# Functional Form Estimation using Oblique Projection Matrices for LS-SVM Regression Models

Alexander Caicedo[1,2*], Carolina Varon[1,2], Sabine Van Huffel[1,2], Johan A. K. Suykens[1],

**1** Department of Electrical Engineering ESAT-STADIUS Center for Dynamical Systems, Signal Processing, and Data Analytics/KU Leuven, Belgium.
**2** imec, Leuven, Belgium.

* acaicedo@esat.kuleuven.be

## Abstract

Kernel regression models have been used as non-parametric methods for fitting experimental data. However, due to their non-parametric nature, they belong to the so-called "black box" models, indicating that the relation between the input variables and the output, depending on the kernel selection, is unknown. In this paper we propose a new methodology to retrieve the relation between each input regressor variable and the output in a least squares support vector machine (LS-SVM) regression model. The method is based on oblique subspace projectors (ObSP), which allows to decouple the influence of input regressors on the output by including the undesired variables in the null space of the projection matrix. Such functional relations are represented by the nonlinear transformation of the input regressors, and their subspaces are estimated using appropriate kernel evaluations. We exploit the properties of ObSP in order to decompose the output of the obtained regression model as a sum of the partial nonlinear contributions and interaction effects of the input variables, we called this methodology Nonlinear ObSP (NObSP). We compare the performance of the proposed algorithm with the component selection and smooth operator (COSSO) for smoothing spline ANOVA models. We use as benchmark 2 toy examples and a real life regression model using the concrete strength dataset from the UCI machine learning repository. We showed that NObSP is able to outperform COSSO, producing stable estimations of the functional relations between the input regressors and the output, without the use of prior-knowledge. This methodology can be used in order to understand the functional relations between the inputs and the output in a regression model, retrieving the physical interpretation of the regression models.

## Introduction

Non-parametric regression is an important field of data analysis. These non-parametric models use some observations of the input data and the desired target to estimate a function and make predictions [1, 2]. However, generally, these models focus on the prediction of the target variable of interest and not on the model interpretability. In this manuscript, we will refer to interpretability as the property of a model to express the output in additive terms of the partial nonlinear contributions of the input variables and their interaction effects. In several applications interpretability plays an important role in the construction of prediction models. In such cases, the main goal is not the prediction of the response of a system but to determine the underlying mechanisms and

the relationship between the inputs and the output. For instance, in medical applications, this information can be used in order to identify treatment targets, support diagnosis, and facilitate the introduction of these models in clinical practice. As an example, Van Belle et al. proposed the use of a color code to enhance the interpretability of classification models for the clinicians [3].

Interpretability of black box models has already been addressed for classifiers and regression models using different strategies. On the one hand, Van Belle et al. proposed a clinical interpretable classification model based on a least squares support vector machine (LS-SVM) using the radial basis function (RBF) kernel [4]. In that work, the authors retrieved the interpretability of the classifiers by using a truncated multinomial expansion of the RBF kernel. On the other hand, for regression models, interpretability has been tackled by using sparse additive models [5], and functional ANOVA models [6, 7]. In these models the target observations are modeled as a sum of a constant term, main effects, and interaction effects. In this context, the main effects refer to the direct relation between each input variable and the output, considering both the linear and nonlinear contributions of an input regressor on the output, while the interaction effect refers to the combined effect of 2 or more input variables on the output [6]. These models retrieve interpretability similarly to the case of generalized linear models [8], where a direct link between the contribution of each input in the output is estimated explicitly. However, these models require the use of prior knowledge in terms of which variables should be included in the regression models, as well as which interaction effects are of importance. This can only be achieved if the designer has a profound knowledge of the process and mechanisms underlying the changes in the target variable. To address this problem, some methodologies have been developed in order to select components that are relevant for the regression model in an automatic way, [9–11]. In particular, in the work from Lin et al., a new method for model selection and fitting in non-parametric regression models using smoothing spline ANOVA is proposed [11]. This method is referred as the Component Selection and Smoothing Operator (COSSO). This methodology is able to produce a model that is accurate in the prediction of the target variable, and it retrieves some of the interpretability by accurately identifying the components and functional forms of the relation between the input variables and the output. Nevertheless, it requires to specify a-priori if the user is interested in finding only main effects or interaction effects as well. More importantly, the solution of both problems does not converge, since the common terms in both cases are not the same. This is due to the fact that both problems, computing the main effect or including the interaction effects, lead to different cost functions with the same objective, fit the observed output. Additionally, this method has been developed for smoothing spline ANOVA models (SS-ANOVA) and, to the best of our knowledge, it has not been extended to other kernel based regression models.

From a geometry point of view, interpretability of non-parametric regression models can be addressed as the decomposition of a target observation vector into additive components [12]. Each of these components should lie in the subspaces that are spanned by the respective input regressors. If a basis for the subspaces of each individual input regressor and their interaction effects can be found, then appropriate projection matrices can be constructed in order to retrieve the interpretability of the models. This idea has been previously exploited in the case of linear models, where we have proposed to use a Hankel expansion of the input variables to construct a basis for their subspaces, and construct oblique subspace projectors in order to effectively decouple the dynamics of undesired input variables, representing the output as a sum of the partial linear contributions of each input [13]. Oblique projectors are particularly powerful when the variables that want to be decoupled are not orthogonal [14]. They have been applied as a preprocessing step in different hyper-spectral imaging [15], decoding [16], and

biomedical applications [13].

Considering non-linear regression models using LS-SVM, due to the geometry of the RBF kernel, the subspaces spanned by the nonlinear transformation of the input regressors are not likely to be orthogonal [17]. Therefore, in order to decompose the output into interpretable additive components, oblique projectors should be used. In this paper we address this issue, by formulating a nonlinear extension to oblique subspace projections. We show how to create a basis for the subspaces spanned by the nonlinear transformation of the input regressors, main effects as well as interaction effects, using kernel evaluations. In addition, we suggest how to construct appropriate oblique projectors in order to effectively decompose the output of a kernel based regression model into interpretable additive components.

The manuscript is structured as follows, in section 1 we briefly describe an LS-SVM regression model. In section 2 we describe how to construct oblique subspace projectors and we present the proposed algorithm. In section 3 we evaluate the performance of the proposed algorithm using 2 toy examples and an application using the concrete strength dataset from the UCI machine learning repository. In section 4 we discuss the results and the potential use of the proposed algorithm, and we finalize with some concluding remarks in section 5.

Throughout this manuscript we will use the following notation: we will refer to scalars as italic letters ($x$), vectors will be represented as lowercase bold variables ($\mathbf{x}$), matrices will refer to capital bold variables ($\mathbf{\Omega}$), subspaces will be represented by calligraphic capital letters ($\mathcal{V}$), and spaces will be represented by blackboard bold capital letters ($\mathbb{R}$). With some abuse of notation, we will refer to the projector matrix onto the subspace of the $l^{\text{th}}$ input regressors, along the subspace spanned by the other regressors as $\mathbf{P}_{l/(l)}$, where the subindex $l$ represents the subspace of the regressor where the output will be projected, the subindex $(l)$ represents all the input regressors excluding the $l^{\text{th}}$, and the symbol / represents the oblique projection.

# 1 LS-SVM for nonlinear Regression

LS-SVM is a kernel based methodology that can be used to solve nonlinear classification and regression problems [18]. Due to its flexibility to manage different kind of problems and produce an adequate mathematical model, LS-SVM has been used successfully in different application fields such as: the prediction of electricity energy consumption [19], estimation of water pollution [20], forecasting of carbon price [21], and the prediction of meteorological time series [22], among others. However, its applications have been hampered due to its black-box model nature.

Let's consider the following nonlinear regression model:

$$\hat{y}(\mathbf{x}) = \boldsymbol{w}^T \varphi(\mathbf{x}) + b, \tag{1}$$

where $\hat{y} \in \mathbb{R}$ represents the output of the model, $\mathbf{x} \in \mathbb{R}^d$, and $\mathbf{x} = [x^{(1)}; \ldots; x^{(d)}]$, represents the input vector, $x^{(l)}$ is the entry for the $l^{\text{th}}$ input regressor, and $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^p$ represents the nonlinear mapping of $\mathbf{x}$ into a high dimensional feature space. Given the following training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$, the LS-SVM regression problem can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{w},e,b} J(\boldsymbol{w}, e) = \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2 \\ \text{s.t.} \quad & y_i = \boldsymbol{w}^T\varphi(\mathbf{x}_i) + b + e_i, \quad i = 1, \ldots, N, \end{aligned} \tag{2}$$

where $\gamma$ represents the regularization constant. By taking the Lagrangian and solving for the Karush-Kuhn-Tucker conditions for optimality, the solution is given by:

$$\left[\begin{array}{c|c} 0 & \mathbf{1_N}^T \\ \hline \mathbf{1_N} & \boldsymbol{\Omega} + \frac{1}{\gamma}\mathbf{I} \end{array}\right]\left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{y} \end{array}\right] \tag{3}$$

where $\mathbf{y} = [y_1; \ldots; y_N]$, $\mathbf{1_N} = [1; \ldots; 1]$, $\boldsymbol{\Omega}_{ij} = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$ is the $ij-$th entry of the kernel matrix $\boldsymbol{\Omega}$, and $\boldsymbol{\alpha} = [\alpha_1; \ldots; \alpha_N]$ are the Lagrange multipliers, and $b$ is the bias term.

The resulting regression model becomes:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b. \tag{4}$$

In matrix form for the training, the optimization problem in Eq (2) can be formulated as follows:

$$\begin{array}{rl} \min\limits_{\boldsymbol{w},e,b} J(\boldsymbol{w}, e) = & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}||\mathbf{e}||_2^2 \\ \text{s.t.} & \mathbf{y} = \boldsymbol{\Phi}\boldsymbol{w} + b + \mathbf{e}, \end{array} \tag{5}$$

where $\mathbf{e} = [e_1; \ldots; e_N]$ is the error vector, and $\boldsymbol{\Phi} = [\varphi^T(\mathbf{x}_1); \ldots; \varphi^T(\mathbf{x}_N)]$ is the matrix containing the nonlinear transformations of the input vectors $\mathbf{x}$. From this definition, notice that the kernel matrix can be obtained as $\boldsymbol{\Omega} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$.

The matrix form of the solution, for the training points, is then given by:

$$\hat{\mathbf{y}} = \boldsymbol{\Omega}\boldsymbol{\alpha} + b, \tag{6}$$

with $\hat{\mathbf{y}} = [\hat{y}_1; \ldots; \hat{y}_N]$, and $\boldsymbol{\Omega}_{ij} = \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j)$.

## 1.1 Removing the mean value in LS-SVM regression

Since the proposed algorithm makes use of projection matrices, it is important that the data that is projected is centered. This means, that the bias term in the regression model should be eliminated, as well as the mean value of the nonlinear transformation of the input regressors should be removed. These modifications lead to the following problem formulation [23]:

$$\begin{array}{rl} \min\limits_{\boldsymbol{w},e,b} J(\boldsymbol{w}, e) = & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum\limits_{i=1}^{N} e_i^2 \\ \text{s.t.} & y_i = \boldsymbol{w}^T\big(\varphi(\mathbf{x}_i) - \boldsymbol{\mu}_\varphi\big) + e_i, \quad i = 1, \ldots, N, \end{array} \tag{7}$$

where $\boldsymbol{\mu}_\varphi = \frac{1}{N}\sum_{i=1}^{N}\varphi(\mathbf{x}_i)$. By taking the Lagrangian and solving for the Karush-Kuhn-Tucker conditions for optimality, the solution is given by:

$$\left[\begin{array}{c} \boldsymbol{\Omega}_C + \frac{1}{\gamma}\mathbf{I} \end{array}\right]\left[\begin{array}{c} \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} \mathbf{y} \end{array}\right] \tag{8}$$

where $\boldsymbol{\Omega}_C = \mathbf{M}\boldsymbol{\Omega}\mathbf{M}$, is the centered kernel matrix, and $\mathbf{M} = \mathbf{I} - \mathbf{1}_N\mathbf{1}_N^T/N$ is a centering matrix. The $ij-$th entry of the centered kernel matrix is given by $\boldsymbol{\Omega}_{ij} = \big(\varphi(\mathbf{x}_i) - \mu_\varphi\big)^T\big(\varphi(\mathbf{x}_j) - \mu_\varphi\big)$.

The solution for the training set, in matrix form, is expressed as:

$$\hat{\mathbf{y}} = \boldsymbol{\Omega}_C\boldsymbol{\alpha}. \tag{9}$$

# 2 Nonlinear regression decomposition using ObSP

## 2.1 Oblique subspace Projections

Oblique subspace projection (ObSP) is a generalization of orthogonal projectors, where a given vector is projected onto a target subspace following the direction of a reference subspace [24]. Oblique subspace projectors can be used in order to decompose a signal into the partial contributions of some regressors, even when the signal subspaces of the different input regressors are not orthogonal [13, 16]. An algorithm for the use of ObSP in signal decomposition for linear regression models has been proposed in [13]. Briefly, the algorithm is summarized as follows. Let's define $\mathcal{V} \subset \mathbb{R}^N$ as the subspace spanned by a matrix $\mathbf{A} = [\mathbf{A}_l \quad \mathbf{A}_{(l)}]$, with $\mathbf{A} \in \mathbb{R}^{N\times p}$, $\mathbf{A}_l \in \mathbb{R}^{N\times q}$ the partition of $\mathbf{A}$ that spans the subspace $\mathcal{V}_l \subset \mathcal{V}$, and $\mathbf{A}_{(l)} \in \mathbb{R}^{N\times(p-q)}$ the partition of $\mathbf{A}$ that spans the subspace $\mathcal{V}_{(l)} \subset \mathcal{V}$, such that $\mathcal{V} = \mathcal{V}_l \oplus \mathcal{V}_{(l)}$. Now, let's consider $\mathcal{V} = \mathcal{V}_1 \oplus \mathcal{V}_2 \oplus ... \oplus \mathcal{V}_d$, with $\oplus$ being the direct sum operator, and $d$ represents the number of signal subspaces embedded in $\mathbf{A}$ satisfying $d \leq p$; then the oblique projector onto $\mathcal{V}_l$ along $\mathcal{V}_{(l)} = \mathcal{V}_1 \oplus ... \oplus \mathcal{V}_{l-1} \oplus \mathcal{V}_{l+1} ... \oplus \mathcal{V}_d$, denoted by $P_{l/(l)}$, is given by:

$$\mathbf{P}_{l/(l)} = \mathbf{A}_l(\mathbf{A}_l^T\mathbf{Q}_{(l)}\mathbf{A}_l)^\dagger\mathbf{A}_l^T\mathbf{Q}_{(l)}, \tag{10}$$

where $\dagger$ represents the generalized inverse, and $\mathbf{Q}_{(l)}$ is the orthogonal projector onto $\text{Null}(\mathbf{A}_{(l)}^T) \subset \mathcal{V}_{(l)}^\perp$, which is computed as:

$$\mathbf{Q}_{(l)} = \mathbf{I}_N - \mathbf{P}_{(l)}, \tag{11}$$

where $\mathbf{P}_{(l)} = \mathbf{A}_{(l)}(\mathbf{A}_{(l)}^T\mathbf{A}_{(l)})^\dagger\mathbf{A}_{(l)}^T$ is the orthogonal projector onto $\mathcal{V}_{(l)} \equiv \text{Span}(\mathbf{A}_{(l)})$ [24].

## 2.2 Nonlinear Oblique Subspace Projection (NObSP)

Let's consider the following nonlinear regression problem:

$$y(\mathbf{x}) = \sum_{l=1}^{d} f_l(x^{(l)}) + \sum_{l=1}^{d}\sum_{h>l}^{d} f_{lh}(x^{(l)}, x^{(h)}) + G(\mathbf{x}) + e, \tag{12}$$

where $e$ is the error term, $f_l(x^{(l)})$ represents the partial contribution, linear and non-linear, of the $l^{\text{th}}$ input variable $x^{(l)}$ on the output, $f_{lh}(x^{(l)}, x^{(h)})$ represents the partial contribution of the interaction between the variables $x^{(l)}$ and $x^{(h)}$ on the output, the term $G(\mathbf{x})$ represents the partial contribution of all the other higher order interactions between the input variables, with $\mathbf{x} = [x^{(1)}; \ldots; x^{(d)}]$, and $d$ the number of input variables.

From Eq (12), it can be seen that the partial contributions of each input variable, their second order interactions, and the higher order interactions can be found if an appropriate projection matrix can be created. Such projection matrix will span the subspace of the nonlinear transformation of the (input signal)/(interaction effects) of interest, whilst its null subspace contains the nonlinear transformation of the other remaining input variables and their interactions.

In the case of functional ANOVA models, it is assumed that these subspaces are orthogonal [6, 11]. However, there is no evidence to support this claim, especially when using kernel regression models. In such cases, an oblique projector will be more appropriate. The problem now lies in finding a basis for the signals subspaces that are required to construct the oblique projector operator.

In a kernel regression framework, the output of the regression model can be expressed as a linear combination of kernel evaluations. Therefore, the kernel matrix

spans the column space of the nonlinear transformation of the input variables. This can be seen from Eq (6), where the output can be expressed as $\hat{\mathbf{y}} = \mathbf{P}\mathbf{y}$, where $\mathbf{P}$ is the hat, or projection, matrix computed as $\mathbf{P} = \boldsymbol{\Phi}(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T = \boldsymbol{\Omega}(\boldsymbol{\Omega}^T\boldsymbol{\Omega})^{-1}\boldsymbol{\Omega}^T$. In addition, from the solution of the centered regression problem in Eq (9), we obtained that the output is given by $\hat{\mathbf{y}} = \mathbf{P}_{\mathbf{C}}\mathbf{y}$ where $\mathbf{P}_{\mathbf{C}}$ is the hat matrix computed as $\mathbf{P}_{\mathbf{C}} = \boldsymbol{\Phi}_{\boldsymbol{C}}({\boldsymbol{\Phi}_{\boldsymbol{C}}}^T\boldsymbol{\Phi}_{\boldsymbol{C}})^{-1}{\boldsymbol{\Phi}_{\boldsymbol{C}}}^T$, which leads to $\mathbf{P}_{\mathbf{C}} = \boldsymbol{\Omega}_{\boldsymbol{C}}({\boldsymbol{\Omega}_{\boldsymbol{C}}}^T\boldsymbol{\Omega}_{\boldsymbol{C}})^{-1}{\boldsymbol{\Omega}_{\boldsymbol{C}}}^T$, with $\boldsymbol{\Phi}_C = [\varphi^T(\mathbf{x}_1) - \mu_\varphi; \ldots; \varphi^T(\mathbf{x}_N) - \mu_\varphi]$. Taking this into account, if we consider the matrix $\boldsymbol{\Phi}_{\boldsymbol{C}l}$ representing the centered nonlinear transformation of the $l^{\text{th}}$ input regressor and defined as $\boldsymbol{\Phi}_{\boldsymbol{C}l} = [\varphi(0, \ldots, x_1^{(l)}, \ldots, 0)^T - \mu_\varphi; \ldots; \varphi(0; \ldots, x_N^{(l)}, \ldots, 0)^T - \mu_\varphi]$, then, the projection matrix onto its subspace is given by $\mathbf{P}_{\mathbf{C}i} = \boldsymbol{\Phi}_{\boldsymbol{C}l}({\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T\boldsymbol{\Phi}_{\boldsymbol{C}l})^{-1}{\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T$, since the nonlinear transformation is not known, we cannot directly compute $\mathbf{P}_{\mathbf{C}l}$ but we propose to use kernel evaluations as follows:

**Proposition 1.** *Let $\mathbf{y} = \boldsymbol{\Phi}_C\boldsymbol{w} + \mathbf{e}$, where $\boldsymbol{\Phi}_{\boldsymbol{C}} = [\varphi(\mathbf{x}_1)^T - \mu_\varphi; \ldots; \varphi(\mathbf{x}_N)^T - \mu_\varphi]$ is a matrix containing the centered nonlinear transformation of the regressor variables $\mathbf{x}_i = [x_i^{(1)}; \ldots; x_i^{(d)}]$. In a kernel framework, where the solution to this problem is given by $\hat{\mathbf{y}} = \boldsymbol{\Omega}_{\boldsymbol{C}}\boldsymbol{\alpha}$, with $\boldsymbol{\Omega}(i,j) = K(\mathbf{x}_i, \mathbf{x}_j)$, $K(\cdot,\cdot)$ the kernel function, and $\boldsymbol{\Omega}_{\boldsymbol{C}} = \mathbf{M}\boldsymbol{\Omega}\mathbf{M}$, and $\mathbf{M} = \mathbf{I} - \mathbf{1}_N\mathbf{1}_N^T/N$. Then, the kernel matrix $\boldsymbol{\Omega}_{\boldsymbol{C}l}$, formed using $K(\mathbf{x}^{(l)}, \mathbf{x})$, with $\mathbf{x}^{(l)} = [0; \ldots; x^{(l)}; \ldots; 0]$ a vector containing only the $l^{th}$ element of the vector $\mathbf{x}$ for $l \in \{1, \ldots, d\}$, spans the subspace of the centered nonlinear transformation of the $l^{th}$ input regressor.*

*Proof.* Consider $\boldsymbol{\Phi}_{\boldsymbol{C}l} = [\varphi(0; \ldots; x_1^{(l)}; \ldots; 0)^T - \mu_\varphi; \ldots; \varphi(0; \ldots; x_N^{(l)}; \ldots; 0)^T - \mu_\varphi]$ with $\mathbf{P}_{\mathbf{C}l} = \boldsymbol{\Phi}_{\boldsymbol{C}l}({\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T\boldsymbol{\Phi}_{\boldsymbol{C}l})^{-1}{\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T$ the projection matrix onto the subspace defined by the centered nonlinear transformation of the $l^{\text{th}}$ regressor variable. Defining $\boldsymbol{\Omega}_{\boldsymbol{C}l} = \boldsymbol{\Phi}_{\boldsymbol{C}l}{\boldsymbol{\Phi}_{\boldsymbol{C}}}^T$, then $\mathbf{P}_{\mathbf{C}l} = \boldsymbol{\Omega}_{\boldsymbol{C}l}({\boldsymbol{\Omega}_{\boldsymbol{C}l}}^T\boldsymbol{\Omega}_{\boldsymbol{C}l})^{-1}{\boldsymbol{\Omega}_{\boldsymbol{C}l}}^T$, which leads to $\mathbf{P}_{\mathbf{C}l} = \boldsymbol{\Phi}_{\boldsymbol{C}l}{\boldsymbol{\Phi}_{\boldsymbol{C}}}^T(\boldsymbol{\Phi}_{\boldsymbol{C}}{\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T\boldsymbol{\Phi}_{\boldsymbol{C}l}{\boldsymbol{\Phi}_{\boldsymbol{C}}}^T)^{-1}\boldsymbol{\Phi}_{\boldsymbol{C}}{\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T$. Using the SVD decomposition of the matrix $\boldsymbol{\Phi}_{\boldsymbol{C}}$, $\boldsymbol{\Phi}_{\boldsymbol{C}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$, replacing in the definition of $\mathbf{P}_{\mathbf{C}l}$ and reducing we obtain $\mathbf{P}_{\mathbf{C}l} = \boldsymbol{\Phi}_{\boldsymbol{C}l}({\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T\boldsymbol{\Phi}_{\boldsymbol{C}l})^{-1}{\boldsymbol{\Phi}_{\boldsymbol{C}l}}^T$. □

In the same way we can use $K(\mathbf{x}^{([l])}, \mathbf{x})$, with $\mathbf{x}^{([l])} = [x^{(1)}; \ldots; x^{(l-1)}; 0; x^{(l+1)}; \ldots; x^{(d)}]$ to form the kernel matrix $\boldsymbol{\Omega}_{\boldsymbol{C}(l)}$ which can be used as a basis for the subspace spanned by the centered nonlinear transformation of all the other input regressor variables, excluding the $l-$th input regressor, as well as their interaction effects. It is important to notice that the subspace spanned by the nonlinear transformation of more than one input regressor will contain their interaction effects as well as their individual contribution, or main effect. Hence, if the interest is to find solely the interaction effect between 2 variables on the output, their individual contributions should be found first and subtracted.

Once the basis for the main contributions, and interaction effects, of given input regressors have been found, the output of an LS-SVM regression model can be decomposed into additive components using Algorithm I.

## 2.3 Out-of-Sample Extension

In **Algorithm I** we presented a way to derive the main and interaction effect contributions using training data. This idea can be extended to data that has not been seen during training, however some considerations need to be taken into account. Since the algorithm that is presented in this paper is based on projection matrices, it is important to notice that a proper set of basis vectors are needed in order to construct reliable projection matrices. For instance, if while training the algorithm, it is found that the dimensions of the subspace where the data lies is $N_d$, then at least $N_d$ basis vectors are needed to construct a proper projection matrix. In addition, the number of

data points used to evaluate the model will define the size of the kernel matrix, which defines the maximum number of basis vectors that can be obtained from it. Therefore, in order to construct projection matrices for data-points outside the training set, and considering $N_d$ as the dimension of the subspace of interest, then at least $N_d$ new data points are needed in order to properly decompose the output into the nonlinear partial contributions. In practice, while decomposing the training data, the dimension of each one of the subspaces that represent the nonlinear transformation of the input variables, as well as the interaction effects, should be computed. The maximum dimension of such subspaces can be taken as the minimum number of evaluation points that are needed in order to produce a proper decomposition. The steps needed to evaluate NObSP using new data points are summarized in the Algorithm II.

**Algorithm I. Nonlinear Oblique Subspace Projections (NObSP)**

**Input:** regressor matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, output vector $\mathbf{y} \in \mathbb{R}^N$, estimated output vector $\hat{\mathbf{y}} \in \mathbb{R}^N$.
**Output:** $\hat{\mathbf{y}}^{(l)} \in \mathbb{R}^N$, and $\mathbf{y}^{(l)} \in \mathbb{R}^N$ main effect and interaction effect contributions from given input regressors using the estimated output from the model and the real measured output, respectively.

1. Normalize the input regressors.
2. Train an LS-SVM regression model using the training input/output data $\{\mathbf{X}, \mathbf{y}\}$.
3. Compute the kernel matrices $\mathbf{\Omega}_l$ and $\mathbf{\Omega}_{(l)}$, representing the subspaces spanned by the regressor(s) of interest, as explained in **Proposition 1.**
4. Center the Kernel matrices as follows: $\mathbf{\Omega}_C = \mathbf{M}\mathbf{\Omega}\mathbf{M}$, where $\mathbf{M} = \mathbf{I} - \mathbf{1}_N \mathbf{1}_N{}^T/N$ is a centering matrix, with $\mathbf{I}$ the identity matrix, $\mathbf{1}_N$ a column vector of ones, and $N$ the number of training data points.
5. Compute the oblique projector as in Eq (10) using the centered $\mathbf{\Omega}_l$ and $\mathbf{\Omega}_{(l)}$, $\mathbf{P}_{l/(l)} = \mathbf{\Omega}_l(\mathbf{\Omega}_l^T \mathbf{Q}_{(l)} \mathbf{\Omega}_l)^\dagger \mathbf{\Omega}_l^T \mathbf{Q}_{(l)}$, with $\mathbf{Q}_{(l)} = \mathbf{I}_N - \mathbf{\Omega}_{(l)}(\mathbf{\Omega}_{(l)}^T \mathbf{\Omega}_{(l)})^\dagger \mathbf{\Omega}_{(l)}^T$.
6. Compute the corresponding partial contribution to the output as $\hat{\mathbf{y}}^{(l)} = \mathbf{P}_{l/(l)}\hat{\mathbf{y}}$, or $\mathbf{y}^{(l)} = \mathbf{P}_{l/(l)}\mathbf{y}$ with $l = 1, \ldots, d$.

The output from both algorithms can be used in order to determine which input regressor(s) and/or interaction effects are of importance in the output of the model, by computing the percentage of power that each one of these contributions provide to the output.

# 3 Applications

In this section we present results from the proposed algorithm using 2 simulation examples, as well as a real life example using data from the Concrete Compressive Strength Data Set in the UCI machine learning repository [25]. In the first toy example we compare NObSP with the results given by COSSO, using the same example proposed in [11], where only main effects are included. Additionally, we evaluate the performance of NObSP to select relevant components, and we test the effect of the kernel selection in the decomposition. In the second example, we create an artificial dataset that includes interaction effects, we test for the robustness of the projections and the regression model by means of bootstrapping, we present the results in terms of their mean solution and 95% confidence intervals, in this example we also evaluated the

performance of the model to unseen test data. In the third example we demonstrate the potential use of NObSP in a real life example.

**Algorithm II. Out-of-Sample Extension for NObSP**

**Input:** regressor matrix with training samples $\mathbf{X}_{train} \in \mathbb{R}^{N_{train} \times d}$, output vector for training data $\mathbf{y}_{train} \in \mathbb{R}^{N_{train}}$, regressor matrix with test samples $\mathbf{X}_{test} \in \mathbb{R}^{N_{test} \times d}$, with $N_{test} \geqslant N_d$, with $N_d$ the largest dimension of the subspaces representing the nonlinear transformation of the input data.

**Output:** partial contribution and interaction effects from given input regressors in the test set $\hat{\mathbf{y}}_{test}^{(l)} \in \mathbb{R}^{N_{test}}$.

1. Normalize the input regressors, $\mathbf{X}_{train}$ and $\mathbf{X}_{test}$ .
2. Train an LS-SVM regression model using the training input/output data $\{\mathbf{X}_{train}, \mathbf{y}_{train}\}$.
3. Compute the kernel matrices for the test set $\mathbf{\Omega}_{l}^{(test)}$ and $\mathbf{\Omega}_{(l)}^{(test)}$, by evaluating the kernel function using samples from the test set, with $\mathbf{\Omega}_{l}^{(test)}(i,j) = K\big(x_{test}^{(l)}(i), x_{train}(j)\big)$ the $ij-th$ element of the kernel matrix, where $\mathbf{x}_{test}^{(l)}(i) \in \mathbb{R}^{1\times d}$ is the $i-$ th row of $\mathbf{X}_{test}$ where all elements are zero except the $l-$th component, and $\mathbf{x}_{train}(j) \in \mathbb{R}^{1\times d}$ is the $j-$ th row of $\mathbf{X}_{train}$. These matrices will represent the subspaces spanned by the regressor(s) of interest, as explained in **Proposition 1.**
4. Evaluate the output of the model, $\hat{\mathbf{y}}_{(test)}$ using Eq (9).
5. Center the Kernel matrices as follows: $\mathbf{\Omega}_{C}^{(test)} = \mathbf{\Omega}^{(test)} - \mathbf{M_1}\mathbf{\Omega}^{(test)} - \mathbf{\Omega}^{(test)}\mathbf{M_2} + \mathbf{M_1}\mathbf{\Omega}^{(test)}\mathbf{M_2}$, where $\mathbf{1}_{N_{test}}$ is a column vector of ones, $\mathbf{M_1} = \mathbf{1}_{N_{test}}\mathbf{1}_{N_{test}}^T/N_{test} \in \mathbb{R}^{N_{test}\times N_{test}}$ is a centering matrix , $\mathbf{M_2} = \mathbf{1}_{N_{train}}\mathbf{1}_{N_{train}}^T/N_{train} \in \mathbb{R}^{N_{train}\times N_{train}}$ is a centering matrix, $N_{train}$ is the number of training samples used, and $N_{test}$ is the number of test samples used.
6. Compute the oblique projector as in Algorithm I, using the centered $\mathbf{\Omega}_{l}^{(test)}$ and $\mathbf{\Omega}_{(l)}^{(test)}$.
7. Compute the corresponding partial contribution in the output as $\hat{\mathbf{y}}_{test}^{(l)} = \mathbf{P}_{l/(l)}\hat{\mathbf{y}}_{(test)}$, with $l = 1, \ldots, d$.

## 3.1 Simulation study: Toy example I

In order to compare the performance of NObSP with COSSO, we use the first example presented in [11], where an additive model in $\mathbb{R}^{10}$ is considered. The regression function is defined by $f(\mathbf{x}) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(3)}) + 6g_4(x^{(4)})$, where: $g_1(x) = x$, $g_2(x) = (2x-1)^2$, $g_3(x) = \frac{\sin(2\pi x)}{2-2\sin(2\pi x)}$, and $g_4(x) = 0.1\sin(2\pi x) + 0.2\cos(2\pi x) + 0.3\sin^2(2\pi x) + 0.4\cos^3(2\pi x) + 0.5\sin^3(2\pi x)$. The model contains 4 main effects, no interaction effects, and no influence from the inputs $x^{(5)}$ until $x^{(10)}$. We generated the input data, $\mathbf{x}$, as uniformly distributed random numbers in $[0, 1]$. Additionally, we added noise such that the signal-to-noise ratio is $3:1$ as in [11]. In contrast with the original example, we also imposed a correlation between the input variables of 0.8 in order to increase the complexity of the problem. We use COSSO in 2 ways, first to find only the main effects in the regression, i.e. decompose the output only in additive terms of the partial nonlinear contributions of the input regressors, we will refer to this as COSSO-I. Second, we also computed the output using COSSO for the second order interactions, retrieving not only the main effects but also

the interaction effect, to compare if both approaches for COSSO converge to the same result. We refer to this as COSSO-II.

The results for the decomposition are shown in figure 1. The results from NObSP are presented in a solid gray line, COSSO-I in a black dashed line and COSSO II in a black dotted line. The true output is shown as a black solid line. As can be seen in the figure, the proposed algorithm is able to approximately retrieve the functional form for the contribution of each input regressor variable, with a performance similar to the one provided by COSSO. It is also possible to observe that the output provided by COSSO-I and COSSO-II are not the same. In addition, in contrast with COSSO, NObSP is not able to retrieve a response equal to zero for the contribution of the input variables that are not included in the final model. This is expected, since the proposed algorithm does not include any sparsity or feature selection criteria, instead it is solely based on the output of an existing regression model. However, the magnitude of the contribution of those regressors to the output is smaller than the contribution of the variables that are effectively included in the original model.

In figure 2, the strength of the nonlinear contributions of the input variables and the interaction effects are shown. To compute this strength, first we computed the total power of the decomposed signal as the sum of the root mean squared values from all the components obtained in the decomposition model. Once this is obtained, the strength of each component is computed as the ratio between its root mean squared value and the total power of the decomposed signal. To visualize the magnitude of the main nonlinear contributions and the interaction effects, we present these values in a matrix form. In this matrix the diagonal represents the strength of the contributions for each input variable, and the upper triangular elements represent the strength of the contribution for second order interactions for the corresponding input regressors, which are indicated in the rows and columns of the matrix. The lower triangular elements are not taken into account since they represent redundant information. Finally, the fields in the matrix that belong to the contributions with a larger influence on the output are colored in black. For COSSO-I only the diagonal produces elements different from 0, since it only retrieves main effects, while COSSO-II and NObSP produce a total of 55 components. It can be seen that the results from NObSP indicate that the first 4 components have a higher contribution to the output than the other components, with a contribution between [4.13% - 18.20%] of the total power, compared to [12.48% - 43.01%] produced by COSSO-II. COSSO-I also produces components with a larger magnitude in the first 4 components with a root mean square value of [12.51% - 44.47%] and 0 in the other components. In contrast with NObSP, COSSO-I and COSSO-II produce components with a contribution equal to zero, due to its sparsity properties. Since NObSP does not include sparsity, most of its components contribute to the output. This results in lower strength values for the components with a higher influence in the output for NObSP. However, by selecting an appropriate threshold, these components can be selected. For this example the threshold was set to 4% by visual inspection of the components.

In figure 3 the results from the decomposition using NObSP with different kernels are shown. It can be seen that the linear kernel only produces the linear approximation of the nonlinear contributions of the input regressors on the output. Additionally, the contributions of the interaction effects, in the linear model, are equal to zero, which is expected since the interaction effects in linear models are given by a sum of the contribution of each input variable on the output. Therefore, by identifying such contributions and subtracting from the interaction effect the resulting component should be equal to zero. The performance of the polynomial kernel and the RBF kernel is quite similar, in both cases the decompositions were able to approximate the desired functional forms.

In table 1 the root mean square errors for the regression, and the estimation of the

functional forms using COSSO-I, COSSO-II and NObSP are shown. The simulations were performed 100 times and the values are presented as median [min-max]. It can be seen that COSSO-I and COSSO-II in general produced a lower error in the estimation of the overall function, with COSSO-I producing the lowest error. However, when estimating the functional forms of the main effects, NObSP, using the RBF kernel, outperforms the results from COSSO.

## 3.2 Simulation study: Toy example II

For the second toy example, we use the following additive model in $\mathbb{R}^3$:

$$y = \sin(2\pi x_1) + 1.4e^{x_2} + \cos\left(4\pi(x_1 - x_2)\right) + \eta; \tag{13}$$

with, $x_1, x_2, x_3 \in [0, 1]$, and $\eta$ is a Gaussian noise with a variance such that the signal-to-noise ratio on the output variable is equal to 4db. No contribution from $x_3$ was included. In addition, a correlation of 0.8 was imposed in the input signals.

We computed 400 realizations of the model. From this data, we extracted 1000 different training datasets using random sampling with replacement. For each generated dataset a model using LS-SVM was computed and tuned using 10-fold cross-validation. The partial contributions of the input variables and their second order interactions were calculated using NObSP and COSSO-II [11]. From the output of the 1000 randomizations, the mean and the 95% confidence intervals were computed. Additionally, we computed the output of the model for unseen data using **Algorithm II**. For this, we first found the rank for the different kernels obtained using **Algorithm I**, the maximum rank obtained was 45. Based on that rank we run simulations using a test set size from 1 data point up to $45 \times 5$ data points. For each test set size we generate 100 random sets. In order to identify the convergence of the error in the decomposition with respect to the test set size, we plotted the errors in terms of their mean and 95% confidence interval.

In figure 4 the results from NObSP using **Algorithm I** are shown. The gray area in the figure represents the 95% confidence intervals, the dotted black line represents the mean value for the regression, while the black solid line represents the true component used as reference. This same convention will be used for the rest of the figures. It is important to notice that the mean values plotted in the figures do not represent the output of one decomposition of the regression models, but just the mean value of the output for that specific input value. This gives the impression that the models produce noisy estimates, but that is not the case. In the figure it can be seen that NObSP is able to approximately retrieve the functional form of the simulated model. In addition, the contribution from the input $x_3$ is small in amplitude compared to the other components, indicating that this variable does not contribute largely on the output. The second-order interactions appear to produce larger confidence intervals, indicated by the gray area. However, NObSP was still able to retrieve the approximate functional form of the second-order interaction between the variables $x_1$ and $x_2$, while producing a small output for the components that were not present in the original model. Figure 5 shows the results from the smoothing spline ANOVA models using COSSO-II. It can be seen that COSSO-II, in this case example, is not able to retrieve the desired functional forms. But, when looking at the error in the complete regression model the LS-SVM regression produce a root mean square error of 0.09 [0.02-0.11], while COSSO-II produce an error of 0.04 [0.03-0.05], indicating that COSSO II was able to fit better the model than NObSP.

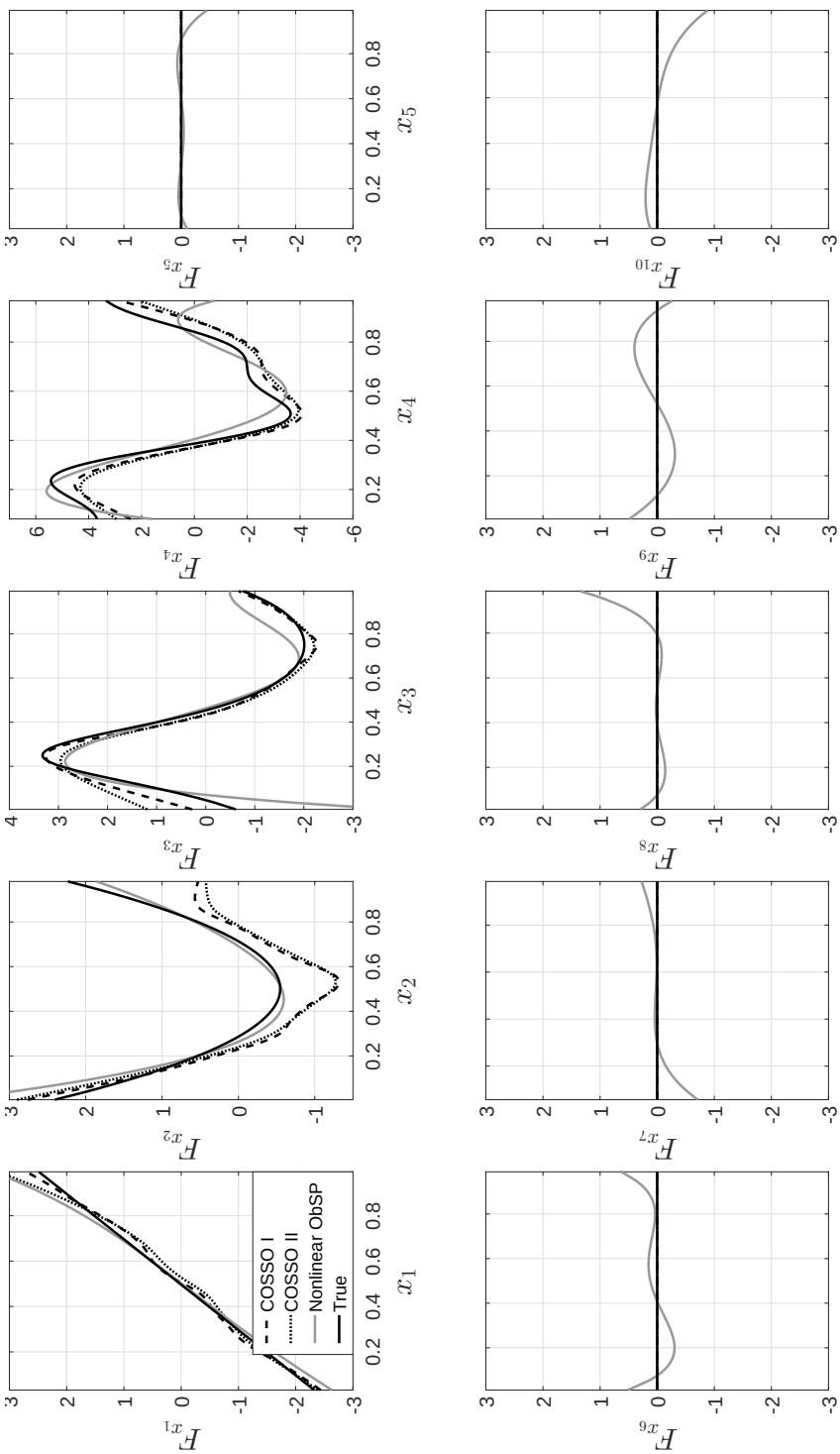



**Fig 1.** Results from the decomposition of the output for the toy example I using NObSP (gray solid line), COSSO I dashed black line, and COSSO II in a black dotted line. The solid black line represents the true component.

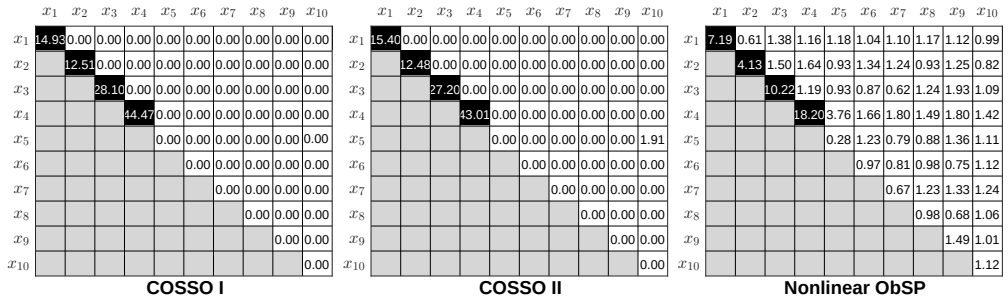



**Fig 2.** Percentage of the contributions of the input regressors, and their interaction effects, on the output. The components with a larger contribution in the final model are indicated by a black square.

In order to evaluate whether the problem with the estimation was caused by the correlation between the input variables, we repeated the simulation using an input with uncorrelated inputs. The results are shown in figure 6. It can be seen that COSSO-II now is able to retrieve, approximately, the functional forms for the variables $x_1$ and $x_2$, it also identifies that $x_3$, as well as some interaction effects, do not have a contribution in the model. But, it still fails to identify the second-order interaction that was present in the original model. The output for the LS-SVM model using NObSP for this case is shown in figure 7. The figure shows that in this case NObSP can still retrieve the approximate functional forms, as shown previously.

Additionally, the output for NObSP using the unseen test set can be seen in figure 8 using 250 data points. The figure shows that NObSP is able to retrieve the functional forms of the relationship between the input and the output of the model using new data samples. Furthermore, in figure 9 we can see that the errors for the estimation of the functional components decrease as the size of the test set increases. This error converges around 45 samples, which is the maximum rank that was needed in order to construct appropriate projection matrices to decompose the output of the model.

## 3.3 Simulation study II: Concrete Compressive Strength Dataset

In this section we used data from the concrete compressive strength database from the UCI machine learning repository. We tested the performance of NObSP and compared with COSSO-II. In this dataset there are 8 different input variables that are used in order to predict the strength of the concrete samples. The input variables are: the amount of cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate, and the age of the cement in days. In total, the dataset contains 1030 samples. More information about the characteristics of the dataset can be found in [25]. For the simulations, we computed 100 realizations of the regression models, using random sampling with replacement. We express the output in terms of the mean and 95% confidence intervals.

Figure 10 shows the results from NObSP. The figure displays not only the decomposition but also the general performance for the regression model. Based on the magnitude of the responses, the elements that contribute the most to the strength of the concrete samples are: amount of cement, furnace slag, ash, and water. Other elements, such as superplasticizer, course and fine aggregate, do not contribute strongly to the final strength of the concrete sample.

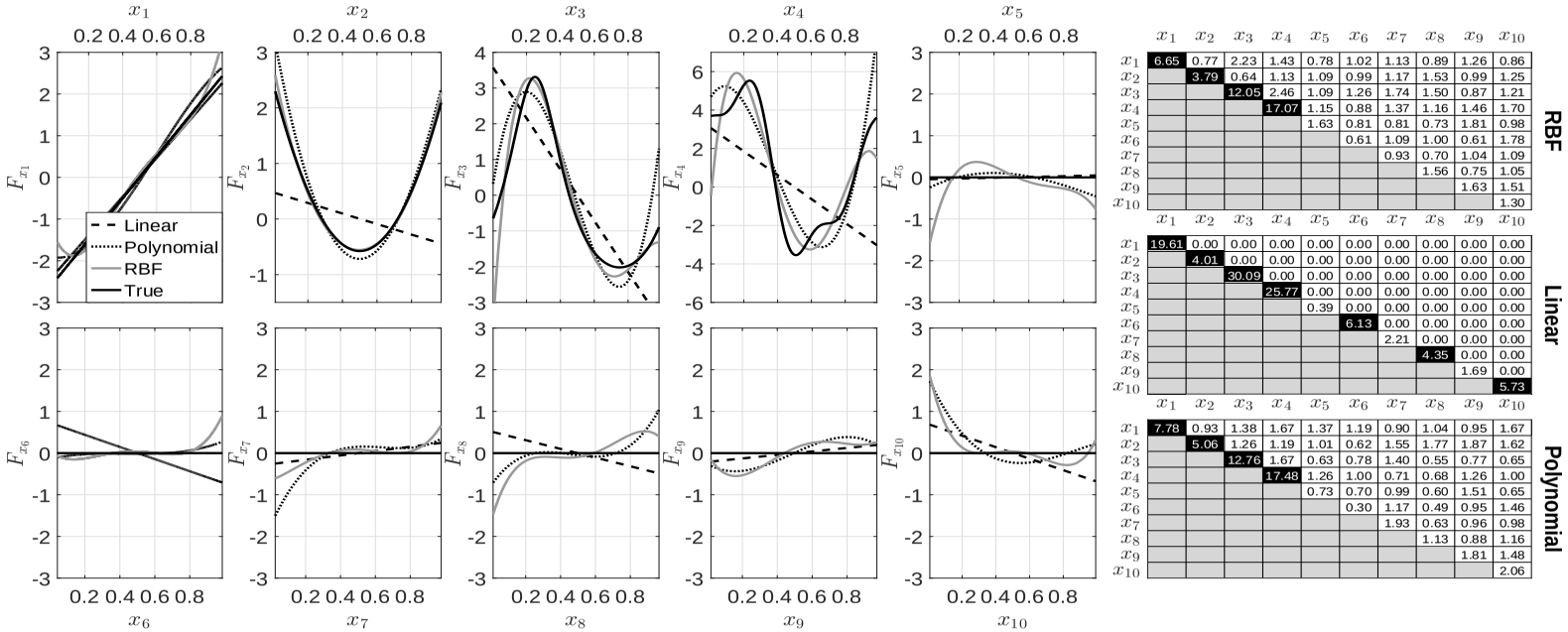



**Fig 3.** Output of NObSP using different kernels, in black dashed line for the linear kernel, in black dotted line for the polynomial kernel, and in gray solid line for the RBF kernel. The black solid line represents the true contribution.

**Table 1.** Root mean square error for regression models and the estimation of the true functional forms using NObSP with three different kernels, COSSO-I, and COSSO-II. Values are expressed as median [min-max].

| | $F_{\times}$ | $F_{x_1}$ | $F_{x_2}$ | $F_{x_3}$ | $F_{x_4}$ | $F_{x_5}$ | $F_{x_6}$ | $F_{x_7}$ | $F_{x_8}$ | $F_{x_9}$ | $F_{x_{10}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| COSSO-I | **1.40[1.33-1.47]** | 0.25[0.15-0.55] | 0.53[0.41-0.78] | 0.34[0.17-0.43] | 1.30[1.08-1.40] | **0[0-0]** | **0[0-0.91]** | 0[0-0.54] | **0[0-0.40]** | **0[0-0.20]** | **0[0-0]** |
| COSSO-II | 1.55[1.39-1.59] | 0.41[0.15-1.05] | 0.54[0.51-0.63] | 0.67[0.27-0.86] | 1.49[1.26-1.57] | 0[0-0] | 0[0-1.02] | **0[0-0.52]** | 0[0-0.71] | 0[0-1.04] | **0[0-0]** |
| $\text{NObSP}_{\text{Linear}}$ | 3.30[3.24-3.46] | 0.39[0.16-0.47] | 0.56[0.54-0.72] | 0.97[0.91-1.00] | 2.46[2.44-2.48] | 0.31[0.15-0.50] | 0.55[0.34-0.68] | 0.20[0.04-0.25] | 0.28[0.16-0.46] | 0.02[0-0.19] | 0.03[0-0.26] |
| $\text{NObSP}_{\text{Poly}}$ | 2.78[2.65-2.78] | 0.18[0.11-0.22] | 0.24[0.16-0.35] | 0.48[0.45-0.55] | 1.27[1.21-1.29] | 0.15[0.07-0.35] | 0.25[0.16-0.37] | 0.20[0.06-0.26] | 0.20[0.08-0.31] | 0.16[0.12-0.20] | 0.26[0.11-0.48] |
| $\text{NObSP}_{\text{RBF}}$ | 2.26[2.09-2.45] | **0.16[0.07-0.28]** | **0.19[0.12-0.27]** | **0.28[0.24-0.35]** | **0.86[0.84-0.88]** | 0.18[0.13-0.38] | 0.19[0.05-0.25] | 0.14[0.09-0.34] | 0.14[0.05-0.20] | 0.17[0.10-0.31] | 0.16[0.08-0.33] |

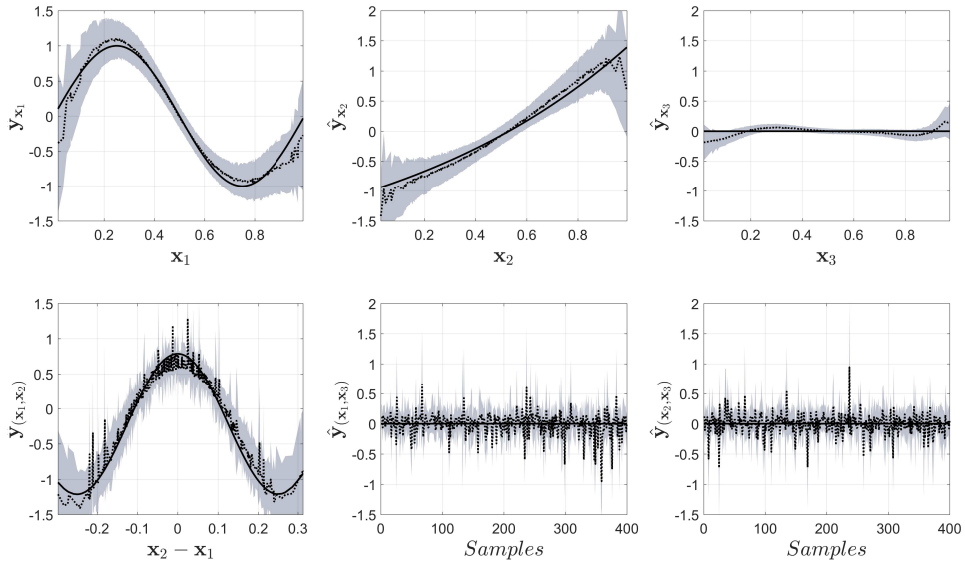



**Fig 4.** Output from NObSP, the black solid line represents the true component, the dotted black line represents the mean for the estimations using NObSP, and the gray area represents the 95% confidence interval obtaining using bootstrap.

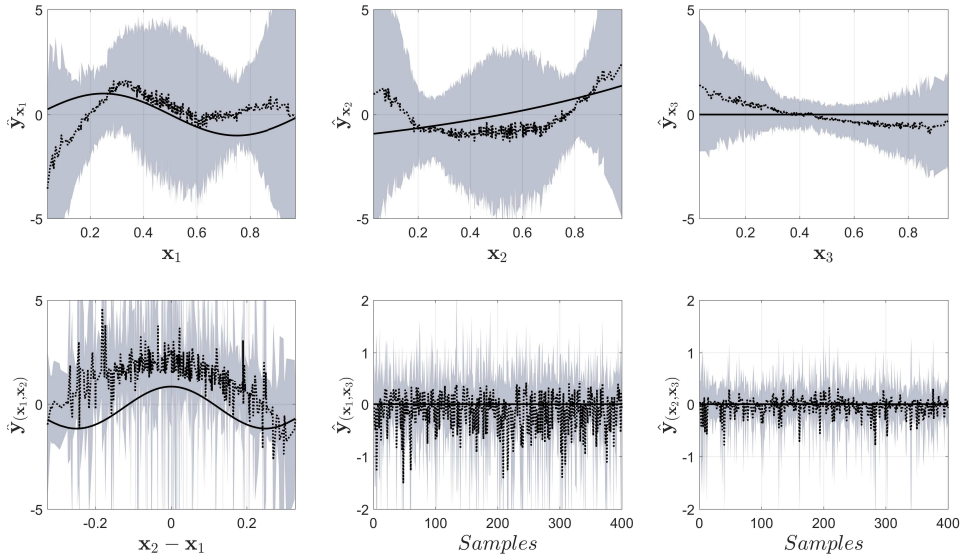



**Fig 5.** Output of the model for the decomposition of the measured output into the main and interaction effect contributions using COSSO-II in smoothing spline ANOVA models for second-order interactions. The black solid line represents the true component, the dotted black line represents the mean for the estimations using NObSP, and the gray area represents the 95% confidence interval obtained using bootstrap.

# 4 Discussion

We have proposed an algorithm to decompose the output of an LS-SVM regression model into the sum of the partial nonlinear contributions, as well as interaction effects, of the input regressors. We have demonstrated that the functional form of the relation

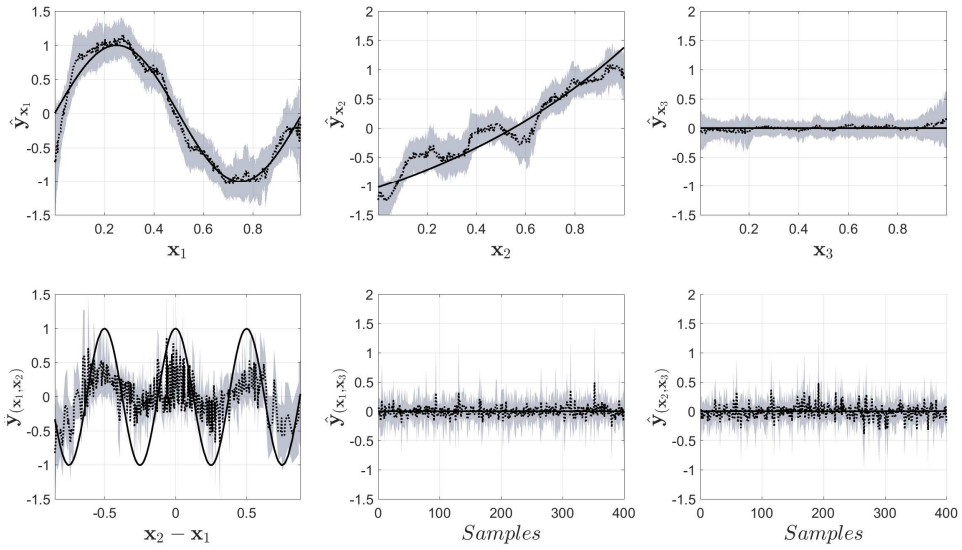



**Fig 6.** Output of the model for the decomposition using COSSO-II and non-correlated inputs. The black solid line represents the true component, the dotted black line represents the mean for the estimations using NObSP, and the gray area represents the 95% confidence interval obtaining using bootstrap.

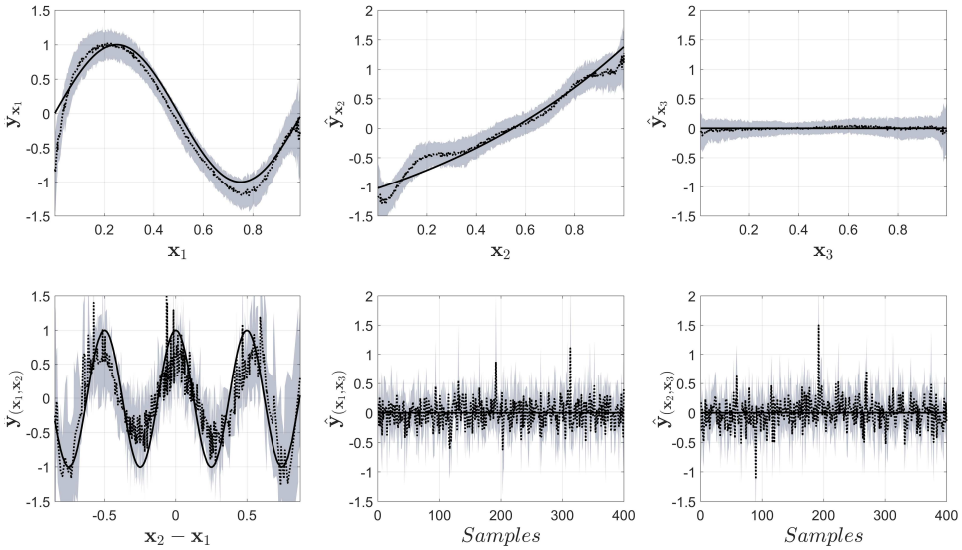



**Fig 7.** Output of the model for the decomposition using NObSP in the LS-SVM regression model with non-correlated inputs. The black solid line represents the true component, the dotted black line represents the mean for the estimations using NObSP, and the gray area represents the 95% confidence interval obtaining using bootstrap.

between the inputs and the output can be retrieved using oblique subspace projections. We have also shown that through appropriate kernel evaluations it is possible to find a proper basis for the subspace representing the nonlinear transformation of the input regression variables.

In contrast with other methodologies, the proposed algorithm does not require to define a-priori the model contributions for the decomposition. NObSP starts with a normal LS-SVM regression model, which solution, in terms of the kernel parameters, is

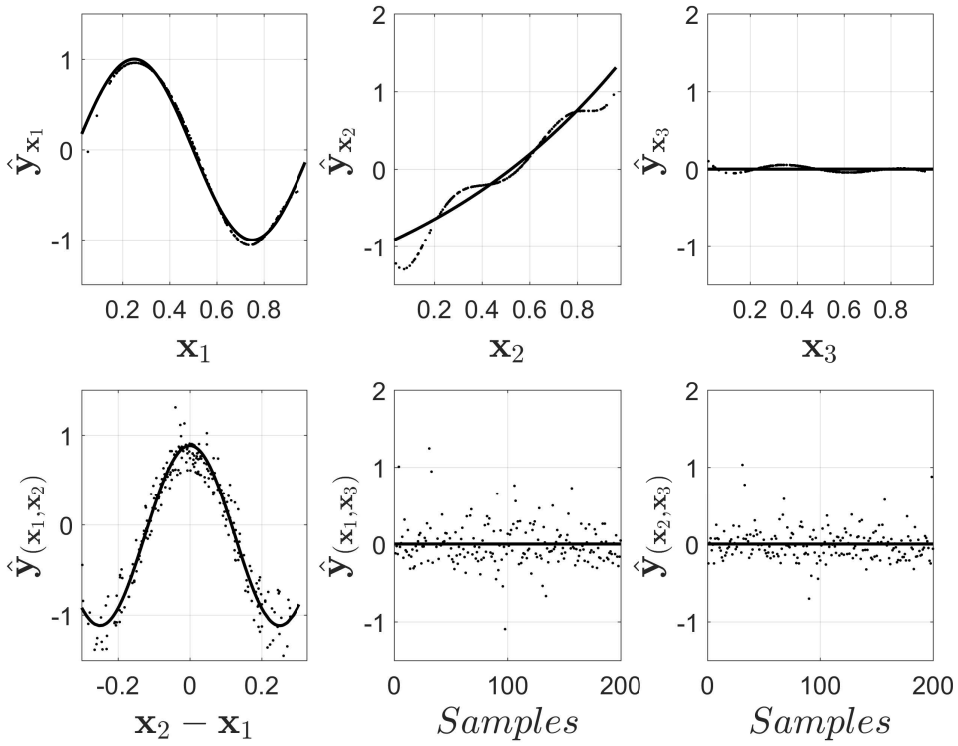



**Fig 8.** Results provided by NObSP using unseen data. The size of the test set was 250 samples. the samples were obtained in the same way as the training samples were produced. The solid line represents the reference functional forms, while the dots represent the output of the decomposition scheme

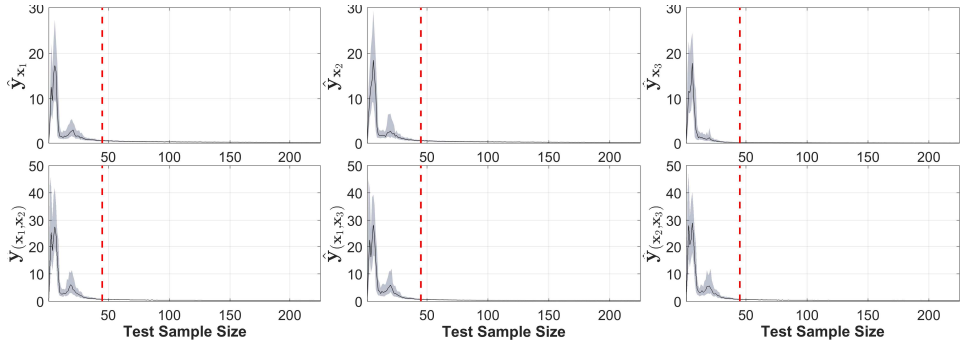



**Fig 9.** Convergence of the error between the real functional form and the estimated output provided by NObSP for different test set sizes. The solid line represents the median value and the shade area represents the 25 and 75 percentiles of the error. For each test set size 100 random simulations were performed. The red dashed line indicates the maximum rank obtained for the kernel matrices during training, which in this particular case was 45.

used in order to implement the decomposition strategy. Other methodologies, such as COSSO, require to define a-priori whether the user is interested to decompose the output in terms of the main effects, or also to include second-order interactions. This might lead to different functional forms since there is no guarantee that both approaches will converge to the same solution. This was demonstrated in section 3.1 where COSSO-I and COSSO-II produced different outputs. In NObSP this is not a

problem since the decomposition scheme does not require to solve an additional optimization problem.

The basis for the subspace spanned by the nonlinear transformation of the data, which is computed using kernel evaluations, can be seen as an $N-$dimensional approximation of the true subspace, which might be embedded in a much larger space. When using a linear kernel, it can be seen that the subspaces of the nonlinear transformations, as expected, cannot be properly constructed, resulting just in linear approximations of the true nonlinear model, as shown in figure 3. When using a polynomial kernel, the decomposition scheme performs better than the linear case, being able to produce a better estimation of the nonlinear influences of the input regressors. However, as shown in section 3.1, the best results were obtained using the RBF kernel. This might be explained by the fact that the RBF kernel can be considered as an universal kernel, which is able to approximate the responses produced by any other kernel by changing its bandwidth.

In the second toy example, it was shown that NObSP was able to approximately identify the functional forms for the main effects, and second-order interaction effects on the output, in both cases when the input variables were correlated and uncorrelated. COSSO-II was only able to retrieve the main effects when the variables were not correlated. However, COSSO-II was able to identify more accurately the components that are not part of the original model.

When considering the concrete strength example, it can be seen that the decomposition scheme using NObSP produces results with small confidence intervals in most of the components. More importantly, the results provided seem to agree with the literature, where the strength of the concrete sample is expected to increase with increasing amounts of cement and decreasing amount of water, and it increases during the first days [25, 26]. Conversely, NObSP indicates that by increasing the amount of furnace slag, the strength of the cement increases, while Yeh et al., have reported the opposite [26]. However, the datasets used in both analyses are different, which might lead to different decompositions. In addition, it is important to notice that their analysis is done considering that several variables are kept constant, while in our case this restriction was not applied.

One of the main drawbacks of NObSP, in contrast with COSSO, is that it does not provide a closed form for the estimation of the nonlinear contributions and the interaction effects. However we have shown that it is possible to evaluate the decomposition scheme when a sufficient number of test samples are available. This is due to the fact that the decomposition is based on the projection of the observed data onto the respective subspace. Therefore, it is important that an adequate number of basis vectors is used in order to construct proper projection matrices. We have shown that the number of data points that are required in order to construct this projector is equal to the maximum rank of the kernel matrices that are created during training. In case that a closed form is required, one solution can be to introduce a multiple kernel learning problem similarly to the one proposed in [27]. This is out of the scope of this paper and will be addressed in future studies.

Additionally, there are some numerical issues that should be considered. Since kernel evaluations are needed to create a basis for the subspace spanned by the nonlinear transformation of a given variable, when using too many input regressors, it is possible that the computation of the Euclidean distances produce values close to machine precision. This is caused by the fact that in such cases several variables are set to 0, leaving only the variable of interest intact. In such cases, the results provided by NObSP are not reliable. Therefore, it is recommended to use this decomposition scheme with a reduced set of input variables. In addition, it is also recommended to normalize the input regressors, in order to guarantee that the estimation of the distances is not

biased by the magnitude of only one input variable.

In summary, the advantages and disadvantages of NObSP and COSSO can be summarized as follows:

- NObSP does not require the define a priori the kind of relations of interest between the input regressors and the output. COSSO does require to define if the user is interested in the main effects or also in the interaction effects.
- NObSP converge to the solution whether the user is interested in finding the main and interaction effects. COSSO produces different results.
- NObSP is able to retrieve the functional forms even in the presence of correlated input regressors.
- COSSO produce models that are sparse. NObSP does not include sparsity in its definition.
- COSSO is able to retrieve close forms for the evaluation of new samples. NObSP, requires to construct projection matrices which require the evaluation of a minimum amount of points to produce a proper output.
- In contrast with COSSO, NObSP suffers of numerical issues due to the evaluation of kernel functions with vectors that contain a considerable amount of zeros.

NObSP can also be extended to the analysis of dynamical systems. For instance, NObSP can be introduced naturally for the nonlinear identification of dynamical systems using subspace identification techniques such as the one proposed in [28]. In the actual formulation, the relation between the input variables and the output in the regression model is considered static. However, by using as input variables a Hankel expansion, or any other expansion, of the input regressors, it is possible to consider a dynamic nonlinear model, introducing this framework in the field of system identification.

Finally, it is interesting to notice that NObSP relies only on the tuning of a kernel in order to decompose the output in the partial nonlinear contributions of the input and the interaction effects. This provides insights about the geometry of the regression models using kernels, by linking the space where the original data is embedded and the subspaces generated by the nonlinear transformation of the data. This also indicates that this methodology is not limited to the use in LS-SVM model, but it can be adapted to any kind of kernel-based regression models, where the output can be represented as a linear combination of kernel evaluations, like in equation Eq (6). Such models include kernel principal component regression [29, 30], kernel partial least squares [31], and kernel ridge regression [32].

# 5 Conclusions

In this manuscript we proposed a decomposition scheme using oblique subspace projections, called NObSP, which is able to retrieve relevant information about the input variables and their relationship with the output in an LS-SVM regression model, facilitating its interpretation. The performance of the proposed model was demonstrated using 2 different toy examples as well as a practical example from a public database. This methodology has a huge potential in many fields, including biomedical applications. For instance, it can be used for the study of the interactions between different physiological signals, providing extra information for the understanding of some underlying regulatory mechanisms, and supporting clinical diagnosis and treatment.

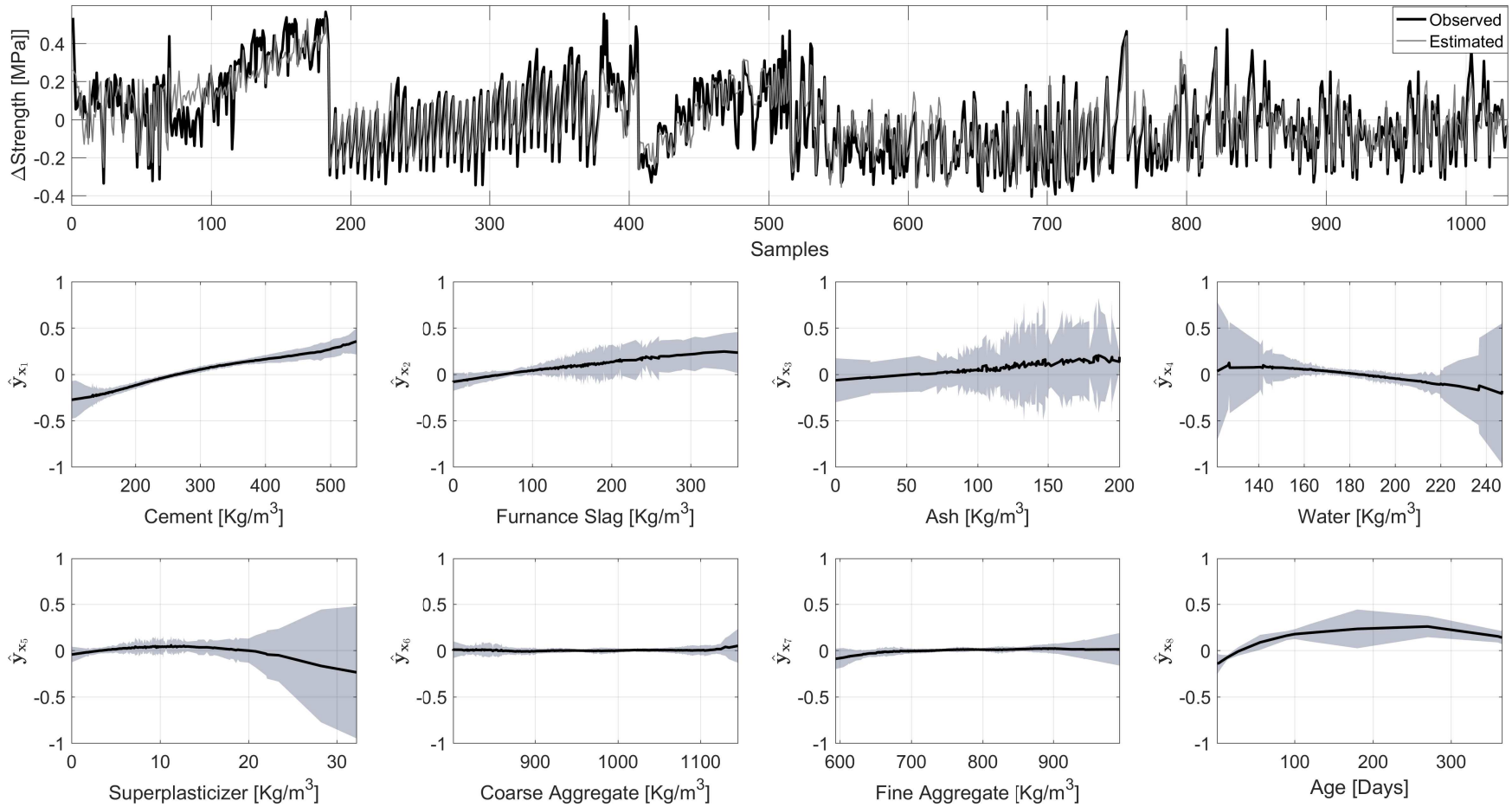



**Fig 10.** Model fitting, presented in the top plot, and estimation of the functional form of the contribution of the input variables in the concrete strength dataset from the UCI machine learning repository. In the top plot, the black line represents the measured strength of the cement mix, and in gray the estimated value using an LS-SVM model. For the contributions, the gray area represents the 95% confidence interval, obtained using bootstrap, and the solid black line the mean value of the contribution.

## Acknowledgment

Alexander Caicedo was a Postdoctoral Fellow of the Flemish Fund for Scientific Research (FWO). This research is supported by the following fund: Bijzonder Onderzoeksfonds KU Leuven (BOF): SPARKLE – Sensor-based Platform for the Accurate and Remote monitoring of Kinematics Linked to E-health #: IDO-13-0358; The effect of perinatal stress on the later outcome in preterm babies #: C24/15/036; TARGID - Development of a novel diagnostic medical device to assess gastric motility #: C32-16-00364. Fonds voor Wetenschappelijk Onderzoek-Vlaanderen (FWO): Hercules Foundation (AKUL 043) 'Flanders BCI Lab - High-End, Modular EEG Equipment for Brain Computer Interfacing'. Agentschap Innoveren en Ondernemen (VLAIO): 150466: OSA+. Agentschap voor Innovatie door Wetenschap en Technologie (IWT): O&O HBC 2016 0184 eWatch. imec funds 2017. imec ICON projects: ICON HBC.2016.0167, 'SeizeIT'. Belgian Foreign Affairs-Development Cooperation: VLIR UOS programs (2013-2019). EU: European Union's Seventh Framework Programme (FP7/2007-2013). The HIP Trial: #260777. ERASMUS +: INGDIVS 2016-1-SE01-KA203-022114. European Research Council: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (n° 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. EU H2020-FETOPEN 'AMPHORA' #766456.

## References

1. Harrell F. Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis. Springer; 2015.

2. DeVore R, Petrova G, Wojtaszczyk P. Approximation of functions of few variables in high dimensions. Constructive Approximation. 2011;33(1):125–143.

3. Van Belle V, Van Calster B. Visualizing risk prediction models. PloS one. 2015;10(7):e0132614.

4. Van Belle V, Lisboa P. White box radial basis function classifiers with component selection for clinical prediction models. Artificial intelligence in medicine. 2014;60(1):53–64.

5. Ravikumar P, Liu H, Lafferty J, Wasserman L. Spam: Sparse additive models. In: Proceedings of the 20th International Conference on Neural Information Processing Systems. Curran Associates Inc.; 2007. p. 1201–1208.

6. Huang JZ. Functional ANOVA models for generalized regression. Journal of multivariate analysis. 1998;67(1):49–71.

7. Abramovich F, Angelini C. Testing in mixed-effects FANOVA models. Journal of statistical planning and inference. 2006;136(12):4326–4348.

8. Hastie T, Tibshirani R. Generalized additive models: some applications. Journal of the American Statistical Association. 1987;82(398):371–386.

9. Gunn SR, Kandola JS. Structural modelling with sparse kernels. Machine learning. 2002;48(1-3):137–163.

10. Signoretto M, Pelckmans K, Suykens JA. Quadratically constrained quadratic programming for subspace selection in kernel regression estimation. In: International Conference on Artificial Neural Networks. Springer; 2008. p. 175–184.

11. Lin Y, Zhang HH, et al. Component selection and smoothing in multivariate nonparametric regression. The Annals of Statistics. 2006;34(5):2272–2297.

12. Bring J. A geometric approach to compare variables in a regression model. The American Statistician. 1996;50(1):57–62.

13. Caicedo A, Varon C, Hunyadi B, Papademetriou M, Tachtsidis I, Van Huffel S. Decomposition of near-infrared spectroscopy signals using oblique subspace projections: applications in brain hemodynamic monitoring. Frontiers in physiology. 2016;7.

14. Takane Y, Yanai H. On oblique projectors. Linear Algebra and its Applications. 1999;289(1-3):297–310.

15. Tu TM, Shyu HC, Lee CH, Chang CI. An oblique subspace projection approach for mixed pixel classification in hyperspectral images. Pattern recognition. 1999;32(8):1399–1408.

16. Behrens RT, Scharf LL. Signal processing applications of oblique projection operators. Signal Processing, IEEE Transactions on. 1994;42(6):1413–1424.

17. Eigensatz M, Pauly M. Insights into the Geometry of the Gaussian Kernel and an Application in Geometric Modeling. Master thesis, Swiss Federal Institute of Technology Zurich; 2006.

18. Suykens JA, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J. Least squares support vector machines. World Scientific Publishing Company Incorporated; 2002.

19. Kaytez F, Taplamacioglu MC, Cam E, Hardalac F. Forecasting electricity consumption: a comparison of regression analysis, neural networks and least squares support vector machines. International Journal of Electrical Power & Energy Systems. 2015;67:431–438.

20. Kisi O, Parmar KS. Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution. Journal of Hydrology. 2016;534:104–112.

21. Zhu B, Wei Y. Carbon price forecasting with a novel hybrid ARIMA and least squares support vector machines methodology. Omega. 2013;41(3):517–524.

22. Mellit A, Pavan AM, Benghanem M. Least squares support vector machine for short-term prediction of meteorological time series. Theoretical and applied climatology. 2013;111(1-2):297–307.

23. Suykens JA, De Brabanter J, Lukas L, Vandewalle J. Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing. 2002;48(1):85–105.

24. Yanai H, Takeuchi K, Takane Y. Projection matrices, generalized inverse matrices, and singular value decomposition. Springer; 2011.

25. Yeh IC. Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete research. 1998;28(12):1797–1808.

26. Yeh IC. Analysis of strength of concrete using design of experiments and neural networks. Journal of Materials in Civil Engineering. 2006;18(4):597–604.

27. Bach FR, Lanckriet GR, Jordan MI. Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the twenty-first international conference on Machine learning. ACM; 2004. p. 6.

28. Van Overschee P, De Moor B. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica. 1994;30(1):75–93.

29. Rosipal R, Cichocki A, Trejo LJ, of Computing PUUKD, Systems; I. Kernel principal component regression with em approach to nonlinear principal components extraction. University of Paisley; 2000.

30. Hoegaerts L, Suykens J, Vandewalle J, De Moor B. Subset based least squares subspace regression in RKHS. Neurocomputing. 2005;63:293–323.

31. Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel hilbert space. The Journal of Machine Learning Research. 2002;2:97–123.

32. Vovk V. Kernel ridge regression. In: Empirical inference. Springer; 2013. p. 105–116.