langle f^r, g^r \rangle_{\mathcal{H}} + \langle f^i, g^i \rangle_{\mathcal{H}} + i\left(\langle f^i, g^r \rangle_{\mathcal{H}} - \langle f^r, g^i \rangle_{\mathcal{H}}\right) \qquad (2)$$

for $f = f^r + if^i$, $g = g^r + ig^i$. It is not difficult to verify that the complexified space $\mathbb{H}$ is a complex RKHS with kernel

κ [38]. We call $\mathbb{H}$ the complexification of $\mathcal{H}$. It can readily be seen that although $\mathbb{H}$ is a complex RKHS, its respective kernel is real (i.e., its imaginary part is equal to zero). To complete the presentation of the complexification procedure, we need a technique to implicitly map the data samples from the complex input space to the complexified RKHS $\mathbb{H}$. This can be done using the simple rule

$$\begin{aligned} \bar{\Phi}_{\mathbb{C}}(z) = \bar{\Phi}_{\mathbb{C}}(x + iy) &= \bar{\Phi}_{\mathbb{C}}(x, y) \\ &= \Phi_{\mathbb{R}}(x, y) + i\Phi_{\mathbb{R}}(x, y) \end{aligned} \qquad (3)$$

where $\Phi_{\mathbb{R}}$ is the feature map of the real reproducing kernel $\kappa_{\mathbb{R}}$, i.e., $\Phi_{\mathbb{R}}(x, y) = \kappa_{\mathbb{R}}(\cdot, (x, y))$ and $z = x + iy$. As a consequence, we observe that

$$\begin{aligned} \langle \bar{\Phi}_{\mathbb{C}}(z), \bar{\Phi}_{\mathbb{C}}(z') \rangle_{\mathbb{H}} &= 2\langle \Phi_{\mathbb{R}}(x, y), \Phi_{\mathbb{R}}(x', y') \rangle_{\mathcal{H}} \\ &= 2\kappa_{\mathbb{R}}((x', y'), (x, y)) \end{aligned}$$

for all $z, z' \in \mathbb{H}$. We have to emphasize that a complex RKHS $\mathbb{H}$ (whether it is constructed through the complexification procedure, or it is produced by a complex kernel) can, always, be represented as a Cartesian product of a Hilbert space with itself, i.e., we can, always, identify $\mathbb{H}$ with a double real space $\mathcal{H}^2$. Furthermore, the complex inner product of $\mathbb{H}$ can always be related to the real inner product of $\mathcal{H}$ as in (2).

To compute the gradients of real-valued cost functions, which are defined on complex domains, we adopt the rationale of Wirtinger's calculus [39]. This was brought into light recently [32], [33], [40] as a means to compute, in an efficient and elegant way, gradients of real-valued cost functions that are defined on complex domains ($\mathbb{C}^\nu$), in the context of widely linear processing [34], [41]. 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 [16], 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. In [42], the notion of Wirtinger calculus was extended to include subgradients in RKHS.

III. COMPLEX SUPPORT VECTOR REGRESSION

We begin the treatment of the complex case with the complex SVR rationale, as this is a direct generalization of the real SVR. Suppose we are given training data of the form $\{(z_n, d_n); n = 1, \ldots, N\} \subset \mathcal{X} \times \mathbb{C}$, where $\mathcal{X} = \mathbb{C}^\nu$ denotes the space of input patterns. As z_n is complex, we denote its real part by x_n and its imaginary part by y_n, respectively, i.e., $z_n = x_n + iy_n$, $n = 1, \ldots, N$. Similarly, we denote the real and the imaginary parts of d_n, i.e., $d_n = d_n^r + id_n^i$, $n = 1, \ldots, N$ by d_n^r and d_n^i.
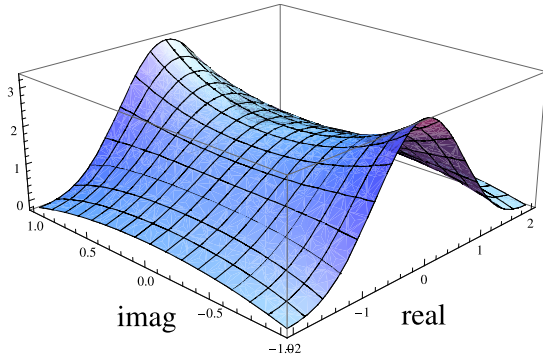
Fig. 1. Function $\kappa_{\mathbb{C}}^r((\cdot, \cdot)^T, (0, 0)^T)$ of the induced real feature space of the complex Gaussian kernel.

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 real channel (DRC) approach [34]. That is, the training data are split into two sets $\{((x_n, y_n)^T, d_n^r); \; n = 1, \ldots, N\} \subset \mathbb{R}^{2\nu} \times \mathbb{R}$ and $\{((x_n, y_n)^T, d_n^i); \; n = 1, \ldots, N\} \subset \mathbb{R}^{2\nu} \times \mathbb{R}$, and an SVR is performed on each set of data using a real kernel $\kappa_{\mathbb{R}}$ and its corresponding RKHS. We will show in the following sections that the DRC approach is equivalent to the complexification procedure [16] described in Section II. The latter, however, often provides a context that enables us to work with complex data compactly and elegantly, as one may employ Wirtinger calculus to compute the respective gradients and develop algorithms directly in complex form [16].

In contrast to the complexification procedure, we emphasize that the pure complex approach (where one directly exploits a complex RKHS) considered in the next subsection is quite different from the DRC 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 (Fig. 1). We demonstrated in [31] and [42], although in a different context than the one we use here, that the DRC approach and the pure complex approaches give, in general, different results. Depending on the case, the pure complex approach might show increased performance over the DRC approach and vice versa.

B. Pure Complex SVR

Prior to the development of the generalized complex SVR rationale, we investigate some significant properties of the complex kernels. In the following, we assume that $\mathbb{H}$ is a complex RKHS with kernel $\kappa_{\mathbb{C}}$. We can decompose $\kappa_{\mathbb{C}}$ into its real and imaginary parts, i.e., $\kappa_{\mathbb{C}}(z, z') = \kappa_{\mathbb{C}}^r(z, z') + i \kappa_{\mathbb{C}}^i(z, z')$, where $\kappa_{\mathbb{C}}^r(z, z'), \kappa_{\mathbb{C}}^i(z, z') \in \mathbb{R}$. As any complex kernel is Hermitian (Section II), we have $\kappa_{\mathbb{C}}^*(z, z') = \kappa_{\mathbb{C}}(z', z)$

and hence we take

$$\kappa_{\mathbb{C}}^r(z, z') = \kappa_{\mathbb{C}}^r(z', z) \tag{4}$$
$$\kappa_{\mathbb{C}}^i(z, z') = -\kappa_{\mathbb{C}}^i(z', z). \tag{5}$$

Lemma 3.1: The imaginary part of any complex kernel, $\kappa_{\mathbb{C}}$, satisfies

$$\sum_{n,m=1}^{N} c_n c_m \kappa_{\mathbb{C}}^i(z_n, z_m) = 0 \tag{6}$$

for any $N > 0$ and any selection of $c_1, \ldots, c_N \in \mathbb{C}$ and $z_1, \ldots, z_N \in \mathcal{X}$.

Proof: Exploiting (5) and rearranging the indices of the summation we get

$$\sum_{n,m=1}^{N} c_n c_m \kappa_{\mathbb{C}}^i(z_n, z_m) = -\sum_{n,m=1}^{N} c_n c_m \kappa_{\mathbb{C}}^i(z_m, z_n)$$
$$= -\sum_{m,n=1}^{N} c_m c_n \kappa_{\mathbb{C}}^i(z_n, z_m).$$

Hence, $2 \sum_{n,m=1}^{N} c_n c_m \kappa_{\mathbb{C}}^i(z_n, z_m) = 0$ and the result follows immediately. ∎

Lemma 3.2: If $\kappa_{\mathbb{C}}(z, z')$ is a complex kernel defined on $\mathbb{C}^\nu \times \mathbb{C}^\nu$, then its real part

$$\kappa_{\mathbb{C}}^r\left(\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x' \\ y' \end{pmatrix}\right) = \mathrm{Re}(\kappa_{\mathbb{C}}(z, z')) \tag{7}$$

where $z = x + iy$, $z' = x' + iy'$, is a real kernel defined on $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$. We call this kernel the induced real kernel of $\kappa_{\mathbb{C}}$.

Proof: As relation (4) implies, $\kappa_{\mathbb{C}}^r$ is symmetric. Moreover, let $N > 0$, $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$ and $z_1, \ldots, z_N \in \mathcal{X}$. As $\kappa_{\mathbb{C}}$ is positive definite, we have

$$\sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}(z_n, z_m) \geq 0.$$

However, splitting $\kappa_{\mathbb{C}}$ into its real and imaginary parts and exploiting Lemma 3.1, we take

$$\sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}(z_n, z_m)$$
$$= \sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}^r(z_n, z_m) + i \sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}^i(z_n, z_m)$$
$$= \sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}^r(z_n, z_m).$$

Hence

$$\sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}^r(z_n, z_m) \geq 0.$$

As a last step, recall that $\kappa_{\mathbb{C}}^r$ may be regarded as defined either on $\mathbb{C}^\nu \times \mathbb{C}^\nu$ or $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$. This leads to

$$\sum_{n,m=1}^{N} \alpha_n \alpha_m \kappa_{\mathbb{C}}^r\left(\begin{pmatrix} x_n \\ y_n \end{pmatrix}, \begin{pmatrix} x_m \\ y_m \end{pmatrix}\right) \geq 0.$$

We conclude that $\kappa_{\mathbb{C}}^r$ is a positive definite kernel on $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$. ∎

At this point, we are ready to present the SVR rationale in complex RKHS. We transform the input data from $\mathcal{X}$ to $\mathbb{H}$, via the feature map $\Phi_{\mathbb{C}}$, to obtain the data $\{(\Phi_{\mathbb{C}}(z_n), d_n);$ $n = 1, \ldots, N\}$. In analogy with the real case and extending the principles of widely linear estimation to complex SVR, the goal is to find a function $T : \mathbb{H} \rightarrow \mathbb{C} : T(f) = \langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c$, 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 $n = 1, \ldots, N$. We emphasize that we employ the widely linear estimation function $S_1 : \mathbb{H} \rightarrow \mathbb{C} : S_1(f) = \langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}}$ instead of the usual complex linear function[3] $S_2 : \mathbb{H} \rightarrow \mathbb{C} : S_2(f) = \langle f, w \rangle_{\mathbb{H}}$ following the ideas of [41], which are becoming popular in complex signal processing [43]–[45] and have been generalized for the case of complex RKHS in [42]. It has been established [46], [47] that the widely linear estimation functions are able to capture the second-order statistical characteristics of the input data, which are necessary if noncircular[4] input sources are considered. Furthermore, as it has been shown in [31], the exploitation of the traditional complex linear function excludes a significant percentage of linear functions from being considered in the estimation process. The correct and natural linear estimation in complex spaces is the widely linear one.

Observe that at the training points, i.e., $\Phi_{\mathbb{C}}(z_n)$, T takes the values $T(\Phi_{\mathbb{C}}(z_n))$. Following similar arguments as with the real case, this is equivalent with finding a complex nonlinear function g defined on $\mathcal{X}$ such that

$$g(z) = T \circ \Phi_{\mathbb{C}}(z) = \langle \Phi_{\mathbb{C}}(z), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z), v \rangle_{\mathbb{H}} + c \quad (8)$$

for some $w, v \in \mathbb{H}$, $c \in \mathbb{C}$, which satisfies the aforementioned properties. We formulate the complex SVR task as follows:

$$\min_{w,v,c} \quad \frac{1}{2}\|w\|_{\mathbb{H}}^2 + \frac{1}{2}\|v\|_{\mathbb{H}}^2 + \frac{C}{N}\sum_{n=1}^N (\xi_n^r + \hat{\xi}_n^r + \xi_n^i + \hat{\xi}_n^i)$$

$$\text{s. t.} \begin{cases} \text{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c - d_n) \leq \epsilon + \xi_n^r \\ \text{Re}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) \leq \epsilon + \hat{\xi}_n^r \\ \text{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c - d_n) \leq \epsilon + \xi_n^i \\ \text{Im}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) \leq \epsilon + \hat{\xi}_n^i \\ \qquad\qquad\qquad\qquad \xi_n^r, \hat{\xi}_n^r, \xi_n^i, \hat{\xi}_n^i \geq 0 \end{cases}$$

$$(9)$$

---

[3]All other attempts to generalize the SVR rationale to complex and hyper complex spaces employed the standard complex linear function S_2.

[4]Note that the issue of circularity has become quite popular recently in the context of complex adaptive filtering. Circularity is intimately related to rotation in the geometric sense. A complex random variable Z is called circular, if for any angle ϕ both Z and $Ze^{i\phi}$ (i.e., the rotation of Z by angle ϕ) follow the same probability distribution [34].

To solve (9), we derive the Lagrangian and the KKT conditions to obtain the dual problem. Thus we take

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + \frac{1}{2}\|v\|^2 + \frac{C}{N}\sum_{n=1}^N (\xi_n^r + \hat{\xi}_n^r + \xi_n^i + \hat{\xi}_n^i)$$

$$+ \sum_{n=1}^N a_n \big(\text{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c - d_n) - \epsilon - \xi_n^r \big)$$

$$+ \sum_{n=1}^N \hat{a}_n \big(\text{Re}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) - \epsilon - \hat{\xi}_n^r \big)$$

$$+ \sum_{n=1}^N b_n \big(\text{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c - d_n) - \epsilon - \xi_n^i \big)$$

$$+ \sum_{n=1}^N \hat{b}_n \big(\text{Im}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) - \epsilon + \hat{\xi}_n^i \big)$$

$$- \sum_{n=1}^N \eta_n \xi_n^r - \sum_{n=1}^N \hat{\eta}_n \hat{\xi}_n^r - \sum_{n=1}^N \theta_n \xi_n^i - \sum_{n=1}^N \hat{\theta}_n \hat{\xi}_n^i \quad (10)$$

where a_n, $\hat{a}_n$, b_n, $\hat{b}_n$, η_n, $\hat{\eta}_n$, θ_n, and $\hat{\theta}_n$ are the Lagrange multipliers. To exploit the saddle point conditions, we employ the rules of Wirtinger calculus for the complex variables on complex RKHS as described in [16] and deduce that

$$\frac{\partial \mathcal{L}}{\partial w^*} = \frac{1}{2}w + \frac{1}{2}\sum_{n=1}^N a_n \Phi_{\mathbb{C}}(z_n) - \frac{1}{2}\sum_{n=1}^N \hat{a}_n \Phi_{\mathbb{C}}(z_n)$$

$$- \frac{i}{2}\sum_{n=1}^N b_n \Phi_{\mathbb{C}}(z_n) + \frac{i}{2}\sum_{n=1}^N \hat{b}_n \Phi_{\mathbb{C}}(z_n)$$

$$\frac{\partial \mathcal{L}}{\partial v^*} = \frac{1}{2}v + \frac{1}{2}\sum_{n=1}^N a_n \Phi_{\mathbb{C}}^*(z_n) - \frac{1}{2}\sum_{n=1}^N \hat{a}_n \Phi_{\mathbb{C}}^*(z_n)$$

$$- \frac{i}{2}\sum_{n=1}^N b_n \Phi_{\mathbb{C}}^*(z_n) + \frac{i}{2}\sum_{n=1}^N \hat{b}_n \Phi_{\mathbb{C}}^*(z_n)$$

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

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

$$\frac{\partial \mathcal{L}}{\partial \xi_n^r} = \frac{C}{N} - a_n - \eta_n, \quad \frac{\partial \mathcal{L}}{\partial \hat{\xi}_n^r} = \frac{C}{N} - \hat{a}_n - \hat{\eta}_n$$

$$\frac{\partial \mathcal{L}}{\partial \xi_n^i} = \frac{C}{N} - b_n - \theta_n, \quad \frac{\partial \mathcal{L}}{\partial \hat{\xi}_n^i} = \frac{C}{N} - \hat{b}_n - \theta_n$$

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

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

$$w = \sum_{n=1}^N (\hat{a}_n - a_n)\Phi_{\mathbb{C}}(z_n) - i\sum_{n=1}^N (\hat{b}_n - b_n)\Phi_{\mathbb{C}}(z_n) \quad (11)$$

$$v = \sum_{n=1}^N (\hat{a}_n - a_n)\Phi_{\mathbb{C}}^*(z_n) - i\sum_{n=1}^N (\hat{b}_n - b_n)\Phi_{\mathbb{C}}^*(z_n) \quad (12)$$

Pure Complex SVR
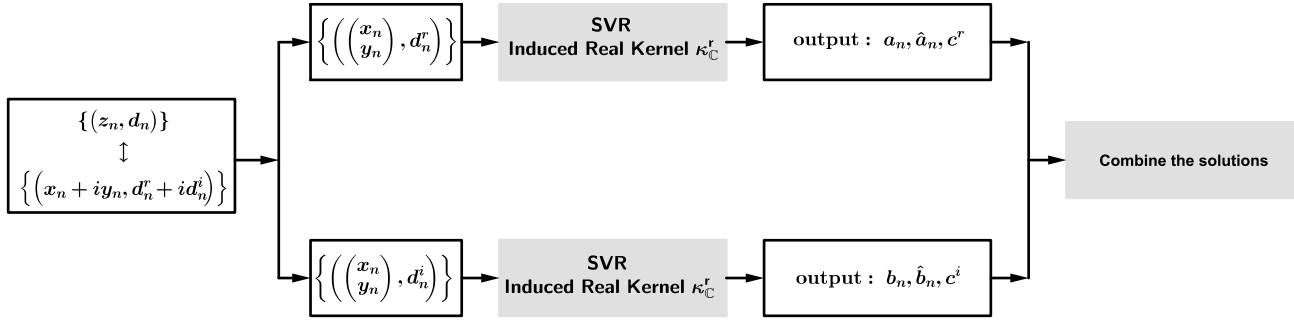**with Complex Kernel $\kappa_{\mathbb{C}}$**

Fig. 2. Pure complex SVR. The difference with the dual channel approach is due to the incorporation of the induced real kernel $\kappa_{\mathbb{C}}^r$, 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.

$$\sum_{n=1}^{N}(\hat{a}_n - a_n) = \sum_{n=1}^{N}(\hat{b}_n - b_n) = 0 \tag{13}$$

$$\eta_n = \frac{C}{N} - a_n \quad \hat{\eta}_n = \frac{C}{N} - \hat{a}_n$$

$$\theta_n = \frac{C}{N} - b_n \quad \hat{\theta}_n = \frac{C}{N} - \hat{b}_n \tag{14}$$

for $n = 1, \dots, N$.

To compute $|w|_{\mathbb{H}}^2 = \langle w, w \rangle_{\mathbb{H}}$, we apply (11), Lemma 3.1, the reproducing property of $\mathbb{H}$, i.e., $\langle \Phi(z_n), \Phi(z_m) \rangle_{\mathbb{H}} = \kappa_{\mathbb{C}}(z_m, z_n)$, and the sesqui-linear property of the inner product of $\mathbb{H}$ to obtain

$$\|w\|_{\mathbb{H}}^2 = \sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$+ \sum_{n,m=1}^{N}(\hat{b}_n - b_n)(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$+ 2\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^i(z_m, z_n).$$

Similarly, we have

$$\|v\|_{\mathbb{H}}^2 = \sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$+ \sum_{n,m=1}^{N}(\hat{b}_n - b_n)(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$- 2\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^i(z_m, z_n)$$

and

$$\frac{\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}}}{2}$$

$$= \sum_{m=1}^{N}(\hat{a}_m - a_m)\kappa_{\mathbb{C}}^r(z_m, z_n) + i\sum_{m=1}^{N}(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^r(z_m, z_n).$$

Eliminating η_n, $\hat{\eta}_n$, θ_n, $\hat{\theta}_n$ via (14), and w, v via the aforementioned relations, we obtain the final form of the

Lagrangian

$$\mathcal{L} = -\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$- \sum_{n,m=1}^{N}(\hat{b}_n - b_n)(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$- \epsilon\sum_{n=1}^{N}(a_n + \hat{a}_n + b_n + \hat{b}_n)$$

$$+ \sum_{n=1}^{N}d_n^r(\hat{a}_n - a_n) + \sum_{n=1}^{N}d_n^i(\hat{b}_n - b_n) \tag{15}$$

where d_n^r, d_n^i are the real and imaginary parts of the output d_n, $n = 1, \dots, N$. This means that we can split the dual problem into two separate maximization tasks:

$$\operatorname*{maximize}_{a,\hat{a}} \begin{cases} -\displaystyle\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_{\mathbb{C}}^r(z_m, z_n) \\ -\epsilon\displaystyle\sum_{n=1}^{N}(\hat{a}_n + a_n) + \sum_{n=1}^{N}d_n^r(\hat{a}_n - a_n) \end{cases}$$

$$\text{s.t.} \ \sum_{n=1}^{N}(\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N] \tag{16a}$$

and

$$\operatorname*{maximize}_{b,\hat{b}} \begin{cases} -\displaystyle\sum_{n,m=1}^{N}(\hat{b}_n - b_n)(\hat{b}_m - b_m)\kappa_{\mathbb{C}}^r(z_m, z_n) \\ -\epsilon\displaystyle\sum_{n=1}^{N}(\hat{b}_n + b_n) + \sum_{n=1}^{N}d_n^i(\hat{b}_n - b_n) \end{cases}$$

$$\text{s.t} \ \sum_{n=1}^{N}(\hat{b}_n - b_n) = 0 \text{ and } b_n, \hat{b}_n \in [0, C/N]. \tag{16b}$$

Observe that (16a) and (16b) are equivalent with the dual problem of a standard real SVR task with kernel $2\kappa_{\mathbb{C}}^r$. This is a real kernel, as Lemma 3.2 establishes. Therefore (Fig. 2), one may solve the two real SVR tasks for a_n, $\hat{a}_n$, c^r and b_n, $\hat{b}_n$, c^i, respectively, using any one of the algorithms, which have
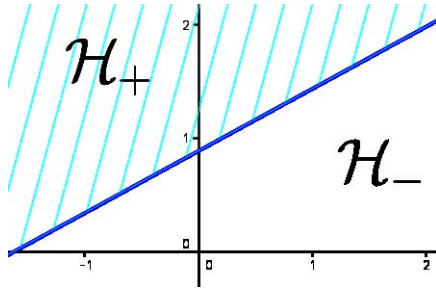
Fig. 3.   Hyperplane separates the space $\mathcal{H}$ into two parts, $\mathcal{H}_+$ and $\mathcal{H}_-$.

been developed for this purpose, and then combine the two solutions to find the final nonlinear solution of the complex problem as

$$g(z) = \langle \Phi_{\mathbb{C}}(z), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z), v \rangle_{\mathbb{H}} + c$$

$$= 2 \sum_{n=1}^{N} (\hat{a}_n - a_n) \kappa_{\mathbb{C}}^r (z_n, z)$$

$$+ 2i \sum_{n=1}^{N} (\hat{b}_n - b_n) \kappa_{\mathbb{C}}^r (z_n, z) + c. \qquad (17)$$

In this paper, we are focusing mainly on the complex Gaussian kernel. It is important to emphasize that, in this case, the induced kernel $\kappa_{\mathbb{C}}^r$ is not the real Gaussian RBF. Fig. 1 shows the element $\kappa_{\mathbb{C}}^r(\cdot, (0,0)^T)$ of the induced real feature space.

Remark 3.1: For the complexification procedure, we select a real kernel $\kappa_{\mathbb{R}}$ and transform the input data from $\mathcal{X}$ to the complexified space $\mathbb{H}$, via the feature map $\bar{\Phi}_{\mathbb{C}}$, to obtain the data $\{(\bar{\Phi}_{\mathbb{C}}(z_n), d_n); \; n = 1, \dots, N\}$. Following a similar procedure as the one described above and considering that

$$\langle \bar{\Phi}_{\mathbb{C}}(z_n), \bar{\Phi}_{\mathbb{C}}(z_m) \rangle_{\mathbb{H}} = 2\kappa_{\mathbb{R}}(z_m, z_n)$$

we can easily deduce that the dual of the complexified SVR task is equivalent to two real SVR tasks employing the kernel $2\kappa_{\mathbb{R}}$. Hence, the complexification technique is identical to the DRC approach.

IV. COMPLEX SUPPORT VECTOR MACHINE

A. Complex Hyperplanes

Recall that in any real Hilbert space $\mathcal{H}$, a hyper plane consists of all the elements $f \in \mathcal{H}$ that satisfy

$$\langle f, w \rangle_{\mathcal{H}} + c = 0 \qquad (18)$$

for some $w \in \mathcal{H}$, $c \in \mathbb{R}$. Moreover, as Fig. 3 shows, any hyperplane of $\mathcal{H}$ divides the space into two parts, $\mathcal{H}_+ = \{f \in \mathcal{H}; \; \langle f, w \rangle_{\mathcal{H}} + c > 0\}$ and $\mathcal{H}_- = \{f \in \mathcal{H}; \; \langle f, w \rangle_{\mathcal{H}} + c < 0\}$. In the traditional SVM classification task, the goal is to separate two distinct classes of data by a maximum margin hyper plane, so that one class falls into $\mathcal{H}_+$ and the other into $\mathcal{H}_-$ (excluding some outliers). To be able to generalize the SVM rationale to complex spaces, firstly, we need to determine an appropriate definition for a complex hyper plane. The difficulty is that the set of complex numbers is not an ordered one, and thus one may not assume that a complex version of (18) divides the space into two parts, as $\mathcal{H}_+$ and $\mathcal{H}_-$ cannot be defined. Instead, we will provide a novel definition

of complex hyper planes that divide the complex space into four parts. This will be our kick off point for deriving the complex SVM rationale, which classifies objects into four (instead of two) classes.

Lemma 4.1: The relations

$$\mathrm{Re}(\langle f, w \rangle_{\mathbb{H}} + c) = 0 \qquad (19a)$$

$$\mathrm{Im}(\langle f, w \rangle_{\mathbb{H}} + c) = 0 \qquad (19b)$$

for some $w \in \mathbb{H}$, $c \in \mathbb{C}$, where $f \in \mathbb{H}$, represent two orthogonal hyper planes of the doubled real space, i.e., $\mathcal{H}^2$, in general positions.

Proof: Observe that

$$\langle f, w \rangle_{\mathbb{H}} = \langle f^r, w^r \rangle_{\mathcal{H}} + \langle f^i, w^i \rangle_{\mathcal{H}} + i(\langle f^i, w^r \rangle_{\mathcal{H}} - \langle f^r, w^i \rangle_{\mathcal{H}})$$

where $f = f^r + if^i$, $w = w^r + iw^i$. Hence, we take

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} w^r \\ w^i \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^r = 0$$

and

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} -w^i \\ w^r \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^i = 0$$

where $c = c^r + ic^i$. These are two distinct hyper planes of $\mathcal{H}^2$. Moreover, as

$$\begin{pmatrix} -w^i & w^r \end{pmatrix} \begin{pmatrix} w^r \\ w^i \end{pmatrix} = 0$$

the two hyper planes are orthogonal. ∎

Lemma 4.2: The relations

$$\mathrm{Re}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c\right) = 0 \qquad (20a)$$

$$\mathrm{Im}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c\right) = 0 \qquad (20b)$$

for some $w, v \in \mathbb{H}$, $c \in \mathbb{C}$, where $f \in \mathbb{H}$, represent two hyper planes of the doubled real space, i.e., $\mathcal{H}^2$. Depending on the values of w, v, these hyper planes may be placed arbitrarily on $\mathcal{H}^2$.

Proof: Following a similar rationale as in the proof of Lemma 4.1, we take:

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} w^r + v^r \\ w^i - v^i \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^r = 0$$

and

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} -(w^i + v^i) \\ w^r - v^r \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^i = 0$$

where $f = f^r + if^i$, $w = w^r + iw^i$, $v = v^r + iv^i$, $c = c^r + ic^i$. ∎

The following definition comes naturally.

Definition 4.1: Let $\mathbb{H}$ be a complex Hilbert space. We define the complex pair of hyper planes as the set of all $f \in \mathbb{H}$ that satisfy one of the following relations:

$$\mathrm{Re}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c\right) = 0 \qquad (21a)$$

$$\mathrm{Im}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c\right) = 0 \qquad (21b)$$

for some $w, v \in \mathbb{H}$, $c \in \mathbb{C}$.

Lemmas 4.1 and 4.2 demonstrate the significant difference between complex linear estimation and widely linear
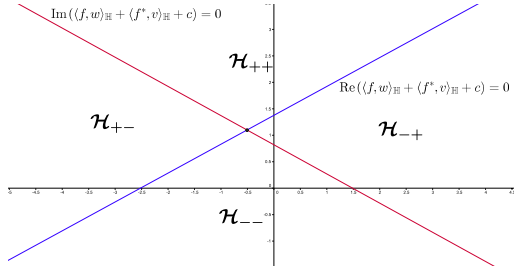
Fig. 4. Complex pair of hyper planes separates the space of complex numbers (i.e., $\mathbb{H} = \mathbb{C}$) into four parts.
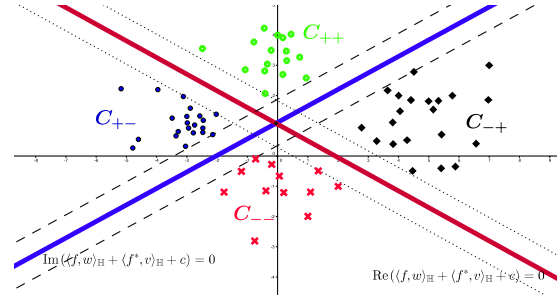

Fig. 5. Complex pair of hyper planes that separates the four given classes. The hyper planes are chosen so as to maximize the margin between the classes.

estimation functions, which have been, already, pointed out in Section III-B, albeit in a different context. The complex linear case is quite restrictive, as the complex hyper planes are always orthogonal. On the other hand, the widely linear case is more general and covers all cases. The complex pair of hyper planes (as defined by definition 4.1) divides the space into four parts

$$\mathcal{H}_{++} = \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) > 0 \\ \operatorname{Im}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) > 0 \end{array} \right\}$$

$$\mathcal{H}_{+-} = \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) > 0 \\ \operatorname{Im}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) < 0 \end{array} \right\}$$

$$\mathcal{H}_{-+} = \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) < 0 \\ \operatorname{Im}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) > 0 \end{array} \right\}$$

$$\mathcal{H}_{--} = \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) < 0 \\ \operatorname{Im}\left(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c \right) < 0 \end{array} \right\}.$$

Fig. 4 demonstrates a simple case of a complex pair of hyper planes that divides $\mathbb{C}$ into four parts. Note that, in some cases, the complex pair of hyper planes might degenerate into two identical hyper planes or two parallel hyper planes.

B. Quaternary Complex SVM

The complex SVM classification task can be formulated as follows. Suppose we are given training data, which belong to four separate classes $C_{++}, C_{+-}, C_{-+}, C_{--}$, i.e., $\{(z_n, d_n); \ n = 1, \ldots, N\} \subset \mathcal{X} \times \{\pm 1 \pm i\}$. If $d_n = +1+i$, then the nth sample belongs to C_{++}, i.e., $z_n \in C_{++}$, if $d_n = 1-i$, then $z_n \in C_{+-}$, if $d_n = -1+i$, then $z_n \in C_{-+}$ and if $d_n = -1-i$, then $z_n \in C_{--}$. Consider the complex RKHS $\mathbb{H}$ with respective kernel $\kappa_{\mathbb{C}}$. Following a similar rationale to the real case, we transform the input data from $\mathcal{X}$ to $\mathbb{H}$, via the feature map $\Phi_{\mathbb{C}}$. The goal of the SVM task is to estimate a complex pair of maximum margin hyper planes that separates the points of the four classes (Fig. 5). Thus, we need to minimize

$$\left\| \begin{pmatrix} w^r + v^r \\ w^i - v^i \end{pmatrix} \right\|^2_{\mathcal{H}^2} + \left\| \begin{pmatrix} -(w^i + v^i) \\ w^r - v^r \end{pmatrix} \right\|^2_{\mathcal{H}^2}$$
$$= \|w^r + v^r\|^2_{\mathcal{H}} + \|w^i - v^i\|^2_{\mathcal{H}} + \|(w^i + v^i)\|^2_{\mathcal{H}} + \|w^r - v^r\|^2_{\mathcal{H}}$$
$$= 2\|w^r\|^2_{\mathcal{H}} + 2\|w^i\|^2_{\mathcal{H}} + 2\|v^r\|^2_{\mathcal{H}} + 2\|v^i\|^2_{\mathcal{H}}$$
$$= 2\left(\|w\|^2_{\mathbb{H}} + \|v\|^2_{\mathbb{H}} \right).$$

Therefore, the primal complex SVM optimization problem can be formulated as

$$\min_{w,v,c} \quad \frac{1}{2}\|w\|^2_{\mathbb{H}} + \frac{1}{2}\|v\|^2_{\mathbb{H}} + \frac{C}{N}\sum_{n=1}^{N}(\xi^r_n + \xi^i_n)$$
$$\text{s.t.} \begin{cases} d^r_n \operatorname{Re}\left(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi^*_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c \right) \geq 1 - \xi^r_n \\ d^i_n \operatorname{Im}\left(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi^*_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + c \right) \geq 1 - \xi^i_n \\ \qquad\qquad\qquad\qquad\qquad\qquad \xi^r_n, \xi^i_n \geq 0 \end{cases}$$
$$\text{for } n = 1, \ldots, N. \tag{22}$$

The Lagrangian function becomes

$$L(w, v, \mathbf{a}, \mathbf{b}) = \frac{1}{2}\|w\|^2_{\mathbb{H}} + \frac{1}{2}\|v\|^2_{\mathbb{H}} + \frac{C}{N}\sum_{n=1}^{N}(\xi^r_n + \xi^i_n)$$
$$- \sum_{n=1}^{N} a_n \left(d^r_n \operatorname{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi^*_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c) - 1 + \xi^r_n \right)$$
$$- \sum_{n=1}^{N} b_n \left(d^i_n \operatorname{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi^*_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + c) - 1 + \xi^i_n \right)$$
$$- \sum_{n=1}^{N} \eta_n \xi^r_n - \sum_{n=1}^{N} \theta_n \xi^i_n$$

where a_n, b_n, η_n, and θ_n are the positive Lagrange multipliers of the respective inequalities, for $n = 1, \ldots, N$. To exploit the saddle point conditions of the Lagrangian function, we employ the rules of Wirtinger calculus to compute the respective gradients. Hence, we take

$$\frac{\partial L}{\partial w^*} = \frac{1}{2}w - \frac{1}{2}\sum_{n=1}^{N} a_n d^r_n \Phi_{\mathbb{C}}(z_n) + \frac{i}{2}\sum_{n=1}^{N} b_n d^i_n \Phi_{\mathbb{C}}(z_n)$$

$$\frac{\partial L}{\partial v^*} = \frac{1}{2}v - \frac{1}{2}\sum_{n=1}^{N} a_n d^r_n \Phi^*_{\mathbb{C}}(z_n) + \frac{i}{2}\sum_{n=1}^{N} b_n d^i_n \Phi^*_{\mathbb{C}}(z_n)$$

$$\frac{\partial L}{\partial c^*} = \frac{1}{2}\sum_{n=1}^{N} a_n d^r_n + \frac{i}{2}\sum_{n=1}^{N} b_n d^i_n$$

and

$$\frac{\partial L}{\partial \xi^r_n} = \frac{C}{N} - a_n - \eta_n, \quad \frac{\partial L}{\partial \xi^i_n} = \frac{C}{N} - b_n - \theta_n$$

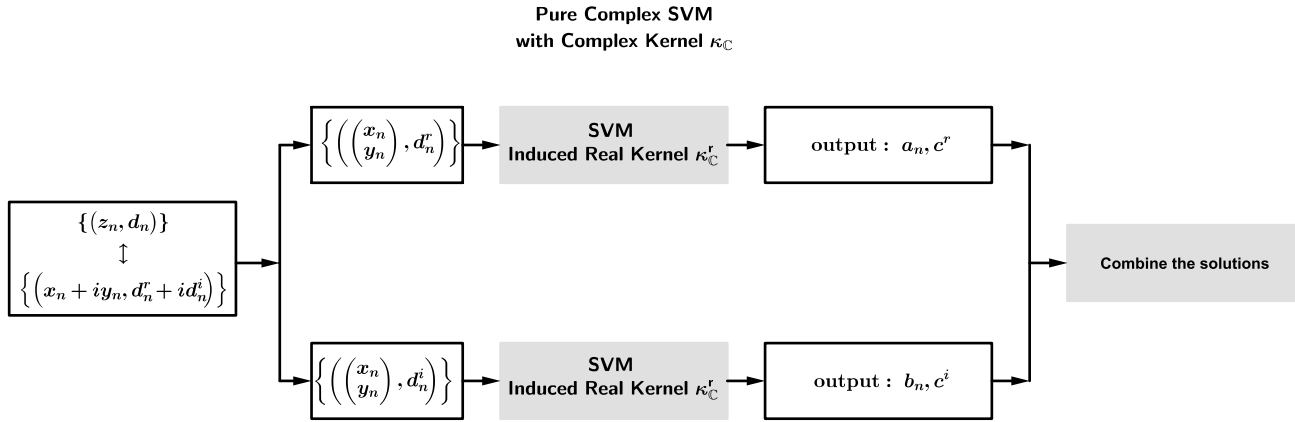for $n = 1, \ldots, N$. As all the gradients have to vanish,

Pure Complex SVM
with Complex Kernel $\kappa_{\mathbb{C}}$

Fig. 6.   Pure complex support vector machine.

we finally take

$$w = \sum_{n=1}^{N}(a_n d_n^r - i b_n d_n^i)\Phi_{\mathbb{C}}(z_n)$$

$$v = \sum_{n=1}^{N}(a_n d_n^r - i b_n d_n^i)\Phi_{\mathbb{C}}^*(z_n)$$

$$\sum_{n=1}^{N} a_n d_n^r = \sum_{n=1}^{N} b_n d_n^i = 0$$

and

$$a_n + \eta_n = \frac{C}{N}, \quad b_n + \theta_n = \frac{C}{N}$$

for $n = 1, \ldots, N$. Following a similar procedure as in the complex SVR case, it turns out that the dual problem can be split into two separate maximization tasks:

$$\underset{a}{\text{maximize}} \quad \sum_{n=1}^{N} a_n - \sum_{n,m=1}^{N} a_n a_m d_n^r d_m^r \kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$\text{s.t.} \quad \begin{cases} \sum_{n=1}^{N} a_n d_n^r = 0 \\ 0 \le a_n \le \frac{C}{N} \quad \text{for} \;\; n = 1, \ldots, N \end{cases} \tag{23a}$$

and

$$\underset{b}{\text{maximize}} \quad \sum_{n=1}^{N} b_n - \sum_{n,m=1}^{N} b_n b_m d_n^i d_m^i \kappa_{\mathbb{C}}^r(z_m, z_n)$$

$$\text{s.t.} \quad \begin{cases} \sum_{n=1}^{N} b_n d_n^i = 0 \\ 0 \le b_n \le \frac{C}{N} \quad \text{for} \;\; n = 1, \ldots, N. \end{cases} \tag{23b}$$

Observe that, similar to the regression case, these problems are equivalent with two distinct real SVM (dual) tasks employing the induced real kernel $2\kappa_{\mathbb{C}}^r$. One may split the (output) data into their real and imaginary parts, as Fig. 6 shows, solve two real SVM tasks employing any one of the standard algorithms, and finally, combine the solutions to take

the complex labeling function

$$g(z) = \underset{i}{\text{sign}}\left(\langle \Phi_{\mathbb{C}}(z), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z), v \rangle_{\mathbb{H}} + c\right)$$

$$= \underset{i}{\text{sign}}\left(2\sum_{n=1}^{N}(a_n d_n^r + i b_n d_n^i)\kappa_{\mathbb{C}}^r(z_n, z) + c^r + i c^i\right)$$

where $\text{sign}_i(z) = \text{sign}(\text{Re}(z)) + i\,\text{sign}(\text{Im}(z))$.

Remark 4.1: Following the complexification procedure, as in Remark 3.1, we select a real kernel $\kappa_{\mathbb{R}}$ and transform the input data from $\mathcal{X}$ to the complexified space $\mathbb{H}$, via the feature map $\bar{\Phi}_{\mathbb{C}}$. We can easily deduce that the dual of the complexified SVM task is equivalent to two real SVM tasks employing the kernel $2\kappa_{\mathbb{R}}$.

Remark 4.2: It is evident that both the complex and the complexified SVM can be employed for binary classification as well. The advantage in this case is that one is able to handle complex input data in both scenarios. Moreover, the popular one-versus-one and one-versus-all strategies [1], [6], which address multiclassification problems, can be directly applied to complex inputs using either the complex or the complexified binary SVM.

V. EXPERIMENTS

To illuminate the advantages that are gained by the complex kernels and to demonstrate the performance of the proposed algorithmic schemes, we compare it with standard real-valued techniques and the DRC approach under various regression and classification scenarios. In the following, we will refer to the pure complex kernel rationale and the complexification trick, presented in this paper, using the terms CSVR (or CSVM) and complexified SVR (or complexified SVM), respectively. The DRC approach, outlined in Section III-A, will be denoted as DRC-SVR. Recall that the DRC approach is equivalent to the complexified rationale, although the latter often provides for more compact formulas and simpler representations. The following experiments were implemented in MATLAB. The respective code can be found in bouboulis.mysch.gr/kernels.html.
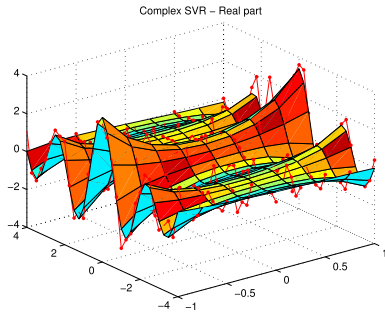
Fig. 7. Real part $(\mathrm{Re}(\mathrm{sinc}(z)))$ of the estimated sinc function from the complex SVR. The points shown in the figure are the real parts of the noisy training data used in the simulation.
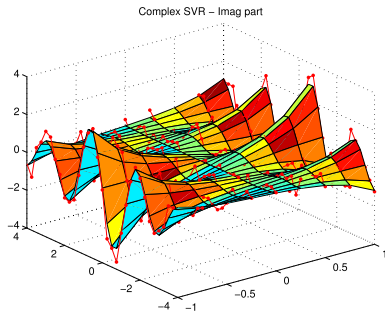

Fig. 8. Imaginary part $(\mathrm{Im}(\mathrm{sinc}(z)))$ of the estimated sinc function from the complex SVR. The points shown in the figure are the imaginary parts of the noisy training data used in the simulation.
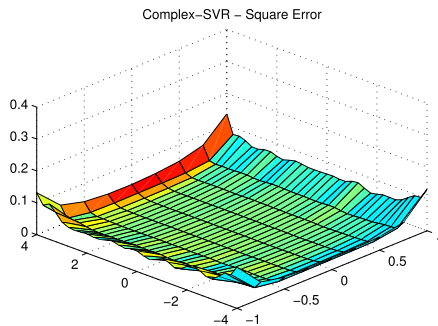

Fig. 9. MSEs (the actual values of the function minus the estimated ones) over 100 realizations of the complex SVR of the sinc function. The MSE of all the estimated values over the complete set of 100 experiments was equal to 0.054 $(-15.75$ dB).

A. SVR - Function Estimation

In this section, we perform a simple regression test on the complex function $\mathrm{sinc}(z)$. An orthogonal grid of 33×9 actual points of the sinc function, corrupted by noise, was adopted as the training data. Figs. 7 and 8 show the real and imaginary parts of the reconstructed function using the CSVR rationale. Note the excellent visual results obtained by the corrupted training data. Figs. 9 and 10 and Table I compare the square errors (i.e., $|\hat{d}_n - \mathrm{sinc}(z_n)|^2$, where $\hat{d}_n$ is the value of the estimated function at z_n) between the CSVR and the DRC-SVR over 100 different realizations of the experiment. In each realization, the sinc function was corrupted by white noise of the form $Z = X + iY$, where X and Y are real random variables following the Gaussian distribution with variances

TABLE I
MEAN SQUARE ERRORS (MSE), THE NUMBER OF SUPPORT VECTORS AND THE COMPUTING TIME OVER 100 REALIZATIONS OF THE sinc ESTIMATION EXPERIMENT

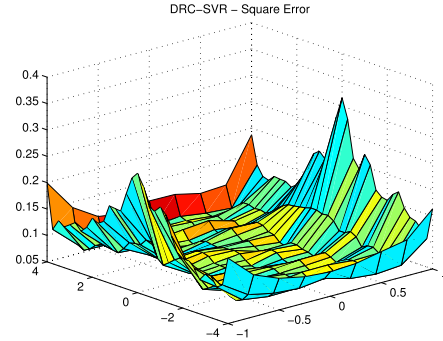|	CSVR	DRC-SVR
Mean MSE (dB)	-15.75 dB	-10.42 dB
Mean number of Support Vectors	282	282
Mean Time	179 secs	430 secs

Fig. 10. MSEs (the actual values of the function minus the estimated ones) over 100 realizations of the DRC regression of the sinc function. The MSE of all the estimated values over the complete set of 100 experiments was equal to 0.102 $(-10.42$ dB).

$\sigma_1 = 0.4$ and $\sigma_2 = 0.3$, respectively. As it is shown in Table I, the DRC-SVR fails to capture the complex structure of the function. On the other hand, the CSVR rationale provides for an estimation function, which exhibits excellent characteristics. A closer look at Figs. 9 and 10 reveals that at the border of the training grid, the square error increases in some cases. This is expected, as the available information (i.e., the neighboring points), which is exploited by the SVR algorithm, is reduced in these areas compared with the interior points of the grid. Besides the significant decrease in the square error, in these experiments, we also observed a significant reduction in the computing time needed for the CSVR compared with the DRC-SVR. In our opinion, this enhanced performance (both in terms of MSE and computational time) is due to the special structure of the sinc function. Recall that the sinc is a complex analytic function, hence it is more natural to use complex analytic functions (e.g., the complex Gaussian kernel function), instead of real analytic functions (e.g., the real Gaussian kernel), to estimate its shape. Both algorithms were implemented in MATLAB on a computer with a Core i5 650 microprocessor running at 3.2 GHz.

In all the performed experiments, the SMO algorithm was employed using the complex Gaussian kernel and the real Gaussian kernel for the CSVR and the DRC-SVR, respectively [48]. The parameters of the kernel for both the complex SVR and the DRC-SVR tasks were tuned (using cross correlation) to provide the smallestMSE. In particular for the CSVR, the parameter of the complex Gaussian kernel was set to $t = 0.3$, while for the DRC-SVR, the parameter was set to $t = 2$. In both cases, the parameters of the SVR task were set as $C = 1000$, $\epsilon = 0.1$.

TABLE II

VALUES OF C AND t THAT MINIMIZE THE MSE OF THE CSVR, FOR THE CHANNEL IDENTIFICATION TASK

C	t
1000	$1/6^2$
2000	$1/6^2$
5000	$1/8^2$
10000	$1/9^2$
20000	$1/11^2$
50000	$1/13^2$

TABLE III

VALUES OF C AND t THAT MINIMIZE THE MSE OF THE DRC-SVR, FOR THE CHANNEL IDENTIFICATION TASK

C	t
1000	$1/4^2$
2000	$1/5^2$
5000	$1/6^2$
10000	$1/7^2$
20000	$1/7^2$
50000	$1/10^2$

B. SVR - Channel Identification

In this section, we consider a nonlinear channel identification task [32]. This channel consists of the 5-tap linear component

$$t(n) = \sum_{k=1}^{5} h(k) \cdot s(n-k+1) \tag{24}$$

where

$$h(k) = 0.432\left(1 + \cos\left(\frac{2\pi(k-3)}{5}\right) - \left(1 + \cos\frac{2\pi(k-3)}{10}\right)i\right)$$

for $k = 1, \ldots, 5$, and the nonlinear component

$$x(n) = t(n) + (0.15 - 0.1i)t^2(n).$$

This is a standard model that has been extensively used in the literature for such tasks (see [16], [27], [31], [42], [49]). At the receiver's end, the signal is corrupted by white Gaussian noise and then observed as y_n. The signal-to-noise ratio (SNR) was set to 15 dB. The input signal that was fed to the channel had the form

$$s(n) = \left(\sqrt{1-\rho^2}X(n) + i\rho Y(n)\right) \tag{25}$$

where $X(n)$ and $Y(n)$ are Gaussian random variables. This input is circular for $\rho = \sqrt{2}/2$ and highly noncircular if ρ approaches zero or one [32]. The CSVR and the DRC-SVR rationales were used to address the channel identification task, which aims to discover the input-out relationship between $(s(n-L+1), s(n-L+2), \ldots, s(n))$, and $y(n)$ (the parameter L was set to $L = 5$). In each experiment, a set of 150 pairs of samples was used to perform the training. After training, a set of 600 pairs of samples was used to test the estimation's performance of both algorithms (i.e., to measure the MSE between the actual channel output, $x(n)$, and the estimated output, $\hat{x}(n)$). To find the best possible values of the parameters, C and t, that minimize the MSE for both SVR tasks, an extensive cross-validation procedure has been employed (Tables II and III) in a total of 20 sets of data. Fig. 11 shows the minimumMSE, which has been obtained for all values of the kernel parameter t versus the SVR parameter C for both cases considering a circular input (Fig. 13). It is evident that the CSVR approach significantly outperforms the DRC-SVR rationale, both in terms of MSE and computational time (Fig. 12). All the figures refer to the circular case. As the results for the noncircular case are similar, they are omitted to save space.
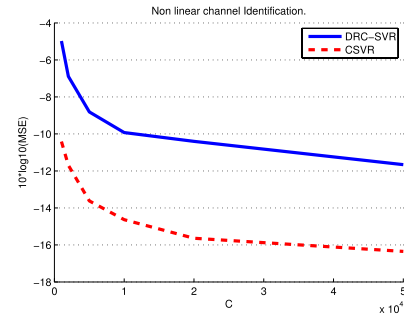

Fig. 11. MSE versus the SVR parameter C for both the CSVR and the DRC-SVR rationales, for the channel identification task.
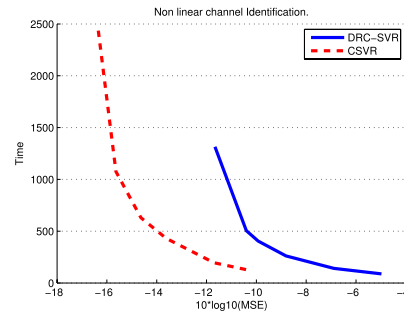

Fig. 12. Time (in seconds) versus MSE (dB) for both the CSVR and the DRC-SVR rationales, for the channel identification task.
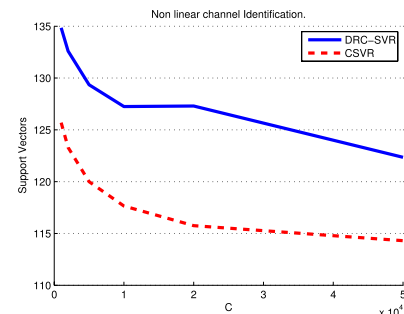

Fig. 13. Number of Support Vectors versus MSE (dB) for both the CSVR and the DRC-SVR rationales, for the channel identification task.

C. SVR - Channel Equalization

In this section, we present a nonlinear channel equalization task that consists of the linear filter (24) and the memoryless nonlinearity

$$x(n) = t(n) + (0.1 - 0.15i) \cdot t^2(n).$$

TABLE IV

VALUES OF C AND t THAT MINIMIZE THE MSE OF THE CSVR, FOR THE CHANNEL EQUALIZATION TASK

C	t
1	$1/2.5^2$
2	$1/2.5^2$
5	$1/2.5^2$
10	$1/3^2$
50	$1/4.5^2$
100	$1/5.5^2$
200	$1/6^2$
500	$1/7^2$
1000	$1/9^2$

TABLE V

VALUES OF C AND t THAT MINIMIZE THE MSE OF THE DRC-SVR, FOR THE CHANNEL EQUALIZATION TASK

C	t
1	$1/1.5^2$
2	$1/1.75^2$
5	$1/1.75^2$
10	$1/2.25^2$
50	$1/2.5^2$
100	$1/3^2$
200	$1/5^2$
500	$1/7^2$
1000	$1/7.5^2$

At the receiver end of the channel, the signal is corrupted by white Gaussian noise and then observed as $y(n)$. The signal-to-noise ratio was set to 15 dB. The input signal that was fed to the channels had the form

$$s(n) = 0.30\left(\sqrt{1 - \rho^2}X(n) + i\rho Y(n)\right) \qquad (26)$$

where $X(n)$ and $Y(n)$ are Gaussian random variables.

The aim of a channel equalization task is to construct an inverse filter, which acts on the output $y(n)$ and reproduces the original input signal as close as possible. To this end, we apply the CSVR and DRC-SVR algorithms to a set of samples of the form

$$((y(n + D), y(n + D - 1), \ldots, r(y + D - L + 1)), s(n))$$

where $L > 0$ is the filter length and D the equalization time delay (in this experiment, we set $L = 5$ and $D = 2$).

Similar to the channel identification case, in each experiment, a set of 150 pairs of samples was used to perform the training. After training, a set of 600 pairs of samples was used to test the performance of both algorithms (i.e., to measure the MSE between the actual input, $s(n)$, and the estimated input, $\hat{s}(n)$). To find the best possible values of the parameters, C and t, that minimize the MSE for both SVR tasks, an extensive cross-validation procedure has been employed (Tables IV and V) in a total of 100 sets of data. Fig. 14 shows the minimum MSE, which has been obtained for all values of the kernel parameter t, versus the SVR parameter C, for both cases considering a circular input. Figs. 15 and 16 show the computational time and the support vectors versus the MSE. The CSVR appears to achieve a slightly lower MSE for all
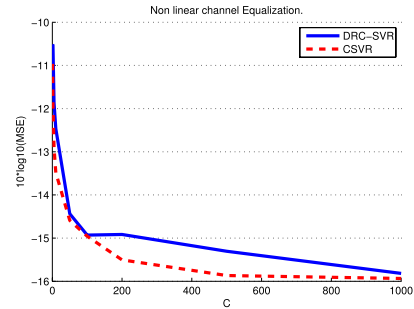

Fig. 14. MSE versus the SVR parameter C for both the CSVR and the DRC-SVR rationales, for the channel equalization task.
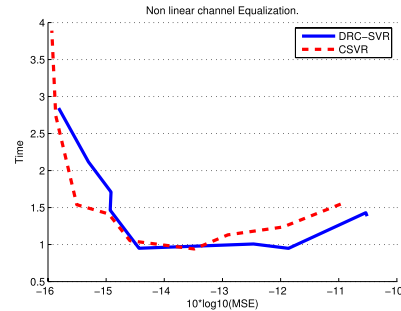

Fig. 15. Time (in seconds) versus MSE (dB) for both the CSVR and the DRC-SVR rationales, for the channel equalization task.
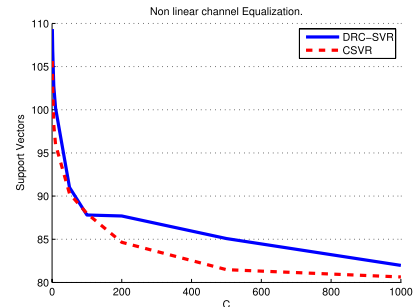

Fig. 16. Number of Support Vectors versus MSE (dB) for both the CSVR and the DRC-SVR rationales, for the channel equalization task.

values of the parameter C, at the cost of a slightly increased computational time. The results for the noncircular case are similar.

D. SVM - Multiclass Classification

We conclude the experimental section with the classification case. We performed two experiments using the popular MNIST database of handwritten digits [50]. In both cases, the respective parameters of the SVM tasks were tuned to obtain the lowest error rate possible. The MNIST database contains 60 000 handwritten digits (from zero to nine) for training and 10 000 handwritten digits for testing. Each digit is encoded as an image file with 28×28 pixels. To quantify the performance of an SVM-like learning machine on the MNIST database, one typically employs a one-versus-all strategy to the training set (using the raw pixel values as input data) and then measures the success using the testing set [51], [52].

TABLE VI

ERROR RATES OF THE ONE-VERSUS-ALL CLASSIFICATION TASK FOR
THE 10 000 TESTING DIGITS OF THE MNIST DATABASE

	raw data	complex Fourier coefficients
Error rate	3.79%	3.46%

TABLE VII

ERROR RATES OF THE ONE-VERSUS-THREE CLASSIFICATION TASK FOR
THE 0, 1, 2, AND 3 TESTING DIGITS OF THE MNIST DATABASE

	one versus three SVM	quaternary SVM
Error rate	0.721%	0.866%

In the first experiment, we compare the aforementioned standard one-versus-all scenario with a classification task that exploits complex numbers. In the complex variant, we perform a Fourier transform to each training image and keep only the 100 most significant coefficients. As these coefficients are complex numbers, we employ a one-versus-all classification task using the binary complexified SVM rationale (Remark 4.2). In both scenarios, we use the first 6000 digits of the MNIST training set (instead of the complete 60 000 digits that are included in the database) to train the learning machines and test their performances using the 10 000 digits of the testing set. In addition, we used the Gaussian kernel with $t = 1/64$ and $t = 1/140^2$, respectively. The SVM parameter C has been set equal to 100 in this case as well. The error rate of the standard real-valued scenario is 3.79%, while the error rate of the complexified (one-versus-all) SVM is 3.46% (Table VI). In both learning tasks, we used the SMO algorithm to train the SVM. The total amount of time needed to perform the training of each learning machine is almost the same for both cases (the complexified task is slightly faster).

In Section IV, we discussed how the 4-classes problem is inherent to the complex SVM. Exploiting the notion of the complex pair of hyper planes (Fig. 4), we have shown that the generalization of the SVM rationale to complex spaces directly assumes quaternary classification. Using this approach, the 4 classes problem can be solved using only two distinct SVM tasks instead of the four tasks needed by the one-versus-three or the one-versus-one strategies. The second experiment compares the quaternary complex SVM approach with the standard one-versus-three scenario using the first four digits (0, 1, 2, and 3) of the MNIST database. In both cases, we used the first 6000 such digits of the MNIST training set to train the learning machines. We tested their performance using the respective digits that are contained in the testing set. The error rate of the one-versus-three SVM was 0.721%, while the error rate of the complexified quaternary SVM was 0.866% (Table VII). However, the one-versus-three SVM task required about double the time for training, compared with the complexified quaternary SVM. This is expected, as the latter solves half as many distinct SVM tasks as the first one. In both experiments, we used the Gaussian kernel with $t = 1/49$ and $t = 1/160^2$, respectively. The SVM parameter C has been set equal to 100 in this case also.

VI. CONCLUSION

A new framework for support vector regression and quaternary classification for complex data has been presented, based on the notion of the widely linear estimation. Both complex kernels as well as complexified real ones have been used. We showed that this problem is equivalent to solving two separate real SVM tasks employing an induced real kernel (Fig. 2). The induced kernel depends on the choice of the complex kernel, and it is different to the standard kernels used in the literature. Although the machinery presented here might seem similar to the dual channel approach, there are important differences. The most important one is due to the incorporation of the induced kernel $\kappa_{\mathbb{C}}^r$, 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.

Furthermore, for the classification problem, we have shown that the complex SVM solves directly a quaternary problem, instead of the binary problem, which is associated with the real SVM. Hence, the complex SVM not only provides the means for treating complex inputs, but also offers an alternative strategy to address multiclassification problems. In this way, such problems can be solved significantly faster (the computational time is almost the half), at the cost of increased error rate. Although, in the present work, we focused on the 4 classes problem only, it is evident that the same rationale can be carried out for any multidimensional problem, where the classes must be divided into four groups each time, following a rationale similar to the one-versus-all mechanism. This will be addressed in future.

REFERENCES

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

[2] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Online learning in reproducing kernel Hilbert spaces," in *Signal Processing Theory and Machine Learning* (Academic Press Library in Signal Processing), R. Chellappa and S. Theodoridis, Eds. San Francisco, CA, USA: Academic, 2014, pp. 883–987.

[3] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[4] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. San Francisco, CA, USA: Academic, 2008.

[5] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1999.

[6] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[7] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*. Hoboken, NJ, USA: Wiley, 2010.

[8] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[9] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[10] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, Jul. 2008.

[11] W. Liu, P. P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, Feb. 2008.

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

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

[14] S. Theodoridis and M. Mavroforakis, "Reduced convex hulls: A geometric approach to support vector machines," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 119–122, May 2007.

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

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

[17] B. Schlkopf, K. Tsuda, and J.-P. Vert, *Kernel Methods in Computational Biology*. Cambridge MA, USA: MIT Press, 2004.

[18] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Proc. 7th Int. Conf. MSRS*, 1997, pp. 999–1004.

[19] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn.*, Sep. 2004, pp. 41–48.

[20] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, Jun. 2008.

[21] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Feb. 2011.

[22] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Jul. 2006.

[23] E. J. Bayro-Corrochano and N. Arana-Daniel, "Clifford support vector machines for classification, regression, and recurrence," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1731–1746, Nov. 2010.

[24] A. Shilton and D. T. H. Lai, "Quaternionic and complex-valued support vector regression for equalization and function approximation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Orlando, FL, USA, Aug. 2007, pp. 920–925.

[25] A. Shilton, D. T. H. Lai, and M. Palaniswami, "A division algebraic framework for multidimensional support vector regression," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 2, pp. 517–528, Apr. 2010.

[26] A. Shilton, D. T. B. Lai, B. K. Santhiranayagam, and M. Palaniswami, "A note on octonionic support vector regression," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 950–955, Jun. 2012.

[27] M. Sanchez-Fernandez, M. de Prado-Cumplido, J. Arenas-Garcia, and F. Perez-Cruz, "SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2298–2307, Aug. 2004.

[28] B. C. Ujang, C. C. Took, and D. P. Mandic, "Quaternion-valued nonlinear adaptive filtering," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1193–1206, Aug. 2011.

[29] M. M. Ramon, N. Xu, and C. G. Christodoulou, "Beamforming using support vector machines," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, no. 1, pp. 439–442, Dec. 2005.

[30] C. C. Gaudes, I. Santamaria, J. Via, E. M. Gómez, and T. S. Paules, "Robust array beamforming with sidelobe control using support vector machines," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 574–584, Feb. 2007.

[31] P. Bouboulis, S. Theodoridis, and M. Mavroforakis, "The augmented complex kernel LMS," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4962–4967, Sep. 2012.

[32] T. Adali and H. Li, "Complex-valued adaptive signal processing," in *Adaptive Signal Processing: Next Generation Solutions*, T. Adali and S. Haykin, Eds. New York, NY, USA: Wiley, 2010, pp. 1–74.

[33] M. Novey and T. Adali, "On extending the complex fastICA algorithm to noncircular sources," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 2148–2154, May 2008.

[34] D. Mandic and V. Goh, *Complex Valued Nonlinear Adaptive Filters*. New York, NY, USA: Wiley, 2009.

[35] A. Kuh and D. Mandic, "Applications of complex augmented kernels to wind profile prediction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 3581–3584.

[36] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, May 1950.

[37] P. Bouboulis and M. Mavroforakis, "Reproducing kernel Hilbert spaces and fractal interpolation," *J. Comput. Appl. Math.*, vol. 235, no. 12, pp. 3425–3434, Apr. 2011.

[38] V. I. Paulsen. (2009, Sep.). *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces* [Online]. Available: http://www.math.uh.edu/~vern/rkhs.pdf

[39] W. Wirtinger, "Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen," *Math. Ann.*, vol. 97, no. 1, pp. 357–375, 1927.

[40] H. Li, "Complex-valued adaptive signal processing using Wirtinger calculus and its application to independent component analysis," Ph.D. dissertation, Dept. Comput. Sci. Electr. Eng., Univ. Maryland Baltimore County, Baltimore, MD, USA, 2008.

[41] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, Aug. 1995.

[42] P. Bouboulis, K. Slavakis, and S. Theodoridis, "Adaptive learning in complex reproducing kernel Hilbert spaces employing Wirtinger's subgradients," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 425–438, Mar. 2012.

[43] C. C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4427–4431, Aug. 2010.

[44] P. Chevalier and F. Pipon, "New insights into optimal widely linear array receivers for the demodulation of BPSK, MSK, and GMSK signals corrupted by noncircular interferences—Application to SAIC," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 870–883, Mar. 2006.

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

[46] B. Picinbono, "On circularity," *IEEE Trans. Signal Process.*, vol. 42, no. 12, pp. 3473–3482, Dec. 1994.

[47] B. Picinbono and P. Bondon, "Second-order statistics of complex signals," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 411–420, Feb. 1997.

[48] J. Platt. (1998, Apr.). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines* [Online]. Available: http://ftp.research.microsoft.com/pub/tr/tr-98-14.doc

[49] D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3217–3226, Nov. 2000.

[50] Y. LeCun and C. Cortes. (2014, Jul. 19). *The MNIST Database* [Online]. Available: http://yann.lecun.com/exdb/mnist/

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[52] D. Decoste and B. Schölkopf, "Training invariant support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 161–190, 2002.

Pantelis Bouboulis (M'10) received the M.Sc. and Ph.D. degrees in informatics and telecommunications from the National and Kapodistrian University of Athens, Greece, in 2002 and 2006, respectively.

He served as an Assistant Professor with the Department of Informatics and Telecommunications, University of Athens, Athens, from 2007 to 2008. He currently teaches mathematics with the Zanneio Model Experimental Lyceum of Pireas, Pireas, Greece, and is a Research Assistant with the Signal and Image Processing Laboratory, Department of Informatics and Telecommunications, University of Athens. His current research interests include machine learning, fractals, wavelets, and image processing.

Dr. Bouboulis serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2012.
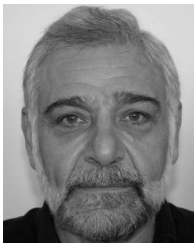
Sergios Theodoridis (F'08) was the Excellence Chair with the Department of Signal Processing and Communications, University Carlos III, Madrid, Spain, in 2011, and an Otto Monstead Guest Professor with the Technical University of Denmark, Kongens Lyngby, Denmark, in 2012. He is currently a Professor of Signal Processing and Machine Learning with the Department of Informatics and Telecommunications, University of Athens, Athens, Greece. He is the Co-Editor of the book *Efficient Algorithms for Signal Processing and System Identification* (Prentice Hall, 1993), and has co-authored the book *Pattern Recognition–4th Edition* (Academic Press, 2008), the book *Introduction to Pattern Recognition: A MATLAB Approach* (Academic Press, 2009), and three books in Greek, two of them for the Greek Open University. His current research interests include adaptive algorithms, distributed and sparsity-aware learning, machine learning and pattern recognition, and signal processing for audio processing and retrieval.

Prof. Theodoridis is a fellow of the Institution of Engineering and Technology, a Corresponding Fellow of the Royal Society of Edinburgh, and a fellow of the European Association for Signal Processing (EURASIP). He serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2015 to 2018. He is the Editor-in-Chief of the *Signal Processing Book Series* (Academic Press), and the Co-Editor-in-Chief (with Rama Chellapa) of the *EReference Signal Processing* (Elsevier). He currently serves as a Distinguished Lecturer of the IEEE Circuits and Systems Society. He has served as an IEEE Signal Processing Society Distinguished Lecturer. He has served as the President of EURASIP, a member of the Board of Governors of the IEEE CAS Society, and a member of the Board of Governors (Member-at-Large) of the IEEE SP Society. He has served as a member of the Greek National Council for Research and Technology, and was the Chairman of the SP Advisory Committee for the Edinburgh Research Partnership. He has served as the Vice Chairman of the Greek Pedagogical Institute, and he was a member of the Board of Directors of COSMOTE (the Greek mobile phone operating company) for four years. He was the General Chairman of the 1998 European Signal Processing Conference, the Technical Program Co-Chair of the 2006 and 2013 IEEE International Symposium on Circuits and Systems, the Co-Chairman and Co-Founder of the 2008 Workshop on Cognitive Information Processing (CIP), the Co-Chairman of CIP-2010, and the Technical Program Co-Chair of the 2014 International Symposium on Communications, Control and Signal Processing. He has co-authored six papers that have received the Best Paper Awards, including the 2009 IEEE Computational Intelligence Society TRANSACTIONS ON NEURAL NETWORKS Outstanding paper Award. He was a recipient of the EURASIP 2014 Meritorious Service Award.

Charalampos Mavroforakis received the B.Sc. degree in computer science from the Athens University of Economics and Business, Athens, Greece, in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Boston University, Boston, MA, USA.

His current research interests include data mining, (social) network analysis, and recommendation systems.

Leoni Evaggelatou-Dalla received the B.Sc. degree from the Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece, the M.Sc degree from the Birkbeck College of London University, London, U.K., and the Ph.D. degree from University College London (UCL), London.

She currently serves as a Professor with the Department of Mathematics, National and Kapodistrian University of Athens, where she has taught several undergraduate and post-graduate courses. For several years, she served as a Visiting Professor with London University, London, and Cyprus University, Nicosia, Cyprus. Her current research interests include convex geometry, fractals, and mathematical analysis.