The Augmented Complex Gaussian Kernel LMS Algorithm

P. Bouboulis[1] S. Theodoridis[1] Ch Mavroforakis[2]

[1]Department of Informatics and Telecommunications
University of Athens
Athens, Greece
[2]Department of Computer Science
Data Management Lab,
Boston University
Boston, MA 02215, USA

29-09-2011

Outline

Outline

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions *f* defined on a set $\mathcal{X}$ (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function (kernel) $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions f defined on a set $\mathcal{X}$ (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function (kernel) $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

1. For every $x \in \mathcal{X}$, $\kappa(x, \cdot)$ belongs to $\mathcal{H}$.

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions f defined on a set $\mathcal{X}$ (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function (kernel) $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

1. For every $x \in \mathcal{X}$, $\kappa(x, \cdot)$ belongs to $\mathcal{H}$.
2. κ has the so called reproducing property, i.e.,

$$f(x) = \langle f, \kappa(x, \cdot) \rangle_{\mathcal{H}}, \text{ for all } f \in \mathcal{H},\ x \in \mathcal{X}. \qquad (1)$$

Reproducing Kernel Hilbert Spaces.

Consider a linear class $\mathcal{H}$ of real (complex) valued functions f defined on a set $\mathcal{X}$ (in particular $\mathcal{H}$ is a Hilbert space), for which there exists a function (kernel) $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}(\mathbb{C})$ with the following two properties:

1. For every $x \in \mathcal{X}$, $\kappa(x, \cdot)$ belongs to $\mathcal{H}$.
2. κ has the so called reproducing property, i.e.,

$$f(x) = \langle f, \kappa(x, \cdot) \rangle_{\mathcal{H}}, \text{ for all } f \in \mathcal{H}, \ x \in \mathcal{X}. \quad (1)$$

Then $\mathcal{H}$ is called a Reproducing Kernel Hilbert Space (RKHS) associated to the the kernel κ.

Kernel Trick

The notion of RKHS is a popular tool for treating non-linear learning tasks.

Kernel Trick

The notion of RKHS is a popular tool for treating non-linear learning tasks.

Usually this is attained by the so called “kernel trick”.

Kernel Trick

The notion of RKHS is a popular tool for treating non-linear learning tasks.

Usually this is attained by the so called “kernel trick”.

If

$\mathcal{X} \ni x \to \Phi(x) := \kappa(x, \cdot) \in \mathcal{H}$

$\mathcal{X} \ni y \to \Phi(y) := \kappa(y, \cdot) \in \mathcal{H}$,

Kernel Trick

The notion of RKHS is a popular tool for treating non-linear learning tasks.

Usually this is attained by the so called “kernel trick”.

If
$\mathcal{X} \ni x \to \Phi(x) := \kappa(x, \cdot) \in \mathcal{H}$
$\mathcal{X} \ni y \to \Phi(y) := \kappa(y, \cdot) \in \mathcal{H}$,
then the inner product in $\mathcal{H}$ is given as a function computed on $\mathcal{X}$:

$$\kappa(x, y) = \langle \kappa(x, \cdot), \kappa(y, \cdot) \rangle_{\mathcal{H}}$$ kernel trick

Developing Learning Algorithms in RKHS

- The black box approach.

Developing Learning Algorithms in RKHS

- The black box approach.
  - Develop the learning Algorithm in $\mathcal{X}$.

Developing Learning Algorithms in RKHS

- The black box approach.
  - Develop the learning Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.

Developing Learning Algorithms in RKHS

- The black box approach.
  - Develop the learning Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.
  - Choose a kernel function κ.

Developing Learning Algorithms in RKHS

- The black box approach.
  - Develop the learning Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.
  - Choose a kernel function κ.
  - Replace inner products with kernel evaluations according to the kernel trick.

Developing Learning Algorithms in RKHS

- The black box approach.
  - Develop the learning Algorithm in $\mathcal{X}$.
  - Express it, **if possible**, in inner products.
  - Choose a kernel function κ.
  - Replace inner products with kernel evaluations according to the kernel trick.
- Work directly in the RKHS, assuming that the data have been mapped and live in the RKHS $\mathcal{H}$, i.e.,

$$\mathcal{X} \ni \boldsymbol{x} \to \Phi(\boldsymbol{x}) := \kappa(\boldsymbol{x}, \cdot) \in \mathcal{H}.$$

Advantages

Advantages of kernel-based learning tasks:

Advantages

Advantages of kernel-based learning tasks:

- The original nonlinear task is transformed into a linear one.

Advantages

Advantages of kernel-based learning tasks:

- The original nonlinear task is transformed into a linear one.
- Different types of nonlinearities can be treated in a unified way.

Outline

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based learning techniques were designed to process real data only.

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based learning techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based learning techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.
- Recently, however, a unified kernel-based framework, which is able to treat complex data, has been presented.

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based learning techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.
- Recently, however, a unified kernel-based framework, which is able to treat complex data, has been presented.
- This machinery transforms the input data into a complex RKHS, i.e.,

$$\boldsymbol{\Phi}(\boldsymbol{z}) = \kappa_{\mathbb{C}}(\cdot, \boldsymbol{z}).$$

Complex RKHS

- Although the theory of RKHS holds for complex spaces too, most of the kernel-based learning techniques were designed to process real data only.
- Moreover, in the related literature the complex kernel functions have been ignored.
- Recently, however, a unified kernel-based framework, which is able to treat complex data, has been presented.
- This machinery transforms the input data into a complex RKHS, i.e.,

$$\mathbf{\Phi}(\boldsymbol{z}) = \kappa_{\mathbb{C}}(\cdot, \boldsymbol{z}).$$

  and employs the Wirtinger's Calculus to derive the respective gradients.

Complex Kernels:

Complex Kernels:

- The complex Gaussian kernel:
$$\kappa(\boldsymbol{z}, \boldsymbol{w}) = \exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right),$$

Complex Kernels:

- The complex Gaussian kernel:
$\kappa(\boldsymbol{z}, \boldsymbol{w}) = \exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right)$,
- The Szego kernel: $\kappa(\boldsymbol{z}, \boldsymbol{w}) = \frac{1}{1-\boldsymbol{w}^H\boldsymbol{z}}$,

Complex Kernels:

- The complex Gaussian kernel: $\kappa(\boldsymbol{z}, \boldsymbol{w}) = \exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right)$,
- The Szego kernel: $\kappa(\boldsymbol{z}, \boldsymbol{w}) = \frac{1}{1-\boldsymbol{w}^H\boldsymbol{z}}$,
- Bergman kernel: $\kappa(\boldsymbol{z}, \boldsymbol{w}) = \frac{1}{(1-\boldsymbol{w}^H\boldsymbol{z})^2}$.

Wirtinger Calculus

- Complex differentiability is a very strict notion.

Wirtinger Calculus

- Complex differentiability is a very strict notion.
- In learning tasks that involve complex data, we often encounter functions (e.g., the cost functions, which are defined in $\mathbb{R}$) that ARE NOT complex differentiable.

Wirtinger Calculus

- Complex differentiability is a very strict notion.
- In learning tasks that involve complex data, we often encounter functions (e.g., the cost functions, which are defined in $\mathbb{R}$) that ARE NOT complex differentiable.
- Example: $f(z) = |z|^2 = zz^*$.

Wirtinger Calculus

- Complex differentiability is a very strict notion.
- In learning tasks that involve complex data, we often encounter functions (e.g., the cost functions, which are defined in $\mathbb{R}$) that ARE NOT complex differentiable.
- Example: $f(z) = |z|^2 = zz^*$.
- In these cases one has to express the cost function in terms of its real part f_r and its imaginary part f_i, and use real derivation with respect to f_r, f_i.

Wirtinger's Calculus

- This approach leads usually to cumbersome and tedious calculations.

Wirtinger's Calculus

- This approach leads usually to cumbersome and tedious calculations.
- Wirtinger's Calculus provides an alternative equivalent formulation.

Wirtinger's Calculus

- This approach leads usually to cumbersome and tedious calculations.
- Wirtinger's Calculus provides an alternative equivalent formulation.
- It is based on simple rules and principles.

Wirtinger's Calculus

- This approach leads usually to cumbersome and tedious calculations.
- Wirtinger's Calculus provides an alternative equivalent formulation.
- It is based on simple rules and principles.
- These rules bear a great resemblance to the rules of the standard complex derivative.

Outline

The primal problem

Suppose we are given training data of the form $\{(\boldsymbol{x}_n, d_n);\ n = 1, \dots, N\} \subset \mathcal{X} \times \mathbb{R}$, where $\mathcal{X} = \mathbb{R}^\nu$ denotes the space of input patterns.

The primal problem

Suppose we are given training data of the form $\{(\boldsymbol{x}_n, d_n);\ n = 1, \dots, N\} \subset \mathcal{X} \times \mathbb{R}$, where $\mathcal{X} = \mathbb{R}^\nu$ denotes the space of input patterns.
For example:

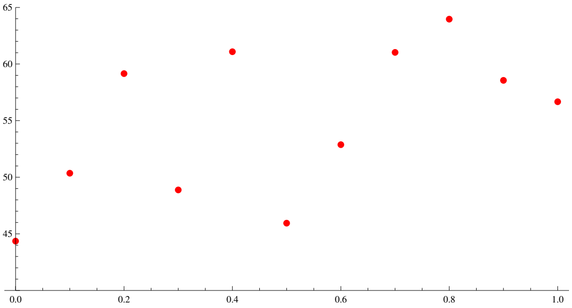


Figure: A set of 11 training points.

The primal problem

We search for a linear function $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \mathrm{b}$, $\boldsymbol{w} \in \mathcal{X}$, $b \in \mathbb{R}$, such that it approximates as close as possible the given data, according to the following minimization problem:

The primal problem

We search for a linear function $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$, $\boldsymbol{w} \in \mathcal{X}$, $b \in \mathbb{R}$, such that it approximates as close as possible the given data, according to the following minimization problem:

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{N}\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n)$$

$$\text{subject to} \quad \begin{cases} \boldsymbol{w}^T\boldsymbol{x}_n + b - d_n & \leq & \epsilon + \xi_n \\ d_n - \boldsymbol{w}^T\boldsymbol{x}_n - b & \leq & \epsilon + \hat{\xi}_n \\ \xi_n, \hat{\xi}_n & \geq & 0 \end{cases}$$

The primal problem

We search for a linear function $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \text{b}$, $\boldsymbol{w} \in \mathcal{X}$, $b \in \mathbb{R}$, such that it approximates as close as possible the given data, according to the following minimization problem:

$$\begin{aligned} \underset{\boldsymbol{w}, b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{N}\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n) \\ \text{subject to} \quad & \begin{cases} \boldsymbol{w}^T\boldsymbol{x}_n + b - d_n & \leq \quad \epsilon + \xi_n \\ d_n - \boldsymbol{w}^T\boldsymbol{x}_n - b & \leq \quad \epsilon + \hat{\xi}_n \\ \xi_n, \hat{\xi}_n & \geq \quad 0 \end{cases} \end{aligned}$$

Note that C and ϵ are chosen a priori.

Physical justification

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{N}\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n)$$

$$\text{subject to} \quad \begin{cases} \boldsymbol{w}^T\boldsymbol{x}_n + b - d_n & \leq & \epsilon + \xi_n \\ d_n - \boldsymbol{w}^T\boldsymbol{x}_n - b & \leq & \epsilon + \hat{\xi}_n \\ \xi_n, \hat{\xi}_n & \geq & 0 \end{cases}$$

Physical justification

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{N}\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n)$$

$$\text{subject to} \quad \begin{cases} \boldsymbol{w}^T\boldsymbol{x}_n + b - d_n & \leq & \epsilon + \xi_n \\ d_n - \boldsymbol{w}^T\boldsymbol{x}_n - b & \leq & \epsilon + \hat{\xi}_n \\ \xi_n, \hat{\xi}_n & \geq & 0 \end{cases}$$

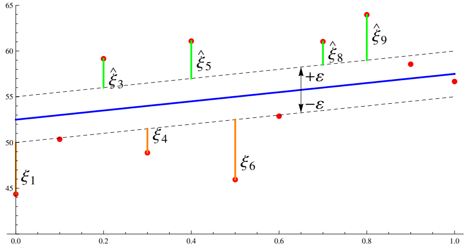


Figure: Linear SVR regression

The dual problem

To solve this task, usually we consider the dual problem derived by the Lagrangian:

$$\underset{\boldsymbol{a},\hat{\boldsymbol{a}}}{\text{maximize}} \begin{cases} -\frac{1}{2}\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\boldsymbol{x}_n^T\boldsymbol{x}_m \\ -\epsilon\sum_{n=1}^{N}(\hat{a}_n + a_n) + \sum_{n=1}^{N}d_n(\hat{a}_n - a_n) \end{cases}$$

$$\text{subject to} \quad \sum_{n=1}^{N}(\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N].$$

Outline

The kernel trick

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$.

The kernel trick

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$.
- In the dual problem, replace the inner products $\boldsymbol{x}_n^T \boldsymbol{x}_m$ with the respective kernel evaluations, i.e., $\kappa_{\mathbb{R}}(\boldsymbol{x}_n, \boldsymbol{x}_m)$.

The kernel trick

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$.
- In the dual problem, replace the inner products $\boldsymbol{x}_n^T \boldsymbol{x}_m$ with the respective kernel evaluations, i.e., $\kappa_{\mathbb{R}}(\boldsymbol{x}_n, \boldsymbol{x}_m)$.
- The application of the kernel trick leads to the nonlinear SVR:

The kernel trick

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$.
- In the dual problem, replace the inner products $\boldsymbol{x}_n^T \boldsymbol{x}_m$ with the respective kernel evaluations, i.e., $\kappa_{\mathbb{R}}(\boldsymbol{x}_n, \boldsymbol{x}_m)$.
- The application of the kernel trick leads to the nonlinear SVR:

$$\underset{\boldsymbol{a},\hat{\boldsymbol{a}}}{\text{maximize}} \quad \begin{cases} -\frac{1}{2}\sum_{n,m=1}^{N}(\hat{a}_n - a_n)(\hat{a}_m - a_m)\kappa_{\mathbb{R}}(\boldsymbol{x}_n, \boldsymbol{x}_m) \\ -\epsilon\sum_{n=1}^{N}(\hat{a}_n + a_n) + \sum_{n=1}^{N} d_n(\hat{a}_n - a_n) \end{cases}$$

$$\text{subject to} \quad \sum_{n=1}^{N}(\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N].$$

Mapping to the feature space

The application of the kernel trick to the dual problem is equivalent to the following procedure:

Mapping to the feature space

The application of the kernel trick to the dual problem is equivalent to the following procedure:

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$, that is associated to a specific RKHS $\mathcal{H}$.

Mapping to the feature space

The application of the kernel trick to the dual problem is equivalent to the following procedure:

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$, that is associated to a specific RKHS $\mathcal{H}$.
- Map the points $\boldsymbol{x}_n$ to $\Phi(\boldsymbol{x}_n) \in \mathcal{H}$, $n = 1, \ldots, N$.

Mapping to the feature space

The application of the kernel trick to the dual problem is equivalent to the following procedure:

- Choose a positive definite kernel $\kappa_{\mathbb{R}}$, that is associated to a specific RKHS $\mathcal{H}$.
- Map the points $\boldsymbol{x}_n$ to $\Phi(\boldsymbol{x}_n) \in \mathcal{H}$, $n = 1, \ldots, N$.
- Solve the linear SVR task on the infinite dimensional RKHS $\mathcal{H}$, for the training data $\{(\Phi(\boldsymbol{x}_n), d_n);\ n = 1, \ldots, N\}$.

A toy example

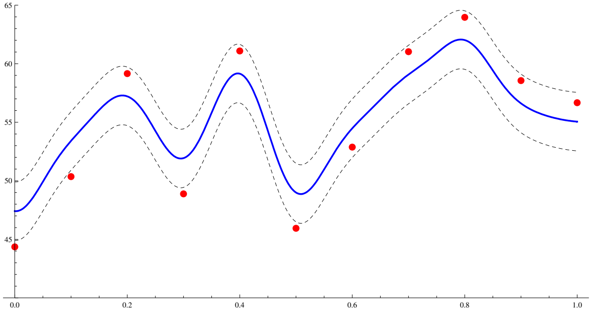


Figure: Non-linear SVR regression, $\epsilon = 2.5$, $C = 1000$.

A toy example

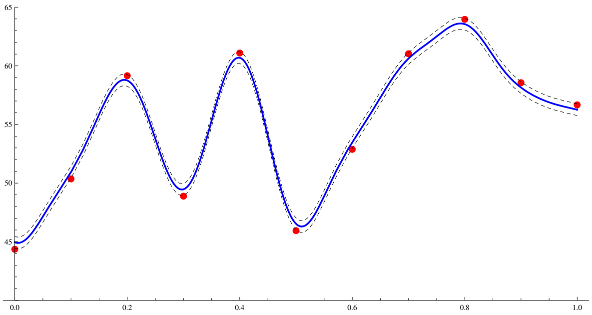


Figure: Non-linear SVR regression, $\epsilon = 0.5$, $C = 1000$.

A toy example

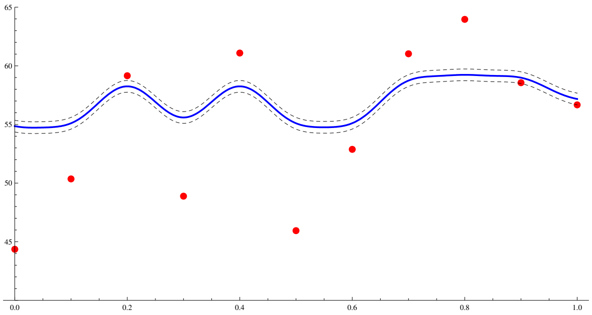


Figure: Non-linear SVR regression, $\epsilon = 0.5$, $C = 20$.

Outline

The problem

Suppose we are given training data of the form $\{(\boldsymbol{z}_n, d_n);\ n = 1, \ldots, N\} \subset \mathcal{X} \times \mathbb{C}$, where $\mathcal{X} = \mathbb{C}^{\nu}$ denotes the space of input patterns.

The problem

Suppose we are given training data of the form $\{(\boldsymbol{z}_n, d_n);\ n = 1, \ldots, N\} \subset \mathcal{X} \times \mathbb{C}$, where $\mathcal{X} = \mathbb{C}^\nu$ denotes the space of input patterns.

As $\boldsymbol{z}_n$ is complex, we denote by $\boldsymbol{x}_n$ its real part and by $\boldsymbol{y}_n$ its imaginary part respectively, i.e.,

$$\boldsymbol{z}_n = \boldsymbol{x}_n + i\boldsymbol{y}_n, \quad n = 1, \ldots, N.$$

The problem

Suppose we are given training data of the form $\{(\boldsymbol{z}_n, d_n);\ n = 1, \ldots, N\} \subset \mathcal{X} \times \mathbb{C}$, where $\mathcal{X} = \mathbb{C}^\nu$ denotes the space of input patterns.

As $\boldsymbol{z}_n$ is complex, we denote by $\boldsymbol{x}_n$ its real part and by $\boldsymbol{y}_n$ its imaginary part respectively, i.e.,

$$\boldsymbol{z}_n = \boldsymbol{x}_n + i\boldsymbol{y}_n, \quad n = 1, \ldots, N.$$

Our objective is to develop an SVR rationale for the complex training data.

Dual Channel approach

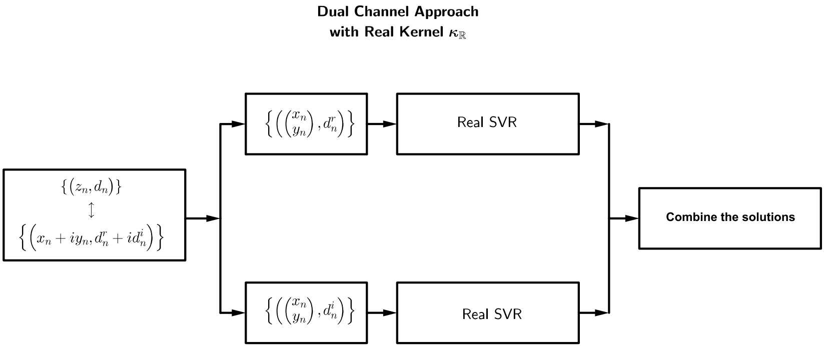
A straightforward approach is to consider two different problems in the real domain:

Dual Channel approach

A straightforward approach is to consider two different problems in the real domain:

This is usually referred to as the Dual Channel Approach.

Dual Channel approach



Pure Complex Kernels

To take into account the complex structure of the data, a different method is needed.

Pure Complex Kernels

To take into account the complex structure of the data, a different method is needed.

In this work we consider a SVR rationale based on complex RKHS.

Outline

Widely Linear Estimation

- We map the data to the complex RKHS, $\mathbb{H}$, via the feature map Φ, i.e. $\boldsymbol{z} \rightarrow \Phi(\boldsymbol{z})$.

Widely Linear Estimation

- We map the data to the complex RKHS, $\mathbb{H}$, via the feature map Φ, i.e. $\boldsymbol{z} \to \Phi(\boldsymbol{z})$.
- There are two ways for estimating the output:

Widely Linear Estimation

- We map the data to the complex RKHS, $\mathbb{H}$, via the feature map Φ, i.e. $\boldsymbol{z} \to \Phi(\boldsymbol{z})$.
- There are two ways for estimating the output:
  - The complex linear estimation function:

  $$T(\Phi(\boldsymbol{z})) = \langle \Phi(\boldsymbol{z}), u \rangle_{\mathbb{H}} + c,$$

  which has been found to perform well for circular input data,

Widely Linear Estimation

- We map the data to the complex RKHS, $\mathbb{H}$, via the feature map Φ, i.e. $\boldsymbol{z} \to \Phi(\boldsymbol{z})$.
- There are two ways for estimating the output:
  - The complex linear estimation function:

$$T(\Phi(\boldsymbol{z})) = \langle \Phi(\boldsymbol{z}), u \rangle_{\mathbb{H}} + c,$$

    which has been found to perform well for circular input data,
  - A more natural approach is the widely linear estimation function:

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

    which works well for both circular and non-circular input data.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.
- A complex random variable is$Z$ 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.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.
- A complex random variable is$Z$ 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.
- Naturally, the assumption of circularity on the input data limits the area for applications, as many practical signals exhibit non-circular characteristics.

Circularity

- Circularity is intimately related to the rotation in the geometric sense.
- A complex random variable is$Z$ 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.
- Naturally, the assumption of circularity on the input data limits the area for applications, as many practical signals exhibit non-circular characteristics.
- Widely linear filters are able to efficiently treat such signals, as they capture the full second order statistics of any given complex-valued data sequence.

Pure Complex Kernels

- Following the principles of widely linear estimation, our aim is to find a widely linear function, T, defined on the complex RKHS, i.e.,

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

such that:

Pure Complex Kernels

- Following the principles of widely linear estimation, our aim is to find a widely linear function, T, defined on the complex RKHS, i.e.,

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

  such that:
  - It is as flat as possible and

Pure Complex Kernels

- Following the principles of widely linear estimation, our aim is to find a widely linear function, T, defined on the complex RKHS, i.e.,

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

  such that:
  - It is as flat as possible and
  - It has at most ϵ deviation from both the real and the imaginary parts of the obtained values d_n.

Complex SVR primal problem

We formulate the complex support vector regression task as follows:

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

Deriving the dual problem

- As both the objective function and the constraints of the primal problem involve complex values, the Lagrangian will be defined on complex domain.

Deriving the dual problem

- As both the objective function and the constraints of the primal problem involve complex values, the Lagrangian will be defined on complex domain.
- To explore the saddle point conditions, we employ the rules of Wirtinger calculus and compute the respective Wirtinger's derivatives.

Deriving the dual problem

- As both the objective function and the constraints of the primal problem involve complex values, the Lagrangian will be defined on complex domain.
- To explore the saddle point conditions, we employ the rules of Wirtinger calculus and compute the respective Wirtinger's derivatives.
- Furthermore, we split the complex kernel to its real and imaginary parts:

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

The dual problem

It turns out that we can split the dual problem into two maximization tasks:

The dual problem

It turns out that we can split the dual problem into two maximization tasks:

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

The dual problem

It turns out that we can split the dual problem into two maximization tasks:

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

and

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

The Induced kernel

- Observe that κ_1 is the real part of the complex kernel.

The Induced kernel

- Observe that κ_1 is the real part of the complex kernel.
- It can be proved that κ_1 is a real kernel. We call this kernel the induced real kernel of κ.

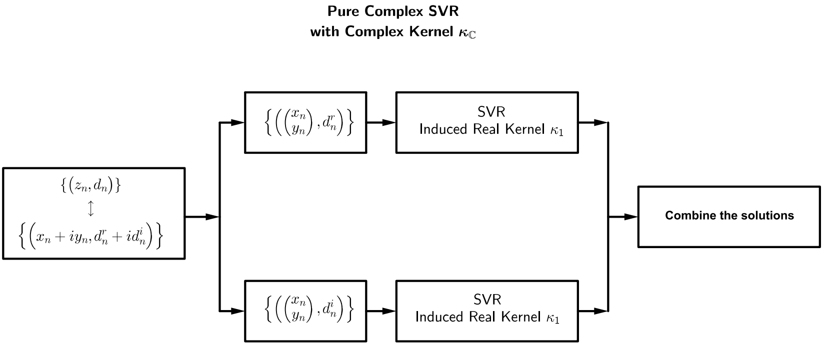
The Induced kernel

- Observe that κ_1 is the real part of the complex kernel.
- It can be proved that κ_1 is a real kernel. We call this kernel the induced real kernel of κ.
- Hence, the complex SVR task can be split into two real SVR tasks using this induced real kernel.

The Induced kernel

- Observe that κ_1 is the real part of the complex kernel.
- It can be proved that κ_1 is a real kernel. We call this kernel the induced real kernel of κ.
- Hence, the complex SVR task can be split into two real SVR tasks using this induced real kernel.
- We should emphasize that this procedure is not identical to the dual channel approach, as the exploited kernel is the one that is induced by the chosen complex kernel. This induced kernel carries the structure of the complex space.

Complex SVR scheme



The combined solution

Combining the solutions of the two real tasks we take:

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

The complex Gaussian kernel

The complex Gaussian kernel is a generalization of the Gaussian RBF for complex data, i.e.,

$$\kappa(\boldsymbol{z}, \boldsymbol{w}) = \exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right).$$

The complex Gaussian kernel

The complex Gaussian kernel is a generalization of the Gaussian RBF for complex data, i.e.,
$\kappa(\boldsymbol{z}, \boldsymbol{w}) = \exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right)$.

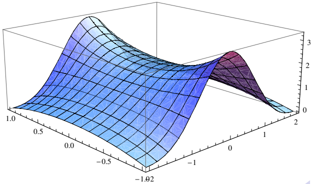
If the complex Gaussian kernel is employed the induced real kernel becomes $\kappa_1(\boldsymbol{z}, \boldsymbol{w}) = \Re\left(\exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right)\right)$.

The complex Gaussian kernel

The complex Gaussian kernel is a generalization of the Gaussian RBF for complex data, i.e.,

$$\kappa(\boldsymbol{z}, \boldsymbol{w}) = \exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right).$$

If the complex Gaussian kernel is employed the induced real kernel becomes $\kappa_1(\boldsymbol{z}, \boldsymbol{w}) = \Re\left(\exp\left(-\frac{\sum_{i=1}^{d}(z_i - w_i^*)^2}{\sigma^2}\right)\right)$.



Outline

Experiment 1

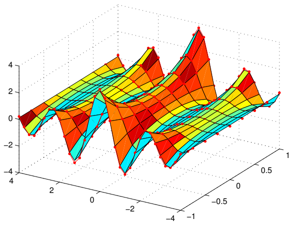
To demonstrate the performance of the proposed scheme, we perform a simple regression test on the complex sinc function:

$$\operatorname{sinc}(z) = \begin{cases} 1, & z = 0 \\ \frac{\sin(\pi z)}{\pi z}, & z \neq 0 \end{cases}.$$

Experiment 1

To demonstrate the performance of the proposed scheme, we perform a simple regression test on the complex sinc function:

$$\mathrm{sinc}(z) = \begin{cases} 1, & z = 0 \\ \frac{\sin(\pi z)}{\pi z}, & z \neq 0 \end{cases}.$$



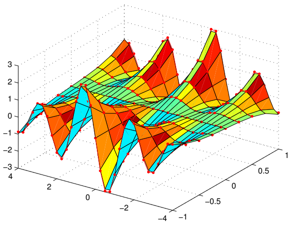


Figure: The real (Re(sinc(z))) and the imaginary (Im(sinc(z))) part of the estimated *sinc* function from the complex SVR. The parameters of the SVR task were chosen as $\sigma = 2$, $C = 1000$, $\epsilon = 0.1$.

Experiment 2

Adding noise to the points of the sinc function we take:

Experiment 2

Adding noise to the points of the sinc function we take:

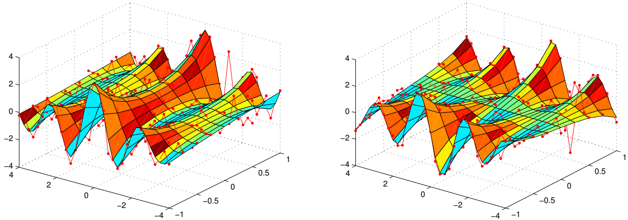
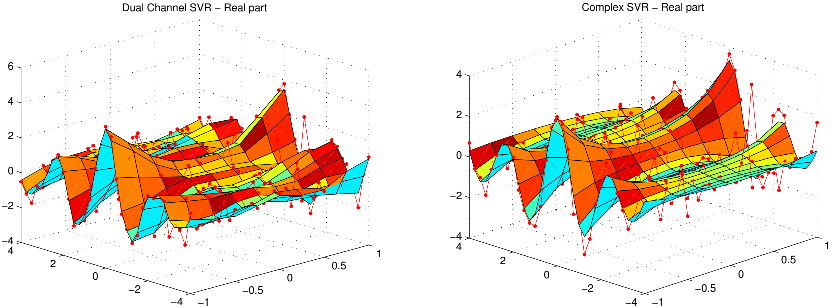


Figure: The real (Re(sinc(z))) and the imaginary (Im(sinc(z))) part of the estimated *sinc* function from the complex SV regression. The points shown in the figure are the real and imaginary parts of the noisy training data used in the simulation.

Comparison with the Dual Channel Approach



(a) (b)

Figure: The real (Re(sinc(z))) part of the estimated sinc function from (a) the dual channel SVR and (b) the complex SVR. In both algorithms we used the same values for C and ϵ.

Comparison with the Dual Channel Approach

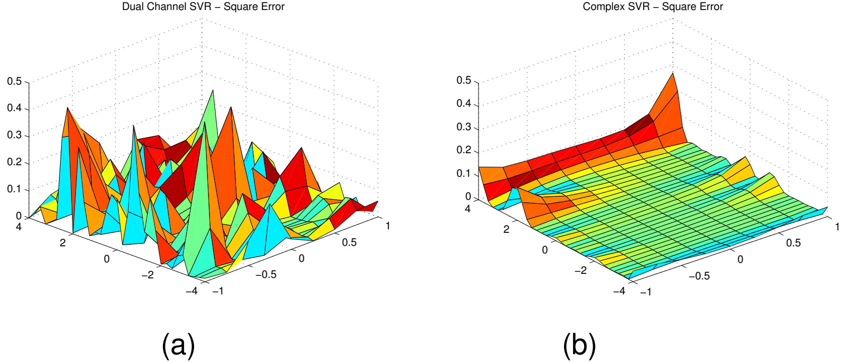


Figure: The square error between the real part of the sinc function (i.e., (Re(sinc(z)))) and the estimated sinc function from (a) the dual channel SVR and (b) the complex SVR.