RKHSMetaMod: An R package to estimate the Hoeffding decomposition of a complex model by solving RKHS ridge group sparse optimization problem

HALALEH KAMARI, SYLVIE HUET, MARIE-LUCE TAUPIN

December 28, 2021

Abstract

In this paper, we propose an R package, called RKHSMetaMod, that implements a procedure for estimating a meta-model of a complex model. The meta-model approximates the Hoeffding decomposition of the complex model and allows us to perform sensitivity analysis on it. It belongs to a reproducing kernel Hilbert space that is constructed as a direct sum of Hilbert spaces. The estimator of the meta-model is the solution of a penalized empirical least-squares minimization with the sum of the Hilbert norm and the empirical $\text{L}_2$-norm. This procedure, called RKHS ridge group sparse, allows both to select and estimate the terms in the Hoeffding decomposition, and therefore, to select and estimate the Sobol indices that are non-zero. The RKHSMetaMod package provides an interface from R statistical computing environment to the C++ libraries Eigen and GSL. In order to speed up the execution time and optimize the storage memory, except for a function that is written in R, all of the functions of this package are written using the efficient C++ libraries through RcppEigen and RcppGSL packages. These functions are then interfaced in the R environment in order to propose a user-friendly package.

Keywords: Meta model, Hoeffding decomposition, Ridge Group Sparse penalty, Reproducing Kernel Hilbert Spaces.

1 Introduction

Consider a phenomenon described by a model $m$ depending on $d$ input variables $X = (X_1, ..., X_d)$. This model $m$ from $\mathbb{R}^d$ to $\mathbb{R}$ may be a known model that is calculable in all points of $X$, i.e. $Y = m(X)$, or it may be an unknown regression model defined as follows:

$$Y = m(X) + \varepsilon,$$

where the error $\varepsilon$ is assumed to be centered with a finite variance, i.e. $E(\varepsilon) = 0$ and $\text{var}(\varepsilon) < \infty$. The components of $X$ are independent with a known law $P_X = \prod_{a=1}^{d} P_{X_a}$ on $X$, a subset of $\mathbb{R}^d$. The number $d$ of components of $X$ may be large. The model $m$ may present high complexity as strong non-linearities and high order interaction effects, and it is assumed to be square-integrable, i.e. $m \in L^2(X,P_X)$. Based on the data points $\{(X_i,Y_i)\}_{i=1}^{n}$, we estimate a meta-model that approximates the Hoeffding decomposition of $m$. This meta-model belongs to a reproducing kernel Hilbert space (RKHS), which is constructed as a direct sum of the Hilbert spaces leading to an additive decomposition including variables and interactions between them (Durrande, Ginsbourger, Roustant, and Carraro 2013). The estimator of the meta-model is calculated by minimizing an empirical least-squares criterion penalized by the sum of two penalty terms: the Hilbert norm and the empirical norm (Huet and Taupin 2017). This procedure allows us to select the subsets of variables $X_1, ..., X_d$ that contribute to predict $Y$. The estimated meta-model is used to perform sensitivity analysis, and therefore, to determine the influence of each variable and groups of them on the output variable $Y$.

In the classical framework of sensitivity analysis, the function $m$ is calculable in all points of $X$, and one may use the method of Sobol (1993) to perform sensitivity analysis on $m$. Let us briefly introduce this

*Halaleh Kamari, Université Paris-Saclay, France, @, Sylvie Huet, INRAE, France, @, Marie-Luce Taupin, Université Evry Val d’Essonne, France, @.
method: The independency between the components of \( X \) leads to write the function \( m \) according to its Hoeffding decomposition (\( \text{Sobol 1993} \), \( \text{Van der Vaart 1998} \)):

\[
m(X) = m_0 + \sum_{a=1}^{d} m_a(X_a) + \sum_{a,a'} m_{a,a'}(X_a, X_{a'}) + ... + m_{1,...,d}(X).
\]

(2)

The terms in this decomposition are defined using the conditional expected values:

\[
m_0 = E_X(m(X)), \quad m_a(X_a) = E_X(m(X)|X_a) - m_0; \quad m_{a,a'}(X_a, X_{a'}) = E_X(m(X)|X_a, X_{a'}) - m_a(X_a) - m_{a'}(X_{a'}) - m_0, \ldots
\]

These terms are known as the constant term, main effects, interactions of order two and so on. Let \( P \) be the set of all subsets of \( \{1, ..., d\} \) with dimension 1 to \( d \). For all \( v \in P \) and \( X \in X \), let \( X_v \) be the vector with components \( X_a, \ a \in v \). For a set \( A \) let \( |A| \) be its cardinality, and for all \( v \in P \), let \( m_v : \mathbb{R}^{|v|} \rightarrow \mathbb{R} \) be the function associated with \( X_v \) in Equation (2). Then Equation (2) can be expressed as follows:

\[
m(X) = m_0 + \sum_{v \in P} m_v(X_v).
\]

(3)

This decomposition is unique, all the terms \( m_v, \ v \in P \) are centered, and they are orthogonal with respect to \( L^2(X, P_X) \). The functions \( m \) and \( m_v, \ v \in P \) in Equation (3) are square-integrable. As any two terms of decomposition (3) are orthogonal, by squaring (3) and integrating it with respect to the distribution of \( X \), a decomposition of the variance of \( m(X) \) is obtained as follows:

\[
\text{var}(m(X)) = \sum_{v \in P} \text{var}(m_v(X_v)).
\]

(4)

The Sobol indices associated with the group of variables \( X_v, \ v \in P \) are defined by:

\[
S_v = \frac{\text{var}(m_v(X_v))}{\text{var}(m(X))}.
\]

(5)

For each \( v \), \( S_v \) expresses the fraction of variance of \( m(X) \) explained by \( X_v \). For all \( v \in P \), when \( |v| = 1 \), the \( S_a, S \) are referred to as the first order indices; when \( |v| = 2 \), i.e. \( v = \{a, a'\} \) and \( a \neq a' \), they are referred to as the second order indices or the interaction indices of order two (between \( X_a \) and \( X_{a'} \)); and the same holds for \( |v| > 2 \).

The total number of the Sobol indices to be calculated is equal to \( |P| = 2^d - 1 \), which raises exponentially with the number \( d \) of the input variables. When \( d \) is large, the evaluation of all the indices can be computationally demanding and even not reachable. In practice, only the indices of order not higher than two are calculated. However, only the first and second order indices may not provide a good information on the model sensitivities. In order to provide better information on the model sensitivities, \( \text{Homma and Saltelli 1996} \) proposed to calculate the first order and the total indices defined as follows:

Let \( P_a \subset P \) be the set of all the subsets of \( \{1, ..., d\} \) including \( a \), then \( S_{T_a} = \sum_{v \in P_a} S_v \). For all \( a \in \{1, ..., d\} \), \( S_{T_a} \) denotes the total effect of \( X_a \). It expresses the fraction of variance of \( m(X) \) explained by \( X_a \) alone and all the interactions of it with the other variables. The total indices allow us to rank the input variables with respect to the amount of their effect on the output variable. However, they do not provide complete information on the model sensitivities as do all the Sobol indices.

The classical computation of the Sobol indices is based on the Monte Carlo methods (see for example: \( \text{Sobol 1993} \) for the main effect and interaction indices, and \( \text{Saltelli 2002} \) for the main effect and total indices). For models that are expensive to evaluate, the Monte Carlo methods lead to a high computational burden. Moreover, in the case where \( d \) is large, \( m \) is complex and the calculation of the variances (see Equation (4)) is numerically complicated or not possible (as in the case where the model \( m \) is unknown) the methods described above are not applicable. Another approach consists in approximating \( m \) by a simplified model, called a meta-model, which is much faster to evaluate and to perform sensitivity analysis on it. Beside the approximations of the Sobol indices of \( m \) at a lower computational cost, a meta-model provides a deeper view of the input variables effects on the model output. Among the meta-modelling methods proposed in the literature, the expansion based on polynomial Chaos (\( \text{Wiener 1938} \), \( \text{Schoutens 2000} \)) can be used to approximate the Hoeffding decomposition of \( m \) (\( \text{Sudret 2008} \)).
project $m$ onto a basis of orthonormal polynomials. The polynomial Chaos expansion of $m$ is written as (Soize and Ghanem 2004):

$$ m(X) = \sum_{j=0}^{\infty} h_j \phi_j(X), \quad (6) $$

where $\{h_j\}_{j=0}^{\infty}$ are the coefficients, and $\{\phi_j\}_{j=0}^{\infty}$ are the multivariate orthonormal polynomials associated with $X$ which are determined according to the distribution of the components of $X$. In practice, expansion (6) shall be truncated for computational purposes, and the model $m$ may be approximated by $\sum_{j=0}^{v_{\text{max}}} h_j \phi_j(X)$, where $v_{\text{max}}$ is determined using a truncation scheme. The Sobol indices are obtained then by summing up the squares of the suitable coefficients. (Blatman and Sudret 2011) proposed a method for truncating the polynomial Chaos expansion and an algorithm based on the least angle regression for selecting the terms in the expansion. In this approach, according to the distribution of the components of $X$, a unique family of orthonormal polynomials $\{\phi_j\}_{j=0}^{\infty}$ is determined. However, this family may not necessarily be the best functional basis to approximate $m$ well.

Gaussian Process (GP) can also be used to construct meta-models as highlighted in (Welch, Buck, Sacks, Wynn, Mitchell, and Morris 1992), (Oakley and O’Hagan 2004), (Kleijnen 2007), (Kleijnen 2009), (Marrel, Iooss, Laurent, and Roustant 2009), (Durrande, Ginsbourger, Roustant, and Carraro 2013) considered a class of the functional approximation methods similar to the GP and obtained a meta-model that satisfies the properties of the Hoeffding decomposition. They proposed to approximate $m$ by functions belonging to a RKHS $\mathcal{H}$ which is a direct sum of the Hilbert spaces. Their RKHS $\mathcal{H}$ is constructed in a way that the projection of $m$ onto $\mathcal{H}$, denoted $f^{\ast}$, is an approximation of the Hoeffding decomposition of $m$. The function $f^{\ast}$ is defined as the minimizer over the functions $f \in \mathcal{H}$ of the criterion $E_X (m(X) - f(X))^2$.

Let $\langle \cdot, \cdot \rangle_\mathcal{H}$ be the scalar product in $\mathcal{H}$, let also $k$ and $k_v$ be the reproducing kernels associated with the RKHS $\mathcal{H}$ and the RKHS $\mathcal{H}_v$, respectively. The properties of the RKHS $\mathcal{H}$ insures that any function $f \in \mathcal{H}$, $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ is written as the following decomposition:

$$ f(X) = \langle f, k(\cdot, \cdot) \rangle_\mathcal{H} = f_0 + \sum_{v \in \mathcal{P}} f_v(X_v), \quad (7) $$

where $f_0$ is constant, and $f_v : \mathbb{R}^{|v|} \to \mathbb{R}$ is defined by $f_v(X_v) = \langle f, k_v(\cdot, \cdot) \rangle_\mathcal{H}$. For more details on the RKHS construction and the definition of the Hilbert norm see Section 6.1.1 in [6.1].

For all $v \in \mathcal{P}$, the functions $f_v(X_v)$ are centered and for all $v \neq v'$, the functions $f_v(X_v)$ and $f_{v'}(X_{v'})$ are orthogonal with respect to $L^2(\mathcal{X}, \mu_X)$. Therefore, the decomposition of the function $f$ presented in Equation (7) is its Hoeffding decomposition. As the function $f^{\ast}$ belongs to the RKHS $\mathcal{H}$, it is decomposed as its Hoeffding decomposition, $f^{\ast} = f_0^{\ast} + \sum_{v \in \mathcal{P}} f_v^{\ast}$, and each function $f_v^{\ast}$ approximates the function $m_v$ in Equation (3). The number of terms $f_v^{\ast}$ that should be estimated in the Hoeffding decomposition of $f^{\ast}$ is equal to $|\mathcal{P}| = 2^d - 1$, which may be huge since it rises very quickly by increasing $d$. In order to deal with this problem, in the regression framework, one may estimate $f^{\ast}$ by a sparse meta-model $\hat{f} \in \mathcal{H}$. To this end, the estimation of $f^{\ast}$ is done on the basis of $n$ observations by minimizing a least-squares criterion suitably penalized in order to deal with both the non-parametric nature of the problem and the possibly large number of functions that have to be estimated. In the classical framework of the sensitivity analysis one may calculate a sparse approximation of $f^{\ast}$ using least-squares penalized criterion as it is done in the non-parametric regression framework. In order to obtain a sparse solution of a minimization problem, the penalty function should enforce the sparsity. There exists various ways of enforcing sparsity for a minimization (maximization) problem, see for example (Hastie, Tibshirani, and Wainwright 2015) for a review. Some methods, such as the Sparse Additive Models (SpAM) procedure (Ravikumar, Lafferty, Liu, and Wasserman 2009), (Liu, Wasserman, and Lafferty 2009) are based on a combination of the $l_1$-norm with the empirical $L^2$-norm: $\|f\|_{l_1,n} = \sum_{a=1}^{d} \|f_a\|_n$, where $\|f_a\|_n^2 = \sum_{i=1}^{n} f_a^2(X_{ai})/n$, is the squared empirical $L^2$-norm of the univariate function $f_a$. The Component Selection and Smoothing Operator (COSSO) method developed by (Lin and Zhang 2006) enforces sparsity using a combination of the $l_1$-norm with the Hilbert norm:
\[ \| f \|_{H,1} = \sum_{a=1}^{d} \| f_{a} \|_{H_{a}}. \]

Instead of focusing on only one penalty term, one may consider a more general family of estimators, called a **doubly penalized estimator**, which is obtained by minimizing a criterion penalized by the sum of two penalty terms. (Raskutti, Wainwright, and Yu 2009; Raskutti, Wainwright, and Yu 2012) proposed a doubly penalized estimator, which is the solution of the minimization of a least-squares criterion penalized by the sum of a sparsity penalty term and a combination of the \(l_1\)-norm with the Hilbert norm:

\[ \gamma \| f \|_{n,1} + \mu \| f \|_{H,1}, \]

where \( \gamma, \mu \in \mathbb{R} \) are the tuning parameters that should be suitably chosen.

(Meier, van de Geer, and Bühlmann 2009) proposed a related family of estimators, based on the penalization with the empirical \(L^2\)-norm. Their penalty function is the sum of the sparsity penalty term, \(\| f \|_{n,1}\), and a smoothness penalty term. (Huet and Taupin 2017) considered the same approximation functional spaces as (Durrande, Ginsbourger, Roustant, and Carraro 2013), and obtained a doubly penalized estimator of a meta-model which approximates the Hoeffding decomposition of \(m\). Their estimator is the solution of the least-squares minimization penalized by the penalty function defined in Equation (8) adapted to the multivariate setting,

\[ \gamma \| f \|_{n} + \mu \| f \|_{H}, \text{ with } \| f \|_{n} = \sum_{v \in P} \| f_{v} \|_{n}, \| f \|_{H} = \sum_{v \in P} \| f_{v} \|_{H_{v}}. \]

This procedure, called RKHS ridge group sparse, estimates the groups \(v\) that are suitable for predicting \(f^*\), and the relationship between \(f_{v}^*\) and \(X_v\) for each group. The obtained estimator, called RKHS meta-model, is used then to estimate the Sobol indices of \(m\). This approach renders it possible to estimate the Sobol indices for all groups in the support of the RKHS meta-model, including the interactions of possibly high order, a point known to be difficult in practice.

In this paper, we introduce an R package, called RKHSMetaMod, that implements the RKHS ridge group sparse procedure. The functions of this package allows us to:

1. calculate the reproducing kernels and their associated Gram matrices (see Section 3.1),
2. implement the RKHS ridge group sparse procedure and a special case of it, called the RKHS group lasso procedure (when \(\gamma = 0\) in the penalty function (9)), in order to estimate the terms \(f_{v}^*\) in the Hoeffding decomposition of the meta-model \(f^*\) leading to an estimation of the function \(m\) (see Section 3.2),
3. choose the tuning parameters \(\mu\) and \(\gamma\) (see Equation (9)), using a procedure that leads to obtain the best RKHS meta-model in terms of the prediction quality,
4. estimate the Sobol indices of the function \(m\) (see Section 2.3).

The current version of the package supports uniformly distributed input variables on \([0,1]^d\). However, it could easily be adapted to a different distribution of input variables by making a small modification to one of its functions (see Remark 6 of Section 3.1).

Let us give a brief overview of the related existing statistical packages to RKHSMetaMod. The R package **sensitivity** is designed to implement sensitivity analysis methods and provides the approaches for numerical calculation of the Sobol indices. In particular, Kriging method can be used to reduce the number of observations in global sensitivity analysis. The function sobolGP of the package sensitivity builds a Kriging based meta-model using the function \(km\) of the package DiceKriging (Roustant, Ginsbourger, and Deville 2012), and estimates its Sobol indices. This procedure can also be done using the function \(km\) and the function fast99 of the package sensitivity (see Section 4.5. of Roustant, Ginsbourger, and Deville 2012). In this case, the idea is once again to build a Kriging based meta-model using the function \(km\) and then estimate its Sobol indices using the function fast99. In both cases the true function is substituted by a Kriging based meta-model and then its Sobol indices are estimated. In the sobolGP function, the Sobol indices are estimated by Monte Carlo integration, while the fast99 function estimates them using the extended-FAST method (Saltelli, Tarantola, and Chan 1999). To reduce the computational burden when dealing with large datasets and complex models, in RKHSMetaMod package, we propose to use the empirical variances to estimate the Sobol indices (see Section 2.3). Besides, the estimation of the Sobol indices in the RKHSMetaMod package is done based on the RKHS meta-model which is a sparse estimator. It is beneficial since instead of calculating the Sobol indices of all groups \(v \in P\), only the Sobol indices associated with the...
groups in the support of the RKHS meta-model are computed (see Section 2.3). Moreover, the functions sobolGP and fast99 provide the estimation of the first order and the total Sobol indices only, while the procedure in the RKHSMetaMod package makes it possible to estimate the high order Sobol indices. The R packages DiceKriging and DiceOptim (Deep Inside Computer Experiments Kriging/Optim) (Roustant, Ginsbourger, and Deville 2012) implement the Kriging based meta-models to estimate complex models in the high dimensional context. They provide different GP (Kriging) models corresponding to the Gaussian, Matérn, Exponential and Power-Exponential correlation functions. The estimation of the parameters of the correlation functions in these packages relies on the global optimizer with gradient genoud algorithm of the package rgenoud (Mebane and Sekhon 2011). These packages do not implement any method of the sensitivity analysis themselves. However, some authors (see Section 4.5. of Roustant, Ginsbourger, and Deville 2012) perform sensitivity analysis on their estimated meta-models by employing the functions of the package sensitivity. The R package RobustGaSP (Robust Gaussian Stochastic Process) (Gu, Palomo, and Berger 2019) approximates a complex model by a GP meta-model. This package implements marginal posterior mode estimation of the GP model parameters. The estimation method in this package insures the robustness of the parameter estimation in the GP model, and allows one also to identify input variables that have no effect on the variability of the function under study. The R package mlegp (maximum likelihood estimates of Gaussian processes) (Dancik and Dorman 2008) provides functions to implement both meta-modelling approaches and sensitivity analysis methods. It obtains maximum likelihood estimates of GP model for the output of costly computer experiments. The GP models are built either on the basis of the Gaussian correlation function or on the basis of the first degree polynomial trend. The sensitivity analysis methods implemented in this package include Functional Analysis of Variance (FANOVA) decomposition, plot functions to obtain diagnostic plots, main effects, and second order interactions. The prediction quality of the meta-model depends on the quality of the estimation of its parameters and more precisely the estimation of parameters in the correlation functions (Kennedy and O’Hagan 2000). The maximum likelihood estimation of these parameters often produce unstable results, and as a consequence, the obtained meta-model may have an inferior prediction quality ((Gu, Wang, and Berger 2018), (Gu 2019)). The RKHSMetaMod package is devoted to the meta-model estimation on the RKHS $\mathcal{H}$. It implements the convex optimization algorithms to calculate meta-models; provides the functions to compute the prediction error of the obtained meta-models; performs the sensitivity analysis on the obtained meta-models and more precisely calculate their Sobol indices. The convex optimization algorithms used in this package are all written using C++ libraries, and are adapted to take into account the problem of high dimensionality in this context. This package is available from the Comprehensive R Archive Network (CRAN) (Kamari 2019).

The organization of the paper is as follows: In the next Section, we describe the estimation method. In Section 3, we present in details the algorithms used in the RKHSMetaMod package to obtain the RKHS meta-model. In Section 6.2, an overview of the package functions as well as a brief documentation of them are given. Section 4 includes two parts: In the first part, Section 4.1, the performance of the RKHSMetaMod package functions is validated through a simulation study. In the second part, Section 4.2, the comparison in terms of the predictive accuracy between the RKHS meta-model and the Kriging based meta-models from RobustGaSP (Gu, Palomo, and Berger 2019) and DiceKriging (Roustant, Ginsbourger, and Deville 2012) packages is given through two examples.

2 Estimation method

In this Section, we present: the RKHS ridge group sparse and the RKHS group lasso procedures (see 2.1), the strategy of choosing the tuning parameters in the RKHS ridge group sparse algorithm (see 2.2), and the calculation of the empirical Sobol indices of the RKHS meta-model (see 2.3).

2.1 RKHS ridge group sparse and RKHS group lasso procedures

Let us denote by $n$ the number of observations. The dataset consists of a vector of $n$ observations $Y = (Y_1, ..., Y_n)$, and a $n \times d$ matrix of features $X$ with components $(X_{ai}, i = 1, ..., n, a = 1, ..., d) \in \mathbb{R}^{n \times d}$. For some tuning parameters $\gamma_v, \mu_v, v \in \mathcal{P}$, the RKHS ridge group sparse criterion is defined by,

$$\mathcal{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - f_0 - \sum_{v \in \mathcal{P}} f_v(X_{vi}) \right)^2 + \sum_{v \in \mathcal{P}} \gamma_v \|f_v\|_n + \sum_{v \in \mathcal{P}} \mu_v \|f_v\|_{\mathcal{H}_v}, \quad (10)$$
where $X_v$ represents the matrix of variables corresponding to the $v$-th group, i.e. $X_v = (X_{v,i}, i = 1, ..., n, v \in \mathcal{P}) \in \mathbb{R}^{n \times |\mathcal{P}|}$, and where $\|f_v\|_n$ is the empirical $L^2$-norm of $f_v$ defined by the sample $\{X_{v,i}\}_{i=1}^n$ as $\|f_v\|_n = \sqrt{\sum_{i=1}^n f_v^2(X_{v,i})/n}$.

The penalty function in the criterion (10) is the sum of the Hilbert norm and the empirical norm, which allows us to select few terms in the additive decomposition of $f$ over sets $v \in \mathcal{P}$. Moreover, the Hilbert norm favours the smoothness of the estimated $f_v$, $v \in \mathcal{P}$.

Let $\mathcal{F} = \{f : f = f_0 + \sum_{v \in \mathcal{P}} f_v, f_v \in \mathcal{H}_v, \|f_v\|_{\mathcal{H}_v} \leq r_v, r_v \in \mathbb{R}^+\}$ be the set of functions. Then the RKHS meta-model is defined by,

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \mathcal{L}(f).$$ (11)

According to the Representer Theorem (Kimeldorf and Wahba 1970), the non-parametric functional minimization problem described above is equivalent to a parametric minimization problem. Indeed, the solution of the minimization problem (11) belonging to the RKHS $\mathcal{H}$ is written as $f = f_0 + \sum_{v \in \mathcal{P}} f_v$, where for some matrix $\theta = (\theta_{vi}, i = 1, ..., n, v \in \mathcal{P}) \in \mathbb{R}^{n \times |\mathcal{P}|}$ we have for all $v \in \mathcal{P}$,

$$f_v(\cdot) = \sum_{i=1}^n \theta_{vi} k_v(X_{v,i}, \cdot), \text{ and } \|f_v\|^2_{\mathcal{H}_v} = \sum_{i,i'=1}^n \theta_{vi} \theta_{v'i'} k_v(X_{v,i}, X_{v'i'}).$$ (12)

Let $\|\cdot\|$ be the Euclidean norm (called also $L^2$-norm) in $\mathbb{R}^n$, and for each $v \in \mathcal{P}$, let $K_v$ be the $n \times n$ Gram matrix associated with the kernel $k_v(\cdot, \cdot)$, i.e. $(K_v)_{i,i'} = k_v(X_{v,i}, X_{v'i'})$. Let also $K^{1/2}$ be the matrix that satisfies $t(K^{1/2})K^{1/2} = K$, and let $\tilde{f}_0$ and $\tilde{\theta}$ be the minimizers of the following penalized least-squares criterion:

$$C(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v\|^2 + \sqrt{n} \sum_{v \in \mathcal{P}} \gamma_v \|K_v \theta_v\| + n \sum_{v \in \mathcal{P}} \mu_v \|K_v^{1/2} \theta_v\|.$$

Then the estimator $\hat{f}$ defined in Equation (11) satisfies,

$$\hat{f}(X) = \tilde{f}_0 + \sum_{v \in \mathcal{P}} \hat{f}_v(X_v) \text{ with } \hat{f}_v(X_v) = \sum_{i=1}^n \hat{\theta}_{vi} k_v(X_{v,i}, X_v).$$

**Remark 1.** The constraint $\|f_v\|_{\mathcal{H}_v} \leq r_v$ is crucial for theoretical properties, but the value of $r_v$ is generally unknown and has no practical usefulness. In this package, it is not taken into account in the parametric minimization problem.

For each $v \in \mathcal{P}$, let $\gamma'_v$ and $\mu'_v$ be the weights that are chosen suitably. We define $\gamma_v = \gamma \times \gamma'_v$ and $\mu_v = \mu \times \mu'_v$ with $\gamma, \mu \in \mathbb{R}^+$.  

**Remark 2.** This formulation simplifies the choice of the tuning parameters since instead of tuning $2 \times |\mathcal{P}|$ parameters $\gamma_v$ and $\mu_v$, $v \in \mathcal{P}$, only two parameters $\gamma$ and $\mu$ are tuned. Moreover, the weights $\gamma'_v$ and $\mu'_v$, $v \in \mathcal{P}$ may be of interest in practice. For example, one can take weights that increase with the cardinal of $v$ in order to favour the effects with small interaction order between variables.

For the sake of simplicity, in the rest of this paper for all $v \in \mathcal{P}$ the weights $\gamma'_v$ and $\mu'_v$ are assumed to be set as one, and the RKHS ridge group sparse criterion is then expressed as follows:

$$C(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v\|^2 + \sqrt{n} \sum_{v \in \mathcal{P}} \|K_v \theta_v\| + n \mu \sum_{v \in \mathcal{P}} \|K_v^{1/2} \theta_v\|. \quad (13)$$

If we consider only the second part of the penalty function in the criterion (13) (i.e. set $\gamma = 0$), we obtain the RKHS group lasso criterion as follows:

$$C_g(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v\|^2 + n \mu \sum_{v \in \mathcal{P}} \|K_v^{1/2} \theta_v\|, \quad (14)$$

which is a group lasso criterion (Yuan and Lin 2006) up to a scale transformation.

In the RKHSMetaMod package, the RKHS ridge group sparse algorithm is initialized using the solutions obtained by solving the RKHS group lasso algorithm. Indeed, the penalty function in the RKHS group lasso criterion (14) insures the sparsity in the solution. Therefore, for a given value of $\mu$, by implementing
From now on the tuning parameter in the RKHS group lasso criterion will be denoted by: 
\[ \hat{\mu} \]
Implementing RKHS group lasso algorithm will be denoted by 
\[ S_{\hat{f}_{\text{Group Lasso}}} \] and \[ \hat{\theta}_{\text{Group Lasso}} \], respectively. From now on the tuning parameter in the RKHS group lasso criterion will be denoted by: 
\[ \mu_g = \sqrt{n} \mu. \] (15)

2.2 Choice of the tuning parameters

While dealing with an optimization problem of a criterion of the form (13), one of the essential steps is to choose appropriately the tuning parameters. Cross-validation is generally used for that purpose. Nevertheless in the context of high-dimensional complex models, the computational time for a cross-validation procedure may be prohibitively high. Therefore we propose a procedure based on a single testing data set:

- we first choose, a grid of values of the tuning parameters \( \mu \) and \( \gamma \);
  
  Let \( \mu_{\text{max}} \) be the smallest value of \( \mu_g \) (see Equation (15)), such that the solution to the minimization of the RKHS group lasso problem for all \( v \in \mathcal{P} \) is \( \theta_v = 0 \). We have,
  
  \[ \mu_{\text{max}} = \max_v \left( 2 \lfloor K_v \rfloor / \sqrt{n} \right) / \sqrt{n}. \] (16)

  In order to set up the grid of values of \( \mu \), one may find \( \mu_{\text{max}} \) and then a grid of values of \( \mu \) could be defined by \( \mu_l = \mu_{\text{max}} / (\sqrt{n} \times 2^l) \) for \( l \in \{1, \ldots, l_{\text{max}}\} \). The grid of values of \( \gamma \) is chosen by the user.

- next, for the grid of values of \( \mu \) and \( \gamma \), we calculate a sequence of estimators. Each estimator associated with the pair \((\mu, \gamma)\) in the grid of values of \( \mu \) and \( \gamma \), denoted by \( \hat{f}(\mu, \gamma) \), is the solution of the RKHS ridge group sparse optimization problem or the RKHS group lasso optimization problem if \( \gamma = 0 \).

- finally, the obtained estimators \( \hat{f}(\mu, \gamma) \) are evaluated using a testing dataset, \( \{(Y_i^{\text{test}}, X_i^{\text{test}})\}_{i=1}^{n_{\text{test}}} \). The prediction error associated with the estimator \( \hat{f}(\mu, \gamma) \) is calculated by,
  
  \[ \text{ErrPred}(\mu, \gamma) = \sum_{i=1}^{n_{\text{test}}} (Y_i^{\text{test}} - \hat{f}(\mu, \gamma)(X_i^{\text{test}}))^2 / n_{\text{test}}, \]

  where for \( S_f \) being the support of the estimator \( \hat{f}(\mu, \gamma) \) we have,
  
  \[ \hat{f}(\mu, \gamma)(X^{\text{test}}) = \hat{f}_0 + \sum_{v \in S_f} \sum_{i=1}^{n} \theta_v k_v(X_{vi}, X_v^{\text{test}}). \]

  The pair \((\hat{\mu}, \hat{\gamma})\) with the smallest value of the prediction error is chosen, and the estimator \( \hat{f}(\hat{\mu}, \hat{\gamma}) \) is considered as the best estimator of the function \( m \), in terms of the prediction error.

In the \texttt{RKHSMetaMod} package, the algorithm to calculate a sequence of the RKHS meta-models, the value of \( \mu_{\text{max}} \), and the prediction error are implemented as \texttt{RKHSMetaMod}, \texttt{mu_max}, and \texttt{PredErr} functions, respectively. These functions are described in Section 6.2 and illustrated in Example 1, Example 2 and Examples 3,2,3 respectively.

2.3 Estimation of the Sobol indices

The variance of the function \( m \) is estimated by the variance of the estimator \( \hat{f} \). As the estimator \( \hat{f} \) belongs to the RKHS \( \mathcal{H} \), it admits the Hoeffding decomposition and,

\[ \text{var}(\hat{f}(X)) = \sum_{v \in \mathcal{P}} \text{var}(\hat{f}_v(X_v)), \text{ where } \forall v \in \mathcal{P}, \text{ var}(\hat{f}_v(X_v)) = E_X (\hat{f}_v^2(X_v)) = \| \hat{f}_v \|^2. \]

In order to reduce the computational cost in practice, one may estimate the variances of \( \hat{f}_v(X_v) \), \( v \in \mathcal{P} \) by their empirical variances. Let \( \hat{f}_v \) be the empirical mean of \( \hat{f}_v(X_{vi}), i = 1, \ldots, n \), then:

\[ \hat{\text{var}}(\hat{f}_v(X_v)) = \frac{1}{n-1} \sum_{i=1}^{n} (\hat{f}_v(X_{vi}) - \hat{f}_v)^2. \]
For the groups $v$ that do not belong to the support of $\hat{f}$, we have $\hat{S}_v = 0$ and for the groups $v$ that belong to the support of $\hat{f}$, the estimators of the Sobol indices of $m$ are defined by:

$$\hat{S}_v = \frac{\text{var}(\hat{f}_v(X_v))}{\sum_{v \in P} \text{var}(\hat{f}_v(X_v))}.$$ 

In the RKHSMetaMod package, the algorithm to calculate the empirical Sobol indices $\hat{S}_v, v \in P$ is implemented as SI_emp function. This function is described in Section 6.2.2 and illustrated in Examples 1, 2, 3.

3 Algorithms

The RKHSMetaMod package implements two optimization algorithms: the RKHS ridge group sparse (see Algorithm 2) and the RKHS group lasso (see Algorithm 1). These algorithms rely on the Gram matrices $K_v, v \in P$ that have to be positive definite. Therefore, the first and essential step in this package is to calculate these matrices and insure their positive definiteness. This step is detailed in an algorithm that is described in the next Section. The second step is to estimate the RKHS meta-model. In the RKHSMetaMod package, two different objectives based on different procedures are considered to calculate this estimator:

1. The RKHS meta-model with the best prediction quality.

   The procedure to calculate the RKHS meta-model with the best prediction quality has been described in Section 2.2. A sequence of values of the tuning parameters $(\mu, \gamma)$ is considered, and the RKHS meta-models associated with each pair of the values of $(\mu, \gamma)$ are calculated. For $\gamma = 0$, the RKHS meta-model is obtained by solving the RKHS group lasso optimization problem, while for $\gamma \neq 0$ the RKHS ridge group sparse optimization problem is solved to calculate the RKHS meta-model. The obtained RKHS meta-models are evaluated by considering a new dataset and the RKHS meta-model with the minimum value of the prediction error is chosen as the best estimator.

2. The RKHS meta-model with at most $q_{max}$ groups in its support, i.e. $|S_f| \leq q_{max}$.

   First, the tuning parameter $\gamma$ is set as zero. Then, a value of $\mu$ for which the number of groups $v \in P$ in the solution of the RKHS group lasso optimization problem is equal to $q_{max}$, is computed. This value of $\mu$ will be denoted by $\mu_{q_{max}}$. Finally, the RKHS meta-models containing at most $q_{max}$ groups in their support are obtained by implementing the RKHS ridge group sparse algorithm for a grid of values of $\gamma \neq 0$ and $\mu_{q_{max}}$. This procedure is described in more details in Section 3.2.3.

3.1 Calculation of the Gram matrices

The available kernels in the RKHSMetaMod package are: Gaussian kernel, Matérn 3/2 kernel, Brownian kernel, quadratic kernel and linear kernel. The usual presentation of these kernels is given in Table 1. The choice of the kernel that is done by the user determines the functional approximation space. For a chosen kernel, the algorithm to calculate the Gram matrices $K_v, v \in P$ in the RKHSMetaMod package, is implemented as calc_Kv function. This algorithm is based on three essential points:

1. Modify the chosen kernel:

   In order to satisfy the conditions of constructing the RKHS $\mathcal{H}$ described in Section 6.1.1 these kernels

| Kernel type | Mathematics formula for $u \in \mathbb{R}^n, v \in \mathbb{R}$ | RKHSMetaMod name |
|-------------|--------------------------------------------------|------------------|
| Gaussian    | $k_\alpha(u, v) = \exp(-\|u - v\|^2/2\alpha^2)$ | "gaussian"       |
| Matérn 3/2  | $k_\alpha(u, v) = (1 + \sqrt{3}\|u - v\|/\alpha)\exp(-\sqrt{3}\|u - v\|/\alpha)$ | "matern"         |
| Brownian    | $k_\alpha(u, v) = \min(u, v) + 1$               | "brownian"       |
| Quadratic   | $k_\alpha(u, v) = (u^Tv + 1)^2$                | "quad"           |
| Linear      | $k_\alpha(u, v) = u^Tv + 1$                    | "linear"         |

Table 1: List of the reproducing kernels used to construct the RKHS $\mathcal{H}$. The range parameters $\alpha$ in the Gaussian and Matérn 3/2 kernels are assumed to be fixed and set as $1/2$ and $\sqrt{3}/2$, respectively. The value 1 is added to the Brownian kernel to relax the constraint of nullity at the origin.
are modified according to Equation [19]. Let us take the example of the Brownian kernel:

The RKHS associated with the Brownian kernel \( k_a(X_a, X'_a) = \min(X_a, X'_a) + 1 \) is well known to be \( \mathcal{H}_a = \{ f : [0, 1] \to \mathbb{R} \text{ is absolutely continuous, and } f(0) = 0, \int_0^1 f(X_a)^2 dX_a < \infty \} \), with the inner product \( \langle f, h \rangle_{\mathcal{H}_a} = \int_0^1 f(X_a)h'(X_a)dX_a \). Easy calculations lead to obtain the Brownian kernel as follows,

\[
k_{0a} = \min(X_a, X'_a) + 1 - (3/4)(1 + X_a - X_a^2/2)(1 + X'_a - X'_a^2/2).
\]

The RKHS associated with the kernel \( k_{0a} \) is the set \( \mathcal{H}_{0a} = \{ f \in \mathcal{H}_a : \int_0^1 f(X_a)dX_a = 0 \} \), and we have \( \mathcal{H} = 1 + \sum_{a \in \mathcal{P}} \mathcal{H}_a = \{ f : [0, 1]^d \to \mathbb{R} : f = f_0 + \sum_{a \in \mathcal{P}} f_a(X_a), \text{ with } f_a \in \mathcal{H}_a \} \).

**Remark 3.** In the current version of the package, we consider the input variables \( X = (X_1, ..., X_d) \) that are uniformly distributed on \([0, 1]^d\). In order to consider the input variables that are not distributed uniformly, it suffices to modify a part of the function \( \text{calc}_Kv \) related to the calculation of the kernels \( k_{0a}, \ a = 1, ..., d \). For example, for \( X = (X_1, ..., X_d) \) being distributed with law \( P_X = \prod_{a=1}^d P_a \) on \( \mathcal{X} = \bigotimes_{a=1}^d X_a \subset \mathbb{R}^d \), the kernel \( k_{0a} \) associated with the Brownian kernel is calculated as follows,

\[
k_{0a} = \min(X_a, X'_a) + 1 - \frac{\int_{X_a} (\min(X_a, U) + 1)dP_a(\int_{X_a} (\min(X_a, U) + 1)dP_a)}{(\int_{X_a} \min(U, V) + 1)dP_a dP_a}.
\]

The other parts of the function \( \text{calc}_Kv \) remain unchanged.

(2) Calculate the Gram matrices \( K_v \) for all \( v \):

First, for all \( a = 1, ..., d \), the Gram matrices \( K_a \) associated with kernels \( k_{0a} \) are calculated using Equation [19], \( (K_a)_{i,i'} = k_{0a}(X_{ai}, X_{ai'}) \). Then, for all \( v \in \mathcal{P} \), the Gram matrices \( K_v \) associated with kernel \( k_v = \prod_{a \in v} k_{0a} \) are computed by \( K_v = \bigcap_{a \in v} K_a \).

(3) Insure the positive definiteness of the matrices \( K_v \):

The output of the function \( \text{calc}_Kv \) is one of the input arguments of the functions associated with the RKHS group lasso and the RKHS ridge group sparse algorithms. Throughout these algorithms we need to calculate the inverse and the square root of the matrices \( K_v \). In order to avoid the numerical problems and insure the invertibility of the matrices \( K_v \), it is mandatory to have these matrices positive definite. One way to render the matrices \( K_v \) positive definite is to add a nugget effect to them. That is, to modify matrices \( K_v \) by adding a diagonal with a constant term, i.e. \( K_v + \epsilon \times I_n \). The value of \( \epsilon \) is computed based on the data and through a part of the algorithm of the function \( \text{calc}_Kv \). Let us briefly explain this part of the algorithm:

For each group \( v \in \mathcal{P} \), let \( \lambda_{v,i}, i = 1, ..., n \) be the eigenvalues associated with the matrix \( K_v \). Set \( \lambda_{v,\max} = \max_i \lambda_{v,i} \) and \( \lambda_{v,\min} = \min_i \lambda_{v,i} \). For some fixed value of tolerance "tol", and for each matrix \( K_v \), if \( \lambda_{v,\min} < \lambda_{v,\max} \times \text{tol} \) then, the eigenvalues of \( K_v \) are replaced by \( \lambda_{v,i} + \epsilon \), with "epsilon" being equal to \( \lambda_{v,\max} \times \text{tol} \). The value of "tol" is set as \( 1e^{-8} \) by default, but one may consider a smaller or a greater value for it depending on the kernel chosen and the value of \( n \).

The function \( \text{calc}_Kv \) is described in Section 6.2.2 and illustrated in Example 2.

### 3.2 Optimization algorithms

The RKHS meta-model is the solution of one of the optimization problems: the minimization of the RKHS group lasso criterion presented in Equation [13] (if \( \gamma = 0 \)), or the minimization of the RKHS ridge group sparse criterion presented in Equation [13] (if \( \gamma \neq 0 \)). In the following, the algorithms to solve these optimization problems are presented.

#### 3.2.1 RKHS group lasso

A popular technique for doing group wise variable selection is group lasso. With this procedure, depending on the value of the tuning parameter \( \mu \), an entire group of predictors may drop out of the model. An efficient algorithm for solving group lasso problem is the classical block coordinate descent algorithm ([Boyd, Parikh, Chu, Peleato, and Eckstein 2011], [Bubeck 2015]). Following the idea of [Fu 1998], [Yuan and Lin 2006] implemented a block wise descent algorithm for the group lasso penalized least-squares under the condition...
that the model matrices in each group are orthonormal. A block coordinate (gradient) descent algorithm for solving the group lasso penalized logistic regression is then developed by [Meier, van de Geer, and Bühlmann 2008]. This algorithm is implemented in the R package **grplasso** available from CRAN [Meier 2020]. [Yang and Zou 2015] proposed a unified algorithm named group wise majorization descent for solving the general group lasso learning problems by assuming that the loss function satisfies a quadratic majorization condition. The implementation of their work is done in the **ggllasso** R package available from CRAN [Yang, Zou, and Bhatnagar 2020].

In order to solve the RKHS group lasso optimization problem, we use the classical block coordinate descent algorithm. The minimization of criterion $C_g(f_0, \theta)$ (see Equation (14)) is done along each group $v$ at a time. At each step of the algorithm, the criterion $C_g(f_0, \theta)$ is minimized as a function of the current block's parameters, while the parameters values for the other blocks are fixed to their current values. The procedure is repeated until convergence. This procedure leads to Algorithm 1 (see 6.1 for more details on this procedure). In the **RKHSMetaMod** package, the Algorithm 1 is implemented as **RKHSgrplasso** function.

**Algorithm 1 RKHS group lasso algorithm:**

1. Set $\theta_0 = [0]_{|P| \times n}$
2. repeat
3. Calculate $f_0 = \arg\min_{f_0} C_g(f_0, \theta)$
4. for $v \in P$ do
5. Calculate $R_v = Y - f_0 - \sum_{v \neq w} K_w \theta_w$
6. if $\|\frac{2}{\sqrt{n}} \| K_v^{1/2} R_v \| \leq \mu_g$ then
7. $\theta_v \leftarrow 0$
8. else
9. $\theta_v \leftarrow \arg\min_{\theta_v} C_g(f_0, \theta)$
10. end if
11. end for
12. until convergence

This function is described in Section 6.2.2.

### 3.2.2 RKHS ridge group sparse

In order to solve the RKHS ridge group sparse optimization problem, we propose an adapted block coordinate descent algorithm. This algorithm is provided in two steps:

**Step 1** Initialize the input parameters by the solutions of the RKHS group lasso algorithm for each value of the tuning parameter $\mu$, and implement the RKHS ridge group sparse algorithm through the active support of the RKHS group lasso solutions until it achieves convergence. This step is provided in order to decrease the execution time. In fact, instead of implementing the RKHS ridge group sparse algorithm over the set of all groups $P$, it is implemented only over the groups in the support of the solution of the RKHS group lasso algorithm, $\hat{S}_{\text{Group Lasso}}$.

**Step 2** Re-initialize the input parameters with the obtained solutions of **Step 1** and implement the RKHS ridge group sparse algorithm through all groups in $P$ until it achieves convergence. This second step makes it possible to verify that no group is missing in the output of **Step 1**.

This procedure leads to Algorithm 2 (see 6.1 for more details on this procedure). In the **RKHSMetaMod** package the Algorithm 2 is implemented as **pen_MetMod** function. This function is described in Section 6.2.2 and illustrated in Example 2.

### 3.2.3 RKHS meta-model with at most $q_{max}$ groups in its support

By considering some prior information about the data, one may be interested in a RKHS meta-model $\hat{f}$ with the number of groups in its support not greater than some "$q_{max}$". In order to obtain such an estimator, we provide the following procedure in the **RKHSMetaMod** package:

- First, the tuning parameter $\gamma$ is set as zero and a value of $\mu$ for which the solution of the RKHS group lasso algorithm, Algorithm 1 contains exactly $q_{max}$ groups in its support is computed. This value is denoted by $\mu_{q_{max}}$. 

This function is described in Section 6.2.2.
Algorithm 2 RKHS ridge group sparse algorithm:

1: **Step 1:**
2: Set $\theta_0 = \hat{\theta}_{\text{Group Lasso}}$ and $\mathcal{P} = \hat{S}_{f_{\text{Group Lasso}}}$
3: repeat
4: Calculate $f_0 = \arg\min_{f_0} C(f_0, \theta)$
5: for $v \in \mathcal{P}$ do
6: Calculate $R_v = Y - f_0 - \sum_{v \neq w} K_w \theta_w$
7: Solve $J^* = \arg\min_{J \in \mathbb{R}^n} \{ J(\hat{\theta}_v), \text{such that } \| K_v^{-1/2} \hat{\theta}_v \| \leq 1 \}$
8: if $J^* \leq \gamma$ then
9: $\theta_v \leftarrow 0$
10: else
11: $\theta_v \leftarrow \arg\min_{\theta_v} C(f_0, \theta)$
12: end if
13: end for
14: until convergence
15: **Step 2:**
16: Implement the same procedure as **Step 1** with $\theta_0 = \hat{\theta}_{\text{old}}, \mathcal{P} = \mathcal{P} \triangledown \hat{\theta}_{\text{old}}$

Then, the RKHS ridge group sparse algorithm, Algorithm 2, is implemented by setting the tuning parameter $\mu$ equal to $\mu_{q_{\text{max}}}$ and a grid of values of the tuning parameter $\gamma > 0$.

This procedure leads to Algorithm 3. This algorithm is implemented in the RHKSMetaMod package, as

Algorithm 3 Algorithm to estimate RKHS meta-model with at most $q_{\text{max}}$ groups in its support:

1: Calculate $\mu_{\text{max}} = \max_v \frac{1}{\sqrt{n}} \| K_v^{1/2}(Y - \bar{Y}) \|$  
2: Set $\mu_1 = \mu_{\text{max}}$ and $\mu_2 = \frac{\mu_{\text{max}}}{\text{"rat"}}$  
3: repeat
4: Implement RKHS group lasso algorithm, Algorithm 1 with $\mu_i = \frac{\mu_1 + \mu_2}{2}$  
5: Set $q = |\hat{S}_{f_{\text{Group Lasso}}}|$
6: if $q > q_{\text{max}}$ then
7: Set $\mu_1 = \mu_1$ and $\mu_2 = \mu_i$
8: else
9: Set $\mu_1 = \mu_i$ and $\mu_2 = \mu_2$
10: end if
11: until $q = q_{\text{max}}$ or $i > \text{Num}$  
12: Implement RKHS ridge group sparse algorithm, Algorithm 2 with $(\mu = \mu_{q_{\text{max}}}, \gamma > 0)$

function RHKSmetMod_qmax (see Section "Main RHKSMetaMod functions" (supplementary materials) for more details on this function).

**Remark 4.** As both terms in the penalty function of criterion (13) enforce sparsity to the solution, the estimator obtained by solving the RKHS ridge group sparse associated with the pair of the tuning parameters $(\mu_{q_{\text{max}}}, \gamma > 0)$ may contain a smaller number of groups than the solution of the RKHS group lasso optimization problem (i.e. the RKHS ridge group sparse with $(\mu_{q_{\text{max}}}, \gamma = 0)$). And therefore, the estimated RKHS meta-model contains at most "$q_{\text{max}}"$ groups in its support.

4 RKHSMetaMod through examples

4.1 Simulation study

Let us consider the g-function of Sobol (Saltelli, Chan, and Scott 2009) in the Gaussian regression framework, i.e. $Y = m(X) + \varepsilon$. The error term $\varepsilon$ is a centered Gaussian random variable with variance $\sigma^2$, and
the function \( m \) is the g-function of Sobol defined over \([0, 1]^d\) by,

\[
m(X) = \prod_{a=1}^{d} \frac{|4X_a - 2| + c_a}{1 + c_a}, \quad c_a > 0.
\]

(17)

The Sobol indices of the g-function can be expressed analytically:

\[
\forall v \in \mathcal{P}, \quad S_v = \frac{1}{D} \prod_{a \in v} D_a, \quad D_a = \frac{1}{3(1 + c_a)^2}, \quad D = \prod_{a=1}^{d} (D_a + 1) - 1.
\]

Set \( c_1 = 0.2, c_2 = 0.6, c_3 = 0.8 \) and \( (c_a)_{a>3} = 100 \). With these values of coefficients \( c_a \), the variables \( X_1, X_2 \) and \( X_3 \) explain 99.98% of the variance of the function \( m(X) \) (see Table 4).

In this Section, three examples are presented. In all examples, the value of \( \text{Dmax} \) is set as three. Example 1 illustrates the use of the \texttt{RKHSMetMod} function as follows:

\[
\text{R> res} \leftarrow \text{RKHSMetMod}(Y, X, \text{kernel}, \text{Dmax}, \gamma, \text{frc}, \text{FALSE})
\]

For each dataset, the experiment is repeated \( N_r = 50 \) times. At each repetition, the RKHS meta-models associated with the pair of the tuning parameters \( (\mu, \gamma) \) are estimated using the \texttt{RKHSMetMod} function:

\[
\text{R> gamma} \leftarrow c(0.2, 0.1, 0.01, 0.005, 0); \quad \text{frc} \leftarrow c(4, 8, 16, 32, 64)
\]

Example 1. RKHS meta-model estimation using \texttt{RKHSMetMod} function:

In this example, three datasets of \( n \) points \texttt{maximinLHS} over \([0, 1]^d\) are generated with \( n \in \{50, 100, 200\} \) and \( d = 5 \), and a grid of five values for each of the tuning parameters \( \mu \) and \( \gamma \) is considered as follows:

\[
\mu_{(1:5)} = \mu_{\text{max}}/(\sqrt{n} \times 2^{(2:6)}), \quad \gamma_{(1:5)} = (0.2, 0.1, 0.01, 0.005, 0).
\]

For each dataset, the experiment is repeated \( N_r = 50 \) times. At each repetition, the RKHS meta-models associated with the pair of the tuning parameters \( (\mu, \gamma) \) are estimated using the \texttt{RKHSMetMod} function:

\[
\text{R> Dmax} \leftarrow 3; \quad \text{kernel} \leftarrow \text{"matern"} \# \quad \text{kernel} \leftarrow \text{"brownian"} \# \quad \text{kernel} \leftarrow \text{"gaussian"}
\]

The performances of this method for estimating a meta-model are evaluated by considering a third dataset \((m(X_{i}^{\text{third}}), X_{i}^{\text{third}}), i = 1, \ldots, N\), with \( N = 1000 \). The global prediction error is calculated as follows:
Let \( \hat{f}_r(.) \) be the best RKHS meta-model obtained in the repetition \( r, r = 1, \ldots, N_r \), then

\[
GPE = \frac{1}{N_r} \sum_{r=1}^{N_r} \frac{1}{N} \sum_{i=1}^{N} \left( \hat{f}_r(X_i^{\text{third}}) - \hat{m}(X_i^{\text{third}}) \right)^2.
\]

The values of \( GPE \) obtained for different kernels and values of \( n \) are given in Table 2. As expected the value of \( GPE \) decreases as \( n \) increases. The lowest values of \( GPE \) are obtained when using the "matern" kernel.

In order to sum up the behaviour of the procedure for estimating the Sobol indices, we consider the mean square error (MSE) criterion obtained by \( \sum_v (\sum_{r=1}^{N_r} (\hat{S}_{v,r} - S_v)^2 / N_r) \), where for each group \( v \), \( S_v \) denotes the true values of the Sobol indices, and \( \hat{S}_{v,r} \) is the empirical Sobol indices of the best RKHS meta-model in repetition \( r \).

The obtained values of MSE for different kernels and values of \( n \) are given in Table 3. As expected, the values of MSE are smaller for larger values of \( n \). The smallest values are obtained when using "matern" kernel.

The means of the empirical Sobol indices of the best RKHS meta-models through all repetitions for \( n = 200 \) and "matern" kernel are displayed in Table 4. It appears that the estimated Sobol indices are close to the true ones, nevertheless, they are overestimated for the main effects, i.e. groups \( v \in \{ \{1\}, \{2\}, \{3\} \} \), and underestimated for the interactions of order two and three, i.e. groups \( v \in \{ \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\} \} \).

Note that the strategy of choosing the tuning parameters is based on the minimization of the prediction error of the estimated meta-model, which may not minimize the error of estimating the Sobol indices.

Taking into account the results obtained for this Example 1, the calculations in the rest of the examples is done using only the "matern" kernel.

**Example 2.** A time-efficient strategy to obtain the "optimal" tuning parameters when dealing with large data sets:

A dataset of \( n \) points \texttt{maximinLHS} over \([0,1]^d\) with \( n = 1000 \) and \( d = 10 \) is generated. First, we use functions \texttt{calcKv} and \texttt{mu_max} to compute the eigenvalues and eigenvectors of the positive definite matrices.
*K_v*, and the value of *µ_max*, respectively:

```r
R> kernel <- "matern"; Dmax <- 3
R> Kv <- calc_Kv(X, kernel, Dmax, TRUE, TRUE)
R> mumax <- mu_max(Y, Kv$kv)
```

Then, we consider the two following steps:

1. Set \( \gamma = 0 \) and \( \mu_g = \mu \times \sqrt{n} \) by using the function `RKHSgrplasso`. Gather the obtained RKHS meta-models in a list, "res_g" (while this job could be done with the function `RKHSMetMod` by setting \( \gamma = 0 \), in this example, we use the function `RKHSgrplasso` in order to avoid the re-calculation of \( K_v \)'s at the next step). Thereafter, for each estimator in res_g, the prediction error is calculated by considering a new dataset and using the function `PredErr`. The value of \( \mu \) with the smallest error of prediction in this step is denoted by \( \mu_i \). Let us implement this step:

   ```r
   frc <- c(4,8,16,32,64,128,256,512,1024)
   mu_g <- mumax/frc
   res_g <- list(); resg <- list()
   for(i in 1:length(mu_g))
     resg[[i]] <- RKHSgrplasso(Y,Kv, mu_g[i],1000,FALSE)
   res_g[[i]] <- list("mu_g"=mu_g,"gamma"=0,"MetaModel"=resg[[i]])
   ```

   Output res_g contains nine RKHS meta-models and they are evaluated using a testing dataset:
   ```r
   gamma <- c(0); Err_g <- PredErr(X, XT, YT, mu_g, gamma, res_g, kernel, Dmax)
   ```

   The prediction errors of the RKHS meta-models obtained in this step are displayed in Table 5. It appears that the minimum prediction error corresponds to the solution of the RKHS group lasso algorithm with \( \mu_g = 0.041 \), so \( \mu_i = 0.041/\sqrt{n} \).

2. Choose a smaller grid of values of \( \mu, (\mu_{i-1}, \mu_i, \mu_{i+1}) \), and set a grid of values of \( \gamma > 0 \). Calculate the RKHS meta-models associated with each pair of the tuning parameters \( (\mu, \gamma) \) by the function `pen_MetMod`. Calculate the prediction errors for the new sequence of the RKHS meta-models using the function `PredErr`. Compute the empirical Sobol indices for the best estimator. Let us go back to the implementation of the example and apply this step 2:

   ```r
   gamma <- c(0.2, 0.1, 0.01, 0.005); mu <- c(mu_g[5], mu_g[6], mu_g[7])/sqrt(n)
   resgnew <- list()
   resgnew[[1]] <- resg[[5]]; resgnew[[2]] <- resg[[6]]; resgnew[[3]] <- resg[[7]]
   res <- pen_MetMod(Y, Kv, gamma, mu, resgnew, 0, 0)
   ```

   The output "res" is a list of twelve RKHS meta-models. These meta-models are evaluated using a new dataset, and their prediction errors are displayed in Table 6.

| \( \mu_g \) | 0.081/\( \sqrt{n} \) | 0.041/\( \sqrt{n} \) | 0.020/\( \sqrt{n} \) |
| --- | --- | --- | --- |
| \( \gamma = 0 \) | 0.153 | 0.131 | 0.119 |
| \( \gamma = 0.1 \) | 0.098 | 0.079 | 0.072 |
| \( \gamma = 0.01 \) | 0.065 | 0.054 | 0.053 |
| \( \gamma = 0.005 \) | 0.064 | 0.054 | 0.054 |

Table 6: Example 2 Obtained prediction errors in step 2.
The minimum prediction error is associated with the pair \((0.020/\sqrt{n}, 0.01)\), and the best RKHS meta-model is then \(\hat{f}_{(0.020/\sqrt{n},0.01)}\).

The performance of this procedure for estimating the Sobol indices is evaluated using the relative error (RE) defined as follows:

For each \(v\), let \(S_v\) be the true value of the Sobol indices displayed in Table 4 and \(\hat{S}_v\) be the estimated empirical Sobol indices. Then

\[
RE = \sum_v |\hat{S}_v - S_v|/S_v.
\]  

(18)

In Table 7 the estimated empirical Sobol indices, their sum, and the value of RE are displayed.

| \(v\) | \{1\} | \{2\} | \{3\} | \{1, 2\} | \{1, 3\} | \{2, 3\} | \{1, 2, 3\} | sum  | RE |
|-----|------|------|------|--------|--------|--------|--------|------|-----|
| \(S_v\) | 42.91 | 25.50 | 20.81 | 4.40   | 3.84   | 2.13   | 0.00   | 99.60| 1.64|

Table 7: Example 2. The estimated empirical Sobol indices \(\times 100\) greater than \(10^{-2}\). The last two columns show \(\sum_v \hat{S}_v\) and RE, respectively.

The obtained RE for each group \(v\) is smaller than 1.64\%, therefore, the estimated Sobol indices in this example are very close to the true values of the Sobol indices displayed in the first row of Table 4.

**Example 3. Dealing with larger datasets:**

Two datasets of \(n\) points \(\text{mininLHS}\) over \([0,1]^d\) with \(n \in \{2000,5000\}\) and \(d = 10\) are generated. In order to obtain one RKHS meta-model associated with one pair of the tuning parameters \((\mu, \gamma)\), the number of coefficients to be estimated is equal to \(n \times \text{vMax} = n \times 175\). Table 8 gives the execution time for different functions used throughout the Examples 1-3. In all examples we used a cluster of computers with: 2 Intel Xeon E5-2690 processors (2.90GHz) and 96Gb Ram (6x16Gb of memory 1600MHz). As we can see, the execution time increases fast as \(n\) increases. In Figure 1 the plot of the logarithm of the time (in seconds) versus the logarithm of \(n\) is displayed for the functions calc_Kv, mu_max, RKHSgrplasso and pen_MetMod. It appears that, the algorithms of these functions are of polynomial time \(O(n^\alpha)\) with \(\alpha \simeq 3\) for the functions calc_Kv and mu_max, and \(\alpha \simeq 2\) for the functions RKHSgrplasso and pen_MetMod.

Taking into account the results obtained for the prediction error and the values of \((\hat{\mu}, \hat{\gamma})\) in Example 2 in this example, only two values of the tuning parameter \(\mu_{(1:12)} = \mu_{\text{max}}/(\sqrt{n} \times 2^7\gamma)\), and one value of the tuning parameter \(\gamma = 0.01\) are considered. The RKHS meta-models associated with the pair of values \((\mu_i, \gamma)\), \(i = 1, 2\) are estimated using the RKHSMetMod function:

```r
R> kernel <- "matern"; Dmax <- 3
R> gamma <- c(0.01); frc <- c(128,256)
R> res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
```

The prediction error and the empirical Sobol indices are then calculated for the obtained meta-models using
Figure 1: Example 3: Timing plot for $d = 10$, $n \in \{100, 300, 500, 1000, 2000, 5000\}$, and different functions of the RKHSMetaMod package. The logarithm of the execution time (in seconds) for the functions RKHSgrplasso and pen_MetMod is displayed for two pairs of values of tuning parameters $(\mu_1 = \mu_{\text{max}}/(\sqrt{n} \times 2^7), \gamma = 0.01)$ in solid lines, and $(\mu_2 = \mu_{\text{max}}/(\sqrt{n} \times 2^8), \gamma = 0.01)$ in dashed lines.

the functions PredErr and SI_emp:
R> mu <- vector(); mu[1] <- res[[1]]$mu; mu[2] <- res[[2]]$mu
R> Err <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
R> SI <- SI_emp(res, NULL)
The result of the prediction errors associated with the obtained estimators for two different values of $n$ are displayed in Table 9. For $n$ equal to 5000, we obtained smaller values of the prediction error, so as expected, the prediction quality improves by increasing the number of the observations $n$. Table 10 gives the estimated empirical Sobol indices as well as their sum and the values of RE (see Equation (18)). Comparing the values of RE, we can see that the empirical Sobol indices are better estimated for $n$ equal to 5000, so as expected, the estimation of the Sobol indices is better for larger values of $n$.

| $n$    | $v$ | $\{1\}$ | $\{2\}$ | $\{3\}$ | $\{1, 2\}$ | $\{1, 3\}$ | $\{2, 3\}$ | $\{1, 2, 3\}$ | sum  | RE  |
|-------|-----|---------|---------|---------|-----------|-----------|-----------|--------------|------|-----|
| 2000  | $S_{v}(\mu_1, \gamma)$ | 45.54 | 24.78 | 21.01 | 3.96 | 3.03 | 1.65 | 0.00 | 99.97 | 2.12 |
|       | $S_{v}(\mu_2, \gamma)$ | 45.38 | 25.07 | 19.69 | 4.36 | 3.66 | 1.79 | 0.00 | 99.95 | 1.79 |
| 5000  | $S_{v}(\mu_1, \gamma)$ | 44.77 | 25.39 | 20.05 | 4.49 | 3.38 | 1.90 | 0.00 | 99.98 | 1.81 |
|       | $S_{v}(\mu_2, \gamma)$ | 43.78 | 24.99 | 19.56 | 5.43 | 3.90 | 2.32 | 0.00 | 99.98 | 1.29 |

Table 10: Example 3 The estimated empirical Sobol indices $\times 100$ greater than $10^{-2}$ associated with each estimated RKHS meta-model is printed. The last two columns show $\sum_v S_v$ and RE, respectively. We have $\mu_1 = \mu_{\text{max}}/(\sqrt{n} \times 2^7)$, $\mu_2 = \mu_{\text{max}}/(\sqrt{n} \times 2^8)$ and $\gamma = 0.01$. Of RE, we can see that the empirical Sobol indices are better estimated for $n$ equal to 5000, so as expected, the estimation of the Sobol indices is better for larger values of $n$.
4.2 Comparison examples

This section includes two examples. In the first example we reproduce an example from paper (Gu, Palomo, and Berger 2019) and compare the prediction quality of the RKHS meta-model with the GP (Kriging) based meta-models from the RobustGaSP (Gu, Palomo, and Berger 2019) and DiceKriging (Roustant, Ginsbourger, and Deville 2012) packages.

The objective is to evaluate the quality of the RKHS estimated meta-model and to compare it with methods recently proposed for approximating complex models. In the first example we consider one-dimensional model and focus on the comparison between the true model and the estimated meta-model. In the second example we reproduce an example from paper (Roustant, Ginsbourger, and Deville 2012) which allows us to compare the prediction quality of the RKHS meta-model with the Kriging based meta-model from DiceKriging package, as well as the estimation quality of the Sobol indices in our package with the well-known package sensitivity.

For the sake of comparison between the three methods, the meta-models are calculated using the same experimental design and outputs and the same kernel function available in three packages is used. However, in packages RobustGaSP and DiceKriging the range parameter $r$ (see Table 1) in the kernel function is estimated by marginal posterior modes with the robust parameterization and by MLE with upper and lower bounds, respectively, while it is assumed to be fixed and set as $\sqrt{3}/2$ in the RKHSMetaMod package.

Example 4. "The modified sine wave function" (Gu, Palomo, and Berger 2019).

We consider the 1-dimensional modified sine wave function defined over $[0, 1]$ by:

$$m(X) = 3\sin(5\pi X) + \cos(7\pi X).$$

The same experimental design as described in (Gu, Palomo, and Berger 2019) is considered: the design matrix $X$ is a sequence of 12 equally spaced points on $[0, 1]$, and the response variable $Y$ is calculated as $Y = m(X)$:

R> X <- as.matrix(seq(0,1,1/11)); Y <- sinewave(X)

where sinewave function is defined in (Gu, Palomo, and Berger 2019). We build the GP based meta-models by the RobustGaSP and the DiceKriging packages using the constant mean function and kernel Matérn 3/2:

R> library(RobustGaSP)
R> res.rgasp <- rgasp(design=X, response=Y, kernel_type="matern_3_2")
R> library(DiceKriging)
As $d = 1$, we have $D_{max} = 1$. We consider the grid of values of $\mu(1:9) = \mu_{max}/(\sqrt{n} \times 2^{(2:10)})$ and $\gamma(1:5) = (0.2, 0.1, 0.01, 0.005, 0)$. The RKHS meta-models associated with the pair of values $(\mu_i, \gamma_j)$, $i = 1, \cdots, 9$, $j = 1, \cdots, 5$ are estimated using the `RKHSMetMod` function:

```r
R> kernel <- "matern"; Dmax <- 1
R> gamma <- c(0.2, 0.1, 0.01, 0.005, 0)
R> frc <- c(4, 8, 16, 32, 64, 128, 256, 512, 1024)
R> res <- RKHSMetMod(Y, X, kernel, Dmax, gamma, frc, FALSE)
```

Given a testing dataset $(X_T, Y_T)$, the prediction errors associated with the obtained RKHS meta-models are calculated using the `PredErr` function, and the best RKHS meta-model is chosen to be the estimator of the model $m(X)$:

```r
R> XT <- as.matrix(seq(0, 1, 1/11)); YT <- sinewave(XT)
R> Err <- PredErr(X, XT, YT, mu, gamma, res, kernel, Dmax)
```

To compare these three estimators in terms of the prediction quality, we perform prediction on 100 test points, equally spaced in $[0, 1]$:

```r
R> predict_X <- as.matrix(seq(0, 1, 1/99))
R> # prediction with the GP based meta-models:
R> rgasp.predict <- predict(res.rgasp, predict_X)
R> km.predict <- predict(res.km, predict_X, type='UK')
R> # prediction with the best RKHS meta-model:
R> res.predict <- prediction(X, predict_X, kernel, Dmax, res, Err)
```

The prediction results are plotted in Figure 3. The black circles that correspond to the prediction from the `RKHSMetMod` package are closer to the real output than the green and the blue circles corresponding to the predictive means from the `RobustGaSP` and `DiceKriging` packages. The meta-model results are plotted in

![Figure 3](image)

Figure 3: Example 4: Prediction of the modified sine wave function with 100 test points, equally spaced in $[0, 1]$. The x-axis is the real output and the y-axis is the prediction. The black circles are the prediction from `RKHSMetMod`, the green circles are the predictive mean from `RobustGaSP`, and the blue circles are the predictive mean from `DiceKriging`.

Figure 4: The prediction from the `RKHSMetaMod` package plotted as the black curve is much more accurate as an estimate of the true function (plotted in red) than the predictive mean from the `RobustGaSP` and `DiceKriging` packages plotted as the blue and green curves, respectively. As already noted by (Gu, Palomo, and Berger 2019), for that sine wave example, the meta-model from the DiceKriging package "degenerates to the fitted mean with spikes at the design points".

Example 5. "A standard SA 8-dimensional example" (Roustant, Ginsbourger, and Deville 2012).
We consider the 8-dimensional g-function of Sobol implemented in the package `sensitivity`: the function $m(X)$ as defined in Equation \( 17 \) with coefficients $c_1 = 0$, $c_2 = 1$, $c_3 = 4.5$, $c_4 = 9$, $(c_a)_{a=5,6,7,8} = 99$. With these values of coefficients $c_a$, the variables $X_1$, $X_2$, $X_3$ and $X_4$ explain 99.96% of the variance of the function $m(X)$ (see Table 11).

We consider the same experimental design as described in (Roustant, Ginsbourger, and Deville 2012): the design matrices $X$ and $XT$ are 80-point optimal Latin Hypercube Samples of the input variables generated by the `optimumLHS` function of package `lhs`, and the response variables $Y$ and $YT$ are calculated as $Y = m(X)$, and $YT = m(XT)$ using `sobol.fun` function of the package `sensitivity`:

\[
\text{R} > \text{library}(\text{lhs}); \text{X} \leftarrow \text{optimumLHS}(\text{n}, \text{d}); \text{XT} \leftarrow \text{optimumLHS}(\text{n}, \text{d}) \\
\text{R} > \text{library}(\text{sensitivity}); \text{Y} \leftarrow \text{sobol.fun}(\text{X}); \text{YT} \leftarrow \text{sobol.fun}(\text{XT})
\]

Let us first consider the RKHS meta-model method. We set $D_{\text{max}} = 3$, and we consider the grid of values of $\mu(1:9) = \mu_{\text{max}}/(\sqrt{n} \times 2^{(2:10)^j})$, and $\gamma(1:5) = (0.2, 0.1, 0.01, 0.005, 0)$. The RKHS meta-models associated with the pair of values $(\mu, \gamma)$, $i = 1, \cdots, 9$, $j = 1, \cdots, 5$ are estimated using the `RKHSMetMod` function:

\[
\text{R} > \text{kernel} \leftarrow \text{"matern"}; \text{Dmax} \leftarrow 3 \\
\text{R} > \text{gamma} \leftarrow c(0.2, 0.1, 0.01, 0.005,0) \\
\text{R} > \text{frc} \leftarrow c(4,8,16,32,64,128,256,512,1024) \\
\text{R} > \text{res} \leftarrow \text{RKHSMetMod}(\text{Y}, \text{X}, \text{kernel}, \text{Dmax}, \text{gamma}, \text{frc}, \text{FALSE})
\]

Given the testing dataset $(XT, YT)$, the prediction errors associated with the obtained RKHS meta-models are calculated using `PredErr` function, and the best RKHS meta-model is chosen to be the estimator of the model $m(X)$. Finally, the Sobol indices are computed for the best RKHS meta-model using the function `SI_emp`:

\[
\text{R} > \text{Err} \leftarrow \text{PredErr}(\text{X}, \text{XT}, \text{YT}, \mu, \text{gamma}, \text{res}, \text{kernel}, \text{Dmax}) \\
\text{R} > \text{SI} \leftarrow \text{SI_emp}(\text{res}, \text{Err})
\]

Secondly, let us build the GP based meta-model. We use the `km` function of the package `DiceKriging` with the constant mean function and kernel Matérn $3/2$:

\[
\text{R} > \text{library}(\text{DiceKriging}) \\
\text{R} > \text{res.km} \leftarrow \text{km}(\text{design} = \text{X}, \text{response} = \text{Y}, \text{covtype} = \"\text{matern3_2}\")
\]

The Sobol indices associated with the estimated GP based meta-model are calculated using `fast99` function of the package `sensitivity`:

\[
\text{R} > \text{SI.km} \leftarrow \text{fast99(\text{model} = \text{kriging.mean}, \text{factors} = \text{d}, \text{n} = 1000,}
\]

Figure 4: Example 4: The red curve is the graph of the modified sine wave function with 100 test points, equally spaced in $[0,1]$. The black curve is the prediction produced by the `RKHSMetMod` package. The blue curve is the predictive mean by the `DiceKriging` package, and the green curve is the predictive mean produced by the `RobustGaSP` package.
+ q = "qunif", q.arg = list(min = 0, max = 1), m = res.km)

where kriging.mean function is defined in (Roustant, Ginsbourger, and Deville 2012).

The result of the estimation with the best RKHS meta-model and the Kriging based meta-model is drawn in Figure 5. The black circles that correspond to the best RKHS meta-model are closer to the real output than the blue circles corresponding to the GP based meta-model from the DiceKriging package. Another way to evaluate the prediction quality of the estimated meta-models is to consider the mean square error of the fitted meta-model computed by \[ \sum_{i=1}^{80} (m(X_i) - \hat{f}(X_i))^2 / 80. \] We obtained 3.96% and 0.07% for the Kriging based meta-model and the RKHS meta-model, respectively, which confirms the good behavior of the RKHS meta-model.

The estimated Sobol indices associated with the RKHS meta-model and the Kriging based meta-model are given in Table 11. As shown, with RKHS meta-model, we obtained non-zero values for the interactions of order two. Concerning the main effects, excepting the first one, the estimated Sobol indices with the RKHS meta-model are closer to the true ones. However, the interactions of order three are ignored by both meta-models. For a general comparison of the estimation quality of the Sobol indices, one may consider the criterion RE defined in Equation (18), which is equal to 7.95 for the Kriging based meta-model, and 5.59 for the RKHS meta-model. Comparing the values of RE, we can point out that the Sobol indices are better estimated with the RKHS meta-model in that model.

| \(v\) | \{1\} | \{2\} | \{3\} | \{4\} | \{1,2\} | \{1,3\} | \{1,4\} | \{2,3\} | \{2,4\} | \{1,2,3\} | \{1,2,4\} | sum  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| \(\hat{S}_v\) | 71.62 | 17.90 | 2.37 | 0.72 | 5.97 | 0.79 | 0.24 | 0.20 | 0.06 | 0.07 | 0.02 | 99.96 |
| \(\hat{S}_{km,v}\) | 75.78 | 17.42 | 1.71 | 0.47 | 4.00 | 0.05 | 0.07 | 0.28 | 0.09 | 0.00 | 0.00 | 99.87 |
| \(\hat{S}_{km,v}\) | 71.18 | 15.16 | 1.42 | 0.44 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 88.20 |

Table 11: Example 5. The true values of the Sobol indices \(\times 10^2\) greater than \(10^{-2}\) are given in the first raw. The estimated Sobol indices associated with the RKHS meta-model (\(\hat{S}_v\)) and the Kriging based meta-model (\(\hat{S}_{km,v}\)) are given in second and third rows, respectively.

5 Summary and discussion

In this paper, we proposed an R package, called RKHSMetaMod, that estimates a meta-model of a complex model \(m\). This meta-model belongs to a reproducing kernel Hilbert space constructed as a direct
sum of Hilbert spaces (Durrande, Ginsbourger, Rousant, and Carraro 2013). The estimation of the meta-model is carried out via a penalized least-squares minimization allowing both to select and estimate the terms in the Hoeffding decomposition, and therefore, to select the Sobol indices that are non-zero and estimate them (Huet and Taupin 2017). This procedure makes it possible to estimate the Sobol indices of high order, a point known to be difficult in practice. Using the convex optimization tools, RKHSMetaMod package implements two optimization algorithms: the minimization of the RKHS ridge group sparse criterion (13) and the RKHS group lasso criterion (14). Both of these algorithms rely on the Gram matrices $K_v$, $v \in \mathcal{P}$ and their positive definiteness. Currently, the package considers only uniformly distributed input variables. If one is interested by another distribution of the input variables, it suffices to modify the calculation of the kernels $k_{a,v}$, $a = 1, \ldots, d$ (see Equation (19)) in the function calc_Kv of this package (see Remark 3). The available kernels in the RKHSMetaMod package are: Gaussian kernel (with the fixed range parameter $r = 1/2$), Matérn kernel (with the fixed range parameter $r = \sqrt{3}/2$), Brownian kernel, quadratic kernel and linear kernel (see Table 1). With regard to the problem being under study, one may consider other kernels or kernels with different values of the range parameter $r$ and add them easily to the list of the kernels in the calc_Kv function. For the large values of $n$ and $d$ the calculation and storage of the eigenvalues and the eigenvectors of all the Gram matrices $K_v$, $v \in \mathcal{P}$ require a lot of time and a very large amount of memory. In order to optimize the execution time and also the storage memory, except for a function that is written in the kernels or kernels with different values of the range parameter $r$ and linear kernel (see Table 1).

The available kernels in the RcppEigen package are written using the efficient C++ libraries through RcppEigen and RcppGSL packages. These functions are then interfaced with the R environment in order to propose a user friendly package.

6 Supplementary materials

6.1 Appendix

**Preliminary 1.** For $F(x) = \|Ax\|$, where $A$ is a symmetric matrix that does not depend on $x$, we have: $\partial F(x) = \{ A^2 x / \|Ax\| \}$ if $x \neq 0$, and $\partial F(x) = \{ w \in \mathbb{R}^n, \quad \|A^{-1}w\| \leq 1 \}$ if $x = 0$.

**Preliminary 2.** Let $F : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. We have the following first order optimality condition:

$$\hat{x} \in \text{argmin}_{x \in \mathbb{R}^n} F(x) \iff 0 \in \partial F(\hat{x}).$$

This results from the fact that $F(y) \geq F(\hat{x}) + \langle y - \hat{x}, \hat{x} \rangle$ for all $y \in \mathbb{R}^n$ in both cases (Giraud 2014).

6.1.1 RKHS construction

We begin this Section with a brief introduction to the RKHS. Let $\mathcal{H}$ be a Hilbert space of real valued functions on a set $\mathcal{X}$. The space $\mathcal{H}$ is a RKHS if for all $X \in \mathcal{X}$ the evaluation functionals $L_X : \mathcal{H}(f) \rightarrow \mathbb{R}(f(X))$ are continuous. The Riesz representation Theorem ensures the existence of an unique element $k_X(\cdot)$ in $\mathcal{H}$ verifying the property that $\forall X \in \mathcal{X}, \forall f \in \mathcal{H}, f(X) = L_X(f) = \langle f, k_X \rangle_{\mathcal{H}}$, where $(\cdot, \cdot)_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$. It follows that for all $X, X' \in \mathcal{X}$ and $k_X(\cdot), k_{X'}(\cdot)$ in $\mathcal{H}$, we have $k_X(X) = L_X(k_X) = \langle k_X, k_{X'} \rangle_{\mathcal{H}}$. This allows to define the reproducing kernel of $\mathcal{H}$ as $k : \mathcal{X} \times \mathcal{X}((X, X')) \rightarrow \mathbb{R}(k_X(X'))$. The reproducing kernel $k(X, X')$ is positive definite since it is symmetric, and for any $n \in \mathbb{N}, \{X_i\}_{i=1}^n \in \mathcal{X}$ and $\{c_i\}_{i=1}^n \in \mathbb{R}$, we have:

$$\sum_{i=1}^n \sum_{j=1}^n c_i c_j k(X_i, X_j) = \sum_{i=1}^n \sum_{j=1}^n \langle c_i k(X_i, \cdot), c_j k(X_j, \cdot) \rangle_{\mathcal{H}} = \| \sum_{i=1}^n c_i k(X_i, \cdot) \|^2_{\mathcal{H}} \geq 0.$$

For more background on RKHS, we refer to various standard references such as (Aronszajn 1950), (Saitoh 1988), and (Berlinet and Thomas-Agnan 2003).

In this work, the idea is to construct a RKHS $\mathcal{H}$ such that any function $f$ in $\mathcal{H}$ is decomposed as its Hoeffding decomposition, and therefore, any function $f$ in $\mathcal{H}$ is a candidate to approximate the Hoeffding decomposition of $m$. To do so, the method of (Durrande, Ginsbourger, Rousant, and Carraro 2013) as described below is used.

Let $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ be a subset of $\mathbb{R}^d$. For each $a \in \{1, \ldots, d\}$, we choose a RKHS $\mathcal{H}_a$ and its associated kernel $k_a$ defined on the set $\mathcal{X}_a \subset \mathbb{R}$ such that the two following properties are satisfied:

(i) $k_a : \mathcal{X}_a \times \mathcal{X}_a \rightarrow \mathbb{R}$ is $P_a \otimes P_a$ measurable,
(ii) \(E_{X_a} \sqrt{k_a(X_a, X_a)} < \infty.\)

The property (iii) depends on the kernel \(k_a, a = 1, \ldots, d\) and the distribution of \(X_a, a = 1, \ldots, d.\) It is not very restrictive since it is satisfied, for example, for any bounded kernel.

The RKHS \(\mathcal{H}_a\) can be decomposed as a sum of two orthogonal sub-RKHS, \(\mathcal{H}_a = \mathcal{H}_{0a} \perp \mathcal{H}_{1a},\) where \(\mathcal{H}_{0a}\) is the RKHS of zero mean functions, \(\mathcal{H}_{0a} = \{ f_a \in \mathcal{H}_a : E_{X_a} (f_a(X_a)) = 0 \};\) and \(\mathcal{H}_{1a}\) is the RKHS of constant functions, \(\mathcal{H}_{1a} = \{ f_a \in \mathcal{H}_a : f_a(X_a) = C \}.\) The kernel \(k_{0a}\) associated with the RKHS \(\mathcal{H}_a\) is defined by:

\[
k_{0a}(X_a, X_a') = k_a(X_a, X_a') - \frac{E_{U \sim P_a} (k_a(X_a, U)) E_{U \sim P_a} (k_a(X_a', U))}{E_{(U, V) \sim P_a \otimes P_a} k_a(U, V)}.
\]

(19)

Let \(k_v(X_v, X_v') = \prod_{a \in v} k_{0a}(X_a, X_a'),\) then the ANOVA kernel \(k(\cdot, \cdot)\) is defined as follows:

\[
k(X, X') = \prod_{a=1}^{d} \left( 1 + k_{0a}(X_a, X_a') \right) = 1 + \sum_{v \in P} k_v(X_v, X_v').
\]

For \(\mathcal{H}_a\) being the RKHS associated with the kernel \(k_v,\) the RKHS associated with the ANOVA kernel is then defined by \(\mathcal{H} = \prod_{a=1}^{d} (1 \perp \mathcal{H}_{0a}) = 1 + \sum_{v \in P} \mathcal{H}_v,\) where \(\perp\) denotes the \(L^2\) inner product. According to this construction, any function \(f \in \mathcal{H}\) satisfies decomposition (7), \(f(X) = (f, k(X, \cdot))_{\mathcal{H}} = f_0 + \sum_{v \in P} f_v(X_v),\) which is the Hoeffding decomposition of \(f.\)

The regularity properties of the RKHS \(\mathcal{H}\) constructed as described above, depend on the set of the kernels \((k_a, a = 1, \ldots, d).\) This method allows us to choose different approximation spaces independently of the distribution of the input variables \(X_1, \ldots, X_d,\) by choosing different sets of kernels. While, as mentioned earlier, in the meta-modelling approach based on the polynomial Chaos expansion, according to the distribution of the input variables \(X_1, \ldots, X_d,\) a unique family of orthonormal polynomials \(\{\phi_j\}_{j=0}^{\infty}\) is determined. Here, the distribution of the components of \(X\) occurs only for the orthogonalization of the spaces \(\mathcal{H}_v, v \in \mathcal{P},\) and not in the choice of the RKHS, under the condition that properties (i) and (ii) are satisfied. This is one of the main advantages of this method compared to the method based on the truncated polynomial Chaos expansion where the smoothness of the approximation is handled only by the choice of the truncation (Blatman and Sudret 2011).

### 6.1.2 RKHS group lasso algorithm

We consider the minimization of the RKHS group lasso criterion given by,

\[
C_g(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v \|^2 + \sqrt{n} \mu_g \sum_{v \in \mathcal{P}} \|K_v^{1/2} \theta_v\|.
\]

We begin with the constant term \(f_0.\) The ordinary first derivative of the function \(C_g(f_0, \theta)\) at \(f_0\) is equal to:

\[
\frac{\partial C_g}{\partial f_0} = -2 \sum_{i=1}^{n} (Y_i - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v),
\]

and therefore,

\[
\hat{f}_0 = \frac{1}{n} \sum_{i=1}^{n} Y_i - \frac{1}{n} \sum_{i=1}^{n} \sum_{v \in \mathcal{P}} (K_v \theta_v),
\]

(20)

where \((K_v \theta_v)_i\) denotes the \(i\)-th component of \(K_v \theta_v.\)

The next step is to calculate \(\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^{n \times |\mathcal{P}|}} C_g(f_0, \theta).\) Since \(C_g(f_0, \theta)\) is convex and separable, we use a block coordinate descent algorithm, group \(v\) by group \(v.\) In the following, we fix a group \(v,\) and we find the minimizer of \(C_g(f_0, \theta)\) with respect to \(\theta_v\) for the given values of \(f_0\) and \(\theta_w, w \neq v.\) Set

\[
C_{g,v}(f_0, \theta_v) = \|R_v - K_v \theta_v\|^2 + \sqrt{n} \mu_g \|K_v^{1/2} \theta_v\|,
\]

where

\[
R_v = Y - f_0 - \sum_{w \neq v} K_w \theta_w.
\]

(21)
We aim to minimize $C_{g,v}(f_0, \theta_v)$ with respect to $\theta_v$. Let $\partial C_{g,v}$ be the sub-differential of $C_{g,v}(f_0, \theta_v)$ with respect to $\theta_v$:

$$\partial C_{g,v}(f_0, \theta) = \{-2K_v(R_v - K_v\theta_v) + \sqrt{n}\mu_g t_v : t_v \in \partial \|K_v^{1/2}\theta_v\|\}.$$ 

The first order optimality condition (see Preliminary (2)) ensures the existence of $\hat{\theta}_v \in \partial \|K_v^{1/2}\theta_v\|$ fulfilling,

$$-2K_v(R_v - K_v\theta_v) + \sqrt{n}\mu_g \hat{\theta}_v = 0.$$  
(22)

Using the sub-differential definition (see Preliminary [1]), we obtain:

$$\partial \|K_v^{1/2}\theta_v\| = \left\{ \frac{K_v\theta_v}{\|K_v^{1/2}\theta_v\|} \right\} \text{ if } \theta_v \neq 0,$$

and,

$$\partial \|K_v^{1/2}\theta_v\| = \left\{ \hat{\theta}_v \in \mathbb{R}^n, \|K_v^{-1/2}\hat{\theta}_v\| \leq 1 \right\} \text{ if } \theta_v = 0.$$ 

Let $\hat{\theta}_v$ be the minimizer of $C_{g,v}$. The sub-differential equations above give the two following cases:

**Case 1.** If $\hat{\theta}_v = 0$, then there exists $\hat{\gamma} \in \mathbb{R}^n$ such that $\|K_v^{-1/2}\hat{\gamma}\| = 1$ and it fulfills Equation (22), $2K_vR_v = \sqrt{n}\mu_g \hat{\gamma}$. Therefore, the necessary and sufficient condition for which the solution $\hat{\theta}_v = 0$ is the optimal one is $2\|K_v^{1/2}R_v\|/\sqrt{n} \leq \mu_g$.

**Case 2.** If $\hat{\theta}_v \neq 0$, then $\hat{\gamma} = K_v\hat{\theta}_v/\|K_v^{1/2}\hat{\theta}_v\|$ and it fulfills Equation (22),

$$2K_v(R_v - K_v\hat{\theta}_v) = \sqrt{n}\mu_g \frac{K_v\hat{\theta}_v}{\|K_v^{1/2}\hat{\theta}_v\|}.$$ 

We obtain then,

$$\hat{\theta}_v = (K_v + \frac{\sqrt{n}\mu_g}{2\|K_v^{1/2}\hat{\theta}_v\|} I_n)^{-1}R_v.$$  
(23)

Since $\hat{\theta}_v$ appears in both sides of the Equation (23), a numerical procedure is needed:

**Proposition 1.** For $\rho > 0$ let $\theta(\rho) = (K_v + \rho I_n)^{-1}R_v$. There exists a non-zero solution to Equation (23) if and only if there exists $\rho > 0$ such that

$$\mu_g = \frac{2\rho}{\sqrt{n}\|K_v^{1/2}\theta(\rho)\|}.$$  
(24)

Then $\hat{\theta}_v = \theta(\rho)$.

**Proof** If there exists a non-zero solution to Equation (23), then $\|K_v^{1/2}\hat{\theta}_v\| \neq 0$ since $K_v$ is positive definite. Take $\rho = \sqrt{n}\mu_g/2\|K_v^{1/2}\hat{\theta}_v\|$, then $\theta(\rho) = (K_v + \frac{\sqrt{n}\mu_g}{2\|K_v^{1/2}\hat{\theta}_v\|} I_n)^{-1}R_v = \hat{\theta}_v$, and, for such $\rho$ Equation (24) is satisfied. Conversely, if there exists $\rho > 0$ such that Equation (24) is satisfied, then $\|K_v^{1/2}\theta(\rho)\| \neq 0$ and $\rho = \frac{\sqrt{n}\mu_g}{2\|K_v^{1/2}\theta(\rho)\|}$. Therefore, $\theta(\rho) = (K_v + \frac{\sqrt{n}\mu_g}{2\|K_v^{1/2}\theta(\rho)\|} I_n)^{-1}R_v$, which is Equation (23) calculated in $\hat{\theta}_v = \theta(\rho)$.

**Remark 5.** Define $y(\rho) = 2\rho||K_v^{1/2}\theta(\rho)|| - \sqrt{n}\mu_g$ with $\theta(\rho) = (K_v + \rho I_n)^{-1}R_v$, then $y(\rho) = 0$ has a unique solution, denoted $\hat{\rho}$, which leads to calculate $\hat{\theta}(\hat{\rho})$.

**Proof** For $\rho = 0$ we have $y(0) = -\sqrt{n}\mu_g < 0$, since $\mu_g > 0$; and for $\rho \to +\infty$ we have $y(\rho) > 0$, since $\|K_v^{1/2}(\frac{L_v}{\rho} + I_n)^{-1}R_v\| \to \|K_v^{1/2}R_v\|$ and $\|2K_v^{1/2}R_v\| > \sqrt{n}\mu_g$.

Moreover, we have $y(\rho) = 2\|((\frac{L_v}{\rho} + k_v^{-1})^{-1}k_v^{-1/2}R_v\| - \sqrt{n}\mu_g$, which is equal to $2(X^TA^{-2}X)^{1/2} - \sqrt{n}\mu_g$, with $A = (I_n/\rho + k_v^{-1})^{-1}$ and $X = k_v^{-1/2}R_v$. The first derivative of $y(\rho)$ in $\rho$ is obtained by $\frac{\partial y(\rho)}{\partial \rho} = (X^TA^{-2}X)^{-1/2}\frac{\partial (X^TA^{-2}X)}{\partial \rho}$. Finally, by simple calculations we get,

$$\frac{\partial y(\rho)}{\partial \rho} = \frac{2\|((\frac{L_v}{\rho} + k_v^{-1})^{-3/2}k_v^{-1/2}R_v\|}{\rho^2\|((\frac{L_v}{\rho} + k_v^{-1})^{-1}k_v^{-1/2}R_v\|} > 0.$$
Therefore \( y(\rho) \) is an increasing function of \( \rho \), and the proof is complete. \( \square \)

In order to calculate \( \rho \) and so \( \tilde{\theta}_v = \theta(\rho) \), we use Algorithm 4 which is a part of the RKHS group lasso Algorithm 1 when \( \tilde{\theta}_v \neq 0 \).

**Algorithm 4 Algorithm to find \( \rho \) as well as \( \tilde{\theta}_v \)**

1: if \( \tilde{\theta}_v^{\text{old}} = 0 \) then \( \triangleright \tilde{\theta}_v^{\text{old}} \) is \( \tilde{\theta}_v \) computed in the previous step of the RKHS group lasso algorithm.
2: Set \( \rho \leftarrow 1 \) and calculate \( y(\rho) \)
3: if \( y(\rho) > 0 \) then
4: Find \( \hat{\rho} \) that minimizes \( y(\rho) \) on the interval \([0, 1]\)
5: else
6: repeat
7: Set \( \rho \leftarrow \rho \times 10 \) and calculate \( y(\rho) \)
8: until \( y(\rho) > 0 \)
9: Find \( \hat{\rho} \) that minimizes \( y(\rho) \) on the interval \([\rho/10, \rho]\)
10: end if
11: else
12: Set \( \rho \leftarrow \frac{\sqrt{|\mu_v^2}}{2|K_v - \tilde{\theta}_v^{\text{old}}|} \) and calculate \( y(\rho) \)
13: if \( y(\rho) > 0 \) then
14: repeat
15: Set \( \rho \leftarrow \rho/10 \) and calculate \( y(\rho) \)
16: until \( y(\rho) < 0 \)
17: Find \( \hat{\rho} \) that minimizes \( y(\rho) \) on the interval \([\rho, \rho \times 10]\)
18: else
19: repeat
20: Set \( \rho \leftarrow \rho \times 10 \) and calculate \( y(\rho) \)
21: until \( y(\rho) > 0 \)
22: Find \( \hat{\rho} \) that minimizes \( y(\rho) \) on the interval \([\rho/10, \rho]\)
23: end if
24: end if
25: calculate \( \hat{\theta}_v = \theta(\hat{\rho}) \)

### 6.1.3 Computational cost

The complexity for the matrices \( K_v, v \in \mathcal{P} \) is equal to \( n^3 \), which is given by the singular value decomposition to get eigenvalues and eigenvectors of each \( K_v \). Supposing that the matrices \( K_v, v \in \mathcal{P} \) was first created and are already stored, the complexity for the constant term \( f_0 \) is given by the second term in equation (20), which is equal to \( n^2 \). Given \( \hat{\rho} \), the complexity for \( \hat{\theta}_v, v \in \mathcal{P} \), is given by the backsolving of \( (K_v + \hat{\rho} I_n) \hat{\theta}_v = R_v \) to get \( \hat{\theta}_v \), which is equal to \( n^2 \). The computation of \( \hat{\rho} \) is done using Brent-Dekker method implemented as function `gsl_root_fsolver_brent` from [Galassi, Davies, Theiler, Gough, Jungman, Alken, Booth, Rossi, and Ulerych 2018](#). This method combines an interpolation strategy with the bisection algorithm and takes \( O(m) \) iterations to converge, where \( m \) is the number of steps that the bisection algorithm would take.

### 6.1.4 RKHS ridge group sparse algorithm

We consider the minimization of the RKHS ridge group sparse criterion:

\[
C(f_0, \theta) = \|Y - f_0 I_n - \sum_{v \in \mathcal{P}} K_v \theta_v \|^2 + \sqrt{n} \gamma \sum_{v \in \mathcal{P}} \|K_v \theta_v\| + n \mu \sum_{v \in \mathcal{P}} \|K_v^{1/2} \theta_v\|.
\]

The constant term \( f_0 \) is estimated as in the RKHS group lasso algorithm. In order to calculate \( \hat{\theta} = \arg\min_{\theta \in \mathbb{R}^{m \times \gamma}} C(f_0, \theta) \), we use once again the block coordinate descent algorithm group \( v \) by group \( v \). In the following, we fix a group \( v \), and we find the minimizer of \( C(f_0, \theta) \) with respect to \( \theta_v \) for given values of \( f_0 \) and \( \theta_w, w \neq v \). We aim at minimizing with respect to \( \theta_v \):

\[
C_v(f_0, \theta_v) = \|R_v - K_v \theta_v\|^2 + \sqrt{n} \gamma \|K_v \theta_v\| + n \mu \|K_v^{1/2} \theta_v\|,
\]

24
where $R_v$ is defined by (21).

Let $\partial C_v$ be the sub-differential of $C_v(f_0, \theta_v)$ with respect to $\theta_v$,

\[
\partial C_v = \{-2K_v(R_v - K_v\theta_v) + \sqrt{n}\gamma s_v + n\mu t_v : s_v \in \partial\|K_v\theta_v\|, \quad t_v \in \partial\|K_v^{1/2}\theta_v\|\},
\]

According to the first order optimality condition (see Preliminary 2), we know that there exists $\hat{s}_v \in \partial\|K_v\theta_v\|$ and $\hat{t}_v \in \partial\|K_v^{1/2}\theta_v\|$ such that

\[-2K_v(R_v - K_v\theta_v) + \sqrt{n}\gamma \hat{s}_v + n\mu \hat{t}_v = 0. \tag{25}\]

The sub-differential definition (see Preliminary 1) gives:

\[
\{\partial\|K_v^{1/2}\theta_v\| = \{\frac{K_v\theta_v}{\|K_v^{1/2}\theta_v\|}\}, \partial\|K_v\theta_v\| = \{\frac{K_v^2\theta_v}{\|K_v\theta_v\|}\}\} \quad \text{if} \quad \theta_v \neq 0,
\]

and,

\[
\{\partial\|K_v^{1/2}\theta_v\| = \{\hat{t}_v \in \mathbb{R}^n, \|K_v^{1/2}\hat{t}_v\| \leq 1\}, \partial\|K_v\theta_v\| = \{\hat{s}_v \in \mathbb{R}^n, \|K_v^{-1}\hat{s}_v\| \leq 1\}\} \quad \text{if} \quad \theta_v = 0.
\]

Let $\hat{\theta}_v$ be the minimizer of the $C_v(f_0, \theta_v)$. Using the sub-differential equations above, the estimator $\hat{\theta}_v$, $v \in \mathcal{P}$ is obtained following the two cases below:

Case 1. If $\hat{\theta}_v = 0$, then there exists $\hat{s}_v \in \mathbb{R}^n$ such that $\|K_v^{-1}\hat{s}_v\| \leq 1$ and it fulfills Equation (25), $2K_v(R_v - n\mu \hat{t}_v) = \sqrt{n}\gamma \hat{s}_v$, with $\hat{t}_v \in \mathbb{R}^n$, $\|K_v^{-1/2}\hat{t}_v\| \leq 1$. Set $J(\hat{t}_v) = \|2R_v - n\mu K_v^{-1}\hat{t}_v\|$, and $J^* = \arg\min_{\hat{t}_v \in \mathbb{R}^n} \{J(\hat{t}_v)\}$, such that $\|K_v^{-1/2}\hat{t}_v\| \leq 1$. Then the solution to Equation (25) is zero if and only if $J^* \leq \gamma$.

Case 2. If $\hat{\theta}_v \neq 0$, then we have $\hat{s}_v = K_v^2\hat{\theta}_v / \|K_v\hat{\theta}_v\|$, and $\hat{t}_v = \hat{\theta}_v / \|K_v^{1/2}\hat{\theta}_v\|$ fulfilling Equation (25), $2K_v(R_v - K_v\hat{\theta}_v) = \sqrt{n}\gamma \hat{s}_v + n\mu \hat{t}_v / \|K_v^{1/2}\hat{\theta}_v\|$. Set $\hat{\theta}_v = (K_v + \frac{\sqrt{n}\gamma}{2\|K_v\hat{\theta}_v\|} - \frac{n\mu}{2\|K_v^{1/2}\hat{\theta}_v\|} I_n)^{-1}R_v$ if $\hat{\theta}_v \neq 0$.

In this case the calculation of $\hat{\theta}_v$ needs a numerical algorithm.

**Proposition 2.** (Proposition 8.4 in [Huet and Taupin 2017]) For $\rho_1, \rho_2 > 0$, let $\theta(\rho_1, \rho_2) = (K_v + \rho_1 K_v + \rho_2 I_n)^{-1}R_v$. If $\mu > 0$, there exists a non zero solution to Equation (26) if and only if there exists $\rho_1, \rho_2 > 0$ such that $\gamma = \frac{2\mu}{\sqrt{n}}\|K_v\theta(\rho_1, \rho_2)\|$, and $\mu = \frac{2\mu}{\sqrt{n}}\|K_v^{-1/2}\theta(\rho_1, \rho_2)\|$. Then $\theta_v = \theta(\rho_1, \rho_2)$.

**Proof** The proof is given in [Huet and Taupin 2017].

### 6.1.5 Computational cost

The complexity for the matrices $K_v$, $v \in \mathcal{P}$ and the constant term $f_0$ is the same as for RKHS group lasso algorithm. Given $\hat{\rho}_1, \hat{\rho}_2$, the complexity for $\hat{\theta}_v$, $v \in \mathcal{P}$, is given by the backsolving of $(K_v + \rho_1 K_v + \rho_2 I_n)\hat{\theta}_v = R_v$ to get $\hat{\theta}_v$, which is equal to $n^2$. The computation of $\hat{\rho}_1, \hat{\rho}_2$, is insured using a combination of three methods: first we implement a modified version of Newton’s method, if it does not achieve the convergence second we implement a version of the Hybrid algorithm and if it does not achieve the convergence, third we implement a version of the discrete Newton algorithm called Broyden algorithm. These methods are implemented as functions `gsl_multroots_fdfsolver_gnewton`, `gsl_multroots_fdfsolver_hybrids`, and `gsl_multroots_fdfsolver_broyden` from [Galassi, Davies, Theiler, Gough, Jungman, Alken, Booth, Rossi, and Uélerich 2018], respectively. These methods converge fast in general, and for $n$ big enough their complexity is dominated by $n^2$.

### 6.2 Overview of the RKHSMetaMod functions

In the R environment, one can install and load the `RKHSMetaMod` package by using the following commands:

**Example 6.** `install.packages("RKHSMetaMod")` library("RKHSMetaMod")
The optimization problems in this package are solved using block coordinate descent algorithm which requires various computational algorithms including generalized Newton, Broyden and Hybrid methods. In order to gain the efficiency in terms of the calculation time and be able to deal with high dimensional problems, the computationally efficient tools of C++ packages Eigen (Guennebaud, Jacob, et al. 2010) and GSL (Galassi, Davies, Theiler, Gough, Jungman, Alken, Booth, Rossi, and Ulerich 2018) via ReppEigen (Bates and Eddelbuettel 2013) and ReppGSL (Eddelbuettel and Francois 2019) packages are used in the RKHSMetaMod package. For different examples of usage of ReppEigen and ReppGSL functions see the work by (Eddelbuettel 2013).

The complete documentation of RKHSMetaMod package is available from CRAN (Kamari 2019). Here, a brief documentation of some of its main and companion functions is presented in the next two Sections.

6.2.1 Main RKHSMetaMod functions

Let us begin by introducing some notations. For a given Dmax ∈ N, let \( P_{\text{Dmax}} \) be the set of parts of \( \{1,...,d\} \) with dimension 1 to Dmax. The cardinal of \( P_{\text{Dmax}} \) is denoted by \( v_{\text{Max}} = \sum_{j=1}^{D\text{max}} (d_j) \).

**RKHSMetMod function:** For a given value of Dmax and a chosen kernel (see Table 1), this function calculates the Gram matrices \( K_v, v \in P_{\text{Dmax}} \), and produces a sequence of estimators \( \hat{f} \) associated with a given grid of values of tuning parameters \( \mu, \gamma \), i.e. the solutions to the RKHS ridge group sparse (if \( \gamma \neq 0 \)) or the RKHS group lasso problem (if \( \gamma = 0 \)). Table 12 gives a summary of all the input arguments of the RKHSMetaMod function as well as the default values for non-mandatory arguments.

| Input parameter | Description |
|-----------------|-------------|
| Y               | Vector of the response observations of size n. |
| X               | Matrix of the input observations with n rows and d columns. Rows correspond to the observations and columns correspond to the variables. |
| kernel          | Character, indicates the type of the kernel (see Table 1) chosen to construct the RKHS \( \mathcal{H} \). |
| Dmax            | Integer, between 1 and d, indicates the maximum order of interactions considered in the RKHS meta-model: Dmax= 1 is used to consider only the main effects, Dmax= 2 to include the main effects and the second-order interactions, and so on. |
| gamma           | Vector of non-negative scalars, values of the tuning parameter \( \gamma \) in decreasing order. If \( \gamma = 0 \) the function solves the RKHS group lasso optimization problem and for \( \gamma > 0 \) it solves the RKHS ridge group sparse optimization problem. |
| frc             | Vector of positive scalars. Each element of the vector sets a value to the tuning parameter \( \mu: \mu = \mu_{\text{max}}/(\sqrt{n} \times \text{frc}) \). The value \( \mu_{\text{max}} \) (see Equation (16)) is calculated inside the program. Logical. Set as True to print: the group \( v \) for which the correction of the Gram matrix \( K_v \) is done (see Section 3.1), and for each pair of the tuning parameters \( (\mu, \gamma) \): the number of current iteration, active groups and convergence criteria. It is set as FALSE by default. |

Table 12: List of the input arguments of the RKHSMetaMod function.

The RKHSMetaMod function returns a list of l components, with l being equal to the number of pairs of the tuning parameters \( (\mu, \gamma) \), i.e. \( l = |\gamma| \times |\text{frc}| \). Each component of the list is a list of three components "mu", "gamma" and "Meta-Model":

- **mu**: value of the tuning parameter \( \mu \) if \( \gamma > 0 \), or \( \mu_0 = \sqrt{n} \times \mu \) if \( \gamma = 0 \).

- **gamma**: value of the tuning parameter \( \gamma \).

- **Meta-Model**: a RKHS ridge group sparse or RKHS group lasso object associated with the tuning parameters mu and gamma. Table 13 gives a summary of all arguments of the output "Meta-Model" of RKHSMetaMod function.
| Output parameter | Description |
|------------------|-------------|
| intercept        | Scalar, estimated value of intercept. |
| teta             | Matrix with vMax rows and n columns. Each row of the matrix is the estimated vector $\theta_v$ for $v = 1, \ldots, vMax$. |
| fit.v            | Matrix with $n$ rows and vMax columns. Each row of the matrix is the estimated value of $f_v = K_v \theta_v$. |
| fitted           | Vector of size $n$, indicates the estimator of $m$. |
| Norm.n           | Vector of size vMax, estimated values for the empirical $L^2$-norm. |
| Norm.H           | Vector of size vMax, estimated values for the Hilbert norm. |
| supp             | Vector of active groups. |
| Nsupp            | Vector of the names of the active groups. |
| SCR              | Scalar equals to $\|Y - f_0 - \sum_v K_v \theta_v\|^2$. |
| crit             | Scalar indicates the value of the penalized criterion. |
| gamma.v          | Vector of size vMax, coefficients of the empirical $L^2$-norm. |
| mu.v             | Vector of size vMax, coefficients of the Hilbert norm. |
| iter             | List of two components: maxIter, and the number of iterations until the convergence is achieved. |
| convergence      | TRUE or FALSE. Indicates whether the algorithm has converged or not. |
| RelDiffCrit      | Scalar, value of the first convergence criterion at the last iteration, i.e. $\|\theta_{lastIter} - \theta_{lastIter-1}/\theta_{lastIter-1}\|^2$. |
| RelDiffPar       | Scalar, value of the second convergence criterion at the last iteration, i.e. $crit_{lastIter} - crit_{lastIter-1}/crit_{lastIter-1}$. |

Table 13: List of the arguments of the output "Meta-Model" of RKHSMetMod function.

**RKHSMetMod_qmax function:** For a given value of Dmax and a chosen kernel (see Table 1), this function calculates the Gram matrices $K_v$, $v \in \mathcal{P}_{Dmax}$; determines $\mu$, referred to as $\mu_{qmax}$, for which the number of groups in the support of the RKHS group lasso solution is equal to $qmax$; and produces a sequence of estimators $\hat{f}$ associated with the tuning parameter $\mu_{qmax}$ and a grid of values of the tuning parameter $\gamma$. All the estimators $\hat{f}$ produced by this function have at most $qmax$ groups in their support. This function has the following input arguments:

- $Y$, $X$, kernel, Dmax, gamma, verbose (see Table 12).
- $qmax$: integer, the maximum number of groups in the support of the obtained solution.
- $\text{rat}$: positive scalar, to restrict the minimum value of $\mu$ considered in Algorithm 3

\[
\mu_{\text{min}} = \frac{\mu_{\text{max}}}{(\sqrt{n} \times \text{rat})},
\]

where the value of $\mu_{\text{max}}$ is given by Equation (16) and is calculated inside the program.

- $\text{Num}$: integer, to restrict the number of different values of the tuning parameter $\mu$ to be evaluated in the RKHS group lasso algorithm until it achieves $\mu_{qmax}$. For example, if Num equals 1, the program is implemented for three different values of $\mu \in [\mu_{\text{min}}, \mu_{\text{max}}]$: \[
\begin{align*}
\mu_1 &= \frac{(\mu_{\text{min}} + \mu_{\text{max}})}{2}, \\
\mu_2 &= \left\{ \begin{array}{ll} 
\frac{(\mu_{\text{max}} + \mu_1)}{2} & \text{if } |\hat{S}_{\hat{f}(\mu_1)_{\text{Group Lasso}}}| < qmax, \\
\frac{(\mu_1 + \mu_{\text{max}})}{2} & \text{if } |\hat{S}_{\hat{f}(\mu_1)_{\text{Group Lasso}}}| > qmax,
\end{array} \right.
\end{align*}
\]

\[
\mu_3 = \mu_{\text{min}},
\]

where $|\hat{S}_{\hat{f}(\mu_1)_{\text{Group Lasso}}}|$ is the number of groups in the support of the solution of the RKHS group lasso problem, Algorithm 1, associated with $\mu_1$.

If Num $> 1$, the path to cover the interval $[\mu_{\text{min}}, \mu_{\text{max}}]$ is detailed in Algorithm 3.
The \texttt{RKHSMetMod_qmax} function returns a list of three components "mus", "qs", and "MetaModel":

- \texttt{mus}: vector of all values of $\mu_i$ in Algorithm 3.
- \texttt{qs}: vector with the same length as \texttt{mus}. Each element of the vector shows the number of groups in the support of the RKHS meta-model obtained by solving RKHS group lasso problem for an element in \texttt{mus}.
- \texttt{MetaModel}: list with the same length as the vector gamma. Each component of the list is a list of three components "mu", "gamma" and "Meta-Model":
  - \texttt{mu}: value of $\mu_{q_{\text{max}}}$.
  - \texttt{gamma}: element of the input vector gamma associated with the estimated "Meta-Model".
  - \texttt{Meta-Model}: a RKHS ridge group sparse or RKHS group lasso object associated with the tuning parameters \texttt{mu} and gamma (see Table 13).

### 6.2.2 Companion functions

\texttt{calc_Kv} function: For a given value of Dmax and a chosen kernel (see Table 1), this function calculates the Gram matrices $K_v$, $v \in \mathcal{P}_{D_{\text{max}}}$, and returns their associated eigenvalues and eigenvectors. This function has,

- four mandatory input arguments:
  - \texttt{Y}, \texttt{X}, \texttt{kernel}, Dmax (see Table 12).
- three facultative input arguments:
  - \texttt{correction}: logical, set as TRUE to make correction to the matrices $K_v$ (see Section 3.1). It is set as TRUE by default.
  - \texttt{verbose}: logical, set as TRUE to print the group for which the correction is done. It is set as TRUE by default.
  - \texttt{tol}: scalar to be chosen small, set as $1e^{-8}$ by default.

The \texttt{calc_Kv} function returns a list of two components "kv" and "names.Grp":

- \texttt{kv}: list of vMax components, each component is a list of,
  - \texttt{Evalues}: vector of eigenvalues.
  - \texttt{Q}: matrix of eigenvectors.
- \texttt{names.Grp}: vector of group names of size vMax.

\texttt{RKHSgrplasso} function: For a given value of the tuning parameter $\mu_g$, this function fits the solution to the RKHS group lasso optimization problem by implementing Algorithm 1. This function has,

- three mandatory input arguments:
  - \texttt{Y} (see Table 12).
  - \texttt{Kv}: list of the eigenvalues and the eigenvectors of the positive definite Gram matrices $K_v$ for $v = 1, \ldots, \text{vMax}$ and their associated group names (output of the function \texttt{calc_Kv}).
  - \texttt{mu}: positive scalar indicates the value of the tuning parameter $\mu_g$ defined in Equation (15).
- two facultative input arguments:
  - \texttt{maxIter}: integer, to set the maximum number of loops through all groups. It is set as 1000 by default.
  - \texttt{verbose}: logical, set as TRUE to print the number of current iteration, active groups and convergence criterion. It is set as FALSE by default.

This function returns a RKHS group lasso object associated with the tuning parameter $\mu_g$. Its output is a list of 13 components:

- intercept, teta, fit.v, fitted, Norm.H, supp, Nsupp, SCR, crit, MaxIter, convergence, RelDiffCrit, and RelDiffPar (see Table 13).
**mu_max function:** This function calculates the value $\mu_{\text{max}}$ defined in Equation 16. It has two mandatory input arguments: the response vector $Y$, and the list matZ of the eigenvalues and eigenvectors of the positive definite Gram matrices $K_v$ for $v = 1, \ldots, \text{vMax}$. This function returns the $\mu_{\text{max}}$ value.

**pen_MetMod function:** This function produces a sequence of the RKHS meta-models associated with a given grid of values of the tuning parameters $\mu, \gamma$. Each RKHS meta-model in the sequence is the solution to the RKHS ridge group sparse optimization problem (obtained by implementing Algorithm 2) associated with a pair of values of $(\mu, \gamma)$ in the grid of values of $\mu, \gamma$. This function has,

- seven mandatory input arguments:
  - $Y$ (see Table 12).
  - gamma: vector of positive scalars. Values of the penalty parameter $\gamma$ in decreasing order.
  - Kv: list of the eigenvalues and the eigenvectors of the positive definite Gram matrices $K_v$ for $v = 1, \ldots, \text{vMax}$ and their associated group names (output of the function calc_Kv).
  - mu: vector of positive scalars. Values of the tuning parameter $\mu$ in decreasing order.
  - resg: list of the RKHS group lasso objects associated with the components of "mu", used as initial parameters at Step 1.
  - gama_v and mu_v: vector of vMax positive scalars. These two inputs are optional. They are provided to associate the weights to the two penalty terms in the RKHS ridge group sparse criterion (13). In order to consider no weights, i.e. all the weights are equal to one, we set these two inputs to scalar zero.

- three facultative input arguments:
  - maxIter: integer, to set the maximum number of loops through initial active groups at Step 1 and maximum number of loops through all groups at Step 2. It is set as 1000 by default.
  - verbose: logical, set as TRUE to print for each pair of the tuning parameters $(\mu, \gamma)$, the number of current iteration, active groups and convergence criterion. It is set as FALSE by default.
  - calcStwo: logical, set as TRUE to execute Step 2. It is set as FALSE by default.

The function **pen_MetMod** returns a list of $l$ components, with $l$ being equal to the number of pairs of the tuning parameters $(\mu, \gamma)$. Each component of the list is a list of three components "mu", "gamma" and "Meta-Model":

- mu: positive scalar, an element of the input vector "mu" associated with the estimated "Meta-Model".
- gamma: positive scalar, an element of the input vector "gamma" associated with the estimated "Meta-Model".
- Meta-Model: a RKHS ridge group sparse object associated with the tuning parameters mu and gamma (see Table 13).

**PredErr function:** By considering a testing dataset, this function calculates the prediction errors for the obtained RKHS meta-models. This function has eight mandatory input arguments:

- $X$, gamma, kernel, Dmax (see Table 12).
- XT: matrix of observations of the testing dataset with $n_{\text{test}}$ rows and $d$ columns.
- YT: vector of response observations of the testing dataset of size $n_{\text{test}}$.
- mu: vector of positive scalars. Values of the tuning parameter $\mu$ in decreasing order.
- res: list of the estimated RKHS meta-models for the learning dataset associated with the tuning parameters $(\mu, \gamma)$ (it could be the output of one of the functions RKHSMetMod or pen_MetMod).

Note that, the same kernel and Dmax have to be chosen as the ones used for the learning dataset.

The function **PredErr** returns a matrix of the prediction errors. Each element of the matrix corresponds to the prediction error of one RKHS meta-model in "res".
**prediction function:** This function calculates the predicted values for a new dataset based on the best RKHS meta-model estimator. It has six input arguments:

- X, kernel, Dmax (see Table 12).
- Xnew: matrix of new observations with \( n^{\text{new}} \) rows and \( d \) columns.
- res: list of the estimated RKHS meta-models for a learning dataset associated with the tuning parameters \((\mu, \gamma)\) (it could be the output of one of the functions RKHSMetMod, RKHSMetMod_qmax or pen_MetMod).
- Err: matrix of the prediction errors associated with the RKHS meta-models in "res" (output of the function PredErr).

The function `prediction` returns a vector of the predicted values based on the best RKHS meta-model estimator in "res". This function is available at [GitHub](https://github.com).

**SI_emp function:** For each RKHS meta-model \( \hat{f} \), this function calculates the empirical Sobol indices for all the groups that are active in the support of \( \hat{f} \). This function has two input arguments:

- res: list of the estimated meta-models using RKHS ridge group sparse or RKHS group lasso algorithms (it could be the output of one of the functions RKHSMetMod, RKHSMetMod_qmax or pen_MetMod).
- ErrPred: matrix or NULL. If matrix, each element of the matrix corresponds to the prediction error of a RKHS meta-model in "res" (output of the function PredErr). Set as NULL by default.

The empirical Sobol indices are then calculated for each RKHS meta-model in "res", and a list of vectors of the Sobol indices is returned. If the argument "ErrPred" is the matrix of the prediction errors, the vector of empirical Sobol indices is returned for the best RKHS meta-model in the "res".
References

Aronszajn, N. (1950). Theory of Reproducing Kernels. Transactions of the American Mathematical Society 68(3), 337–404.

Bates, D. and D. Eddelbuettel (2013). Fast and Elegant Numerical Linear Algebra Using the ReppEigen Package. Journal of Statistical Software 52(5), 1–24.

Berlinet, A. and C. Thomas-Agnan (2003). Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer US.

Blatman, G. and B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on least angle regression. Journal of computational Physics 230, 2345–2367.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011, January). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Found. Trends Mach. Learn. 3(1), 1–122.

Bubeck, S. (2015, November). Convex Optimization: Algorithms and Complexity. Found. Trends Mach. Learn. 8(3-4), 231–357.

Carnell, R. (2021). lhs: Latin Hypercube Samples. R package version 1.1.3.

Dancik, G. M. and K. S. Dorman (2008). mlegp: statistical analysis for computer models of biological systems using R. Bioinformatics 24(17), 1966.

Durrande, N., D. Ginsbourger, and O. Roustant (2012, 4). Additive Covariance kernels for high-dimensional Gaussian Process modeling. Annales de la faculté des sciences de Toulouse Mathématiques 21(3), 481–499.

Durrande, N., D. Ginsbourger, O. Roustant, and L. Carraro (2013). ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis. Journal of Multivariate Analysis 115, 57 – 67.

Eddelbuettel, D. (2013). Seamless R and C++ Integration with Repp. Springer Publishing Company, Incorporated.

Eddelbuettel, D. and R. Francois (2019). RcppGSL: ‘Rcpp’ Integration for ‘GNU GSL’ Vectors and Matrices. R package version 0.3.7.

Fu, W. J. (1998). Penalized Regressions: The Bridge versus the Lasso. Journal of Computational and Graphical Statistics 7(3), 397–416.

Galassi, M., J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich (2018). GNU Scientific Library Reference Manual.

Giraud, C. (2014). Introduction to High-Dimensional Statistics. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Gu, M. (2019). Jointly Robust Prior for Gaussian Stochastic Process in Emulation, Calibration and Variable Selection. Bayesian Analysis 14(3), 857 – 885.

Gu, M., J. Palomo, and O. Berger, James (2019). RobustGaSP: Robust Gaussian Stochastic Process Emulation in R. The R Journal 11(1), 112.

Gu, M., X. Wang, and J. O. Berger (2018). Robust Gaussian stochastic process emulation. The Annals of Statistics 46(6A), 3038 – 3066.

Guennebaud, G., B. Jacob, et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Hastie, T., R. Tibshirani, and M. Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations. Chapman & Hall/CRC.

Homma, T. and A. Saltelli (1996). Importance measures in global sensitivity analysis of nonlinear models. Reliability Engineering & System Safety 52(1), 1 – 17.

Huet, S. and M.-L. Taupin (2017). Metamodel construction for sensitivity analysis. ESAIM: Procs 60, 27–69.

Kamari, H. (2019). RKHSMetaMod: Ridge Group Sparse Optimization Problem for Estimation of a Meta Model Based on Reproducing Kernel Hilbert Spaces. R package version 1.1.

Kennedy, M. C. and A. O’Hagan (2000). Bayesian Calibration of Computer Models. Journal of the Royal Statistical Society, Series B, Methodological 63, 425–464.
Kimeldorf, G. S. and G. Wahba (1970, 04). A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. *Ann. Math. Statist.* 41(2), 495–502.

Kleijnen, J. P. C. (2007). *Design and Analysis of Simulation Experiments* (1st ed.). Springer Publishing Company, Incorporated.

Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research* 192(3), 707 – 716.

Le Gratiet, L., C. Cannamela, and B. Iooss (2014). A Bayesian Approach for Global Sensitivity Analysis of (Multifidelity) Computer Codes. *SIAM/ASA Journal on Uncertainty Quantification* 2(1), 336–363.

Le Gratiet, L., S. Marelli, and B. Sudret (2017). *Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes*, pp. 1289–1325. Cham: Springer International Publishing.

Lin, Y. and H. H. Zhang (2006, 10). Component selection and smoothing in multivariate nonparametric regression. *Ann. Statist.* 34(5), 2272–2297.

Liu, H., L. Wasserman, and J. D. Lafferty (2009). Nonparametric regression and classification with joint sparsity constraints. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems* 21, pp. 969–976. Curran Associates, Inc.

Marrel, A., B. Iooss, B. Laurent, and O. Roustant (2009). Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety* 94(3), 742 – 751.

Mebane, W. and J. Sekhon (2011). Genetic Optimization Using Derivatives: The rgenoud Package for R. *Journal of Statistical Software, Articles* 42(11), 1–26.

Meier, L. (2020). *grplasso: Fitting User-Specified Models with Group Lasso Penalty*. R package version 0.4-7.

Meier, L., S. van de Geer, and P. Bühlmann (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society. Series B* 70(1), 53–71.

Meier, L., S. van de Geer, and P. Bühlmann (2009, 12). High-dimensional additive modeling. *Ann. Statist.* 37(6B), 3779–3821.

Oakley, J. E. and A. O’Hagan (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 66(3), 751–769.

Raskutti, G., M. J. Wainwright, and B. Yu (2009). Lower bounds on minimax rates for nonparametric regression with additive sparsity and smoothness. In *Advances in Neural Information Processing Systems*.

Raskutti, G., M. J. Wainwright, and B. Yu (2012, February). Minimax-optimal Rates for Sparse Additive Models over Kernel Classes via Convex Programming. *J. Mach. Learn. Res.* 13(1), 389–427.

Ravikumar, P., J. Lafferty, H. Liu, and L. Wasserman (2009). Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71(5), 1009–1030.

Roustant, O., D. Ginsbourger, and Y. Deville (2012). DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *Journal of Statistical Software, Articles* 51(1), 1–55.

Saitoh, S. (1988). *Theory of reproducing kernels and its applications*. Pitman research notes in mathematics series. Longman Scientific & Technical.

Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications* 145(2), 280 – 297.

Saltelli, A., K. Chan, and E. Scott (2009). *Sensitivity Analysis*. Wiley.

Saltelli, A., S. Tarantola, and K. P.-S. Chan (1999). A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output. *Technometrics* 41(1), 39–56.

Schoutens, W. (2000). *Stochastic Processes and Orthogonal Polynomials*. Lecture Notes in Statistics. Springer New York.

Sobol, I. M. (1993). Sensitivity Estimates for Nonlinear Mathematical Models. In *Sensitivity Estimates for Nonlinear Mathematical Models*.

Soize, C. and R. Ghanem (2004). Physical Systems with Random Uncertainties: Chaos Representations with Arbitrary Probability Measure. *SIAM Journal on Scientific Computing* 26(2), 395–410.
Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety* 93(7), 964 – 979. Bayesian Networks in Dependability.

Van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Welch, W. J., R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris (1992). Screening, Predicting, and Computer Experiments. *Technometrics* 34(1), 15–25.

Wiener, N. (1938). The Homogeneous Chaos. *American Journal of Mathematics* 60(4), 897–936.

Yang, Y. and H. Zou (2015, November). A Fast Unified Algorithm for Solving Group-lasso Penalize Learning Problems. *Statistics and Computing* 25(6), 1129–1141.

Yang, Y., H. Zou, and S. Bhatnagar (2020). *gglasso: Group Lasso Penalized Learning Using a Unified BMD Algorithm*. R package version 1.5.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1), 49–67.