Supplementary materials

by Halaleh Kamari, Sylvie Huet and Marie-Luce Taupin

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

$$\widehat{x} \in argmin_{x \in \mathbb{R}^n} F(x) \Leftrightarrow 0 \in \partial F(\widehat{x}).$$

This results from the fact that $F(y) \geq F(\widehat{x}) + < 0, y - \widehat{x} >$ for all $y \in \mathbb{R}^n$ in both cases (Giraud, 2014).

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 $L_X : f \in \mathcal{H} \to f(X) \in \mathbb{R}$ are continuous. The Riesz representation Theorem ensures the existence of an unique element $k_X(.)$ 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 $\langle ., . \rangle_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$. It follows that for all X ,X' in $\mathcal{X}$, and $k_X(.)$, $k_{X'}(.)$ 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 $k : \mathcal{X} \times \mathcal{X}((X, X')) \to \mathbb{R}(k_X(X'))$. 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:

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

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 et al. (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 $a \in \{1, \cdots, d\}$, 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) $k_a : \mathcal{X}_a \times \mathcal{X}_a \to \mathbb{R}$ is $P_a \otimes P_a$ measurable,

(ii) $E_{X_a} \sqrt{k_a(X_a, X_a)} < \infty$.

The property (ii) depends on the kernel k_a, $a = 1, ..., d$ and the distribution of X_a, $a = 1, ..., d$. 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-RKHSs, $\mathcal{H}_a = \mathcal{H}_{0a} \overset{\perp}{\oplus} \mathcal{H}_{1a}$, 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 $\mathcal{H}_{0a}$ is defined by:

$$k_{0a}(X_a, X'_a) = k_a(X_a, X'_a) - \frac{E_{U \sim P_a}(k_a(X_a, U)) E_{U \sim P_a}(k_a(X'_a, U))}{E_{(U,V) \sim P_a \otimes P_a} k_a(U, V)}. \quad (1)$$

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

$$k(X, X') = \prod_{a=1}^{d} \Big(1 + k_{0a}(X_a, X'_a)\Big) = 1 + \sum_{v \in \mathcal{P}} k_v(X_v, X'_v).$$

For $\mathcal{H}_v$ being the RKHS associated with the kernel k_v, the RKHS associated with the ANOVA kernel is then defined by $\mathcal{H} = \prod_{a=1}^{d}(\mathbb{1} \overset{\perp}{\oplus} \mathcal{H}_{0a}) = \mathbb{1} + \sum_{v \in \mathcal{P}} \mathcal{H}_v$, where $\perp$ denotes the L^2 inner product. According to this construction, any function $f \in \mathcal{H}$ satisfies $f(X) = \langle f, k(X,.)\rangle_{\mathcal{H}} = f_0 + \sum_{v \in \mathcal{P}} f_v(X_v)$, 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, $a = 1, ..., d$). This method allows us to choose different approximation spaces independently of the distribution of the input variables $X_1, ..., X_d$, by choosing different sets of kernels. While, in the meta-modelling approach based on the polynomial Chaos expansion, according to the distribution of the input variables $X_1, ..., X_d$, 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).

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:

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

and therefore,

$$\widehat{f}_0 = \frac{1}{n} \sum_{i=1}^{n} Y_i - \frac{1}{n} \sum_{i} \sum_{v} (K_v \theta_v)_i, \tag{2}$$

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

The next step is to calculate $\widehat{\theta} = \text{argmin}_{\theta \in \mathbb{R}^{n \times |\mathcal{P}|}} C_g(f_0, \theta)$. 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 θ_v for the given values of f_0 and θ_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

$$R_v = Y - f_0 - \sum_{w \neq v} K_w \theta_w. \tag{3}$$

We aim to minimize $C_{g,v}(f_0, \theta_v)$ with respect to θ_v. Let $\partial C_{g,v}$ be the sub-differential of $C_{g,v}(f_0, \theta_v)$ with respect to θ_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 $\widehat{t}_v \in$

$\partial\|K_v^{1/2}\theta_v\|$ fulfilling,

$$-2K_v(R_v - K_v\theta_v) + \sqrt{n}\mu_g\widehat{t}_v = 0. \tag{4}$$

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

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

and,

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

Let $\widehat{\theta}_v$ be the minimizer of $C_{g,v}$. The sub-differential equations above give the two following cases:
Case 1. If $\widehat{\theta}_v = 0$, then there exists $\widehat{t}_v \in \mathbb{R}^n$ such that $\|K_v^{-1/2}\widehat{t}_v\| \leq 1$ and it fulfils Equation (4), $2K_vR_v = \sqrt{n}\mu_g\widehat{t}_v$. Therefore, the necessary and sufficient condition for which the solution $\widehat{\theta}_v = 0$ is the optimal one is $2\|K_v^{1/2}R_v\|/\sqrt{n} \leq \mu_g$.
Case 2. If $\widehat{\theta}_v \neq 0$, then $\widehat{t}_v = K_v\widehat{\theta}_v/\|K_v^{1/2}\widehat{\theta}_v\|$ and it fulfils Equation (4),

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

We obtain then,

$$\widehat{\theta}_v = (K_v + \frac{\sqrt{n}\mu_g}{2\|K_v^{1/2}\widehat{\theta}_v\|}I_n)^{-1}R_v. \tag{5}$$

Since $\widehat{\theta}_v$ appears in both sides of the Equation (5), 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 (5) if and only if there exists $\rho > 0$ such that:*

$$\mu_g = \frac{2\rho}{\sqrt{n}}\|K_v^{1/2}\theta(\rho)\|. \tag{6}$$

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

Proof If there exists a non-zero solution to Equation (5), then $\|K_v^{1/2}\widehat{\theta}_v\| \neq 0$ since K_v is positive definite. Take $\rho = \sqrt{n}\mu_g/2\|K_v^{1/2}\widehat{\theta}_v\|$, then $\theta(\rho) = (K_v + \frac{\sqrt{n}\mu_g}{2\|K_v^{1/2}\widehat{\theta}_v\|}I_n)^{-1}R_v = \widehat{\theta}_v$, and, for such ρ Equation (6) is satisfied. Conversely, if there exists $\rho > 0$ such that Equation (6) 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 (5) calculated in $\widehat{\theta}_v = \theta(\rho)$. □

Remark 1 *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 $\widehat{\rho}$, which leads to calculate $\widehat{\theta}(\widehat{\rho})$.*

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 $\|K_v^{1/2}(\frac{K_v}{\rho} + I_n)^{-1}R_v\| \to \|K_v^{1/2}R_v\|$ and $\|2K_v^{1/2}R_v\| > \sqrt{n}\mu_g$. Moreover, we have:

$$y(\rho) = 2\|(\frac{I_n}{\rho} + k_v^{-1})^{-1}k_v^{-1/2}R_v\| - \sqrt{n}\mu_g = 2(X^TA^{-2}X)^{1/2} - \sqrt{n}\mu_g,$$

where $A = (I_n/\rho + k_v^{-1})$ and $X = k_v^{-1/2} R_v$. The first derivative of $y(\rho)$ in ρ is obtained by $\frac{\partial y(\rho)}{\partial \rho} = (X^T A^{-2} X)^{-1/2} \frac{\partial (X^T A^{-2} X)}{\partial \rho}$. Finally, by simple calculations we get,

$$\frac{\partial y(\rho)}{\partial \rho} = \frac{2\|(\frac{I_n}{\rho} + k_v^{-1})^{-3/2} k_v^{-1/2} R_v\|}{\rho^2 \|(\frac{I_n}{\rho} + k_v^{-1})^{-1} k_v^{-1/2} R_v\|} > 0.$$

Therefore $y(\rho)$ is an increasing function of ρ, and the proof is complete. □

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

Algorithm 1 Algorithm to find ρ and $\widehat{\theta}_v$:

if $\widehat{\theta}_{\text{old}} = 0$ **then** ▷ $\widehat{\theta}_{\text{old}}$ is $\widehat{\theta}_v$ computed in the previous step of the RKHS group lasso algorithm.
  Set $\rho \leftarrow 1$ and calculate $y(\rho)$
  if $y(\rho) > 0$ **then**
    Find $\widehat{\rho}$ that minimizes $y(\rho)$ on the interval $[0, 1]$
  else
    repeat
      Set $\rho \leftarrow \rho \times 10$ and calculate $y(\rho)$
    until $y(\rho) > 0$
    Find $\widehat{\rho}$ that minimizes $y(\rho)$ on the interval $[\rho/10, \rho]$
  end if
else
  Set $\rho \leftarrow \sqrt{n}\mu_g / 2\|K_v^{1/2}\widehat{\theta}_{\text{old}}\|$ and calculate $y(\rho)$
  if $y(\rho) > 0$ **then**
    repeat
      Set $\rho \leftarrow \rho/10$ and calculate $y(\rho)$
    until $y(\rho) < 0$
    Find $\widehat{\rho}$ that minimizes $y(\rho)$ on the interval $[\rho, \rho \times 10]$
  else
    repeat
      Set $\rho \leftarrow \rho \times 10$ and calculate $y(\rho)$
    until $y(\rho) > 0$
    Find $\widehat{\rho}$ that minimizes $y(\rho)$ on the interval $[\rho/10, \rho]$
  end if
end if
calculate $\widehat{\theta}_v = \theta(\widehat{\rho})$

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 (2), which is equal to n^2. Given $\widehat{\rho}$, the complexity for $\widehat{\theta}_v$, $v \in \mathcal{P}$, is given by the backsolving of $(K_v + \widehat{\rho} I_n)\widehat{\theta}_v = R_v$ to get $\widehat{\theta}_v$, which is equal to n^2. The computation of $\widehat{\rho}$ is done using Brent-Dekker method implemented as function `gsl_root_fsolver_brent` from Galassi et al. (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.

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 $\widehat{\theta} = \text{argmin}_{\theta \in \mathbb{R}^{n\times|\mathcal{P}|}} C(f_0, \theta)$, 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 θ_v for given values of f_0 and θ_w, $w \neq v$. We aim at minimizing with respect to θ_v,

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

where R_v is defined by (3).

Let ∂C_v be the sub-differential of $C_v(f_0, \theta_v)$ with respect to θ_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 $\widehat{s}_v \in \partial\|K_v\theta_v\|$ and $\widehat{t}_v \in \partial\|K_v^{1/2}\theta_v\|$ such that,

$$-2K_v(R_v - K_v\theta_v) + \sqrt{n}\gamma\widehat{s}_v + n\mu\widehat{t}_v = 0. \tag{7}$$

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,

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

Let $\widehat{\theta}_v$ be the minimizer of the $C_v(f_0, \theta_v)$. Using the sub-differential equations above, the estimator $\widehat{\theta}_v$, $v \in \mathcal{P}$ is obtained following the two cases below:
Case 1. If $\widehat{\theta}_v = 0$, then there exists $\widehat{s}_v \in \mathbb{R}^n$ such that $\|K_v^{-1}\widehat{s}_v\| \leq 1$ and it fulfils Equation (7), $2K_vR_v - n\mu\widehat{t}_v = \sqrt{n}\gamma\widehat{s}_v$, with $\widehat{t}_v \in \mathbb{R}^n$, $\|K_v^{-1/2}\widehat{t}_v\| \leq 1$. Set $J(\widehat{t}_v) = \|2R_v - n\mu K_v^{-1}\widehat{t}_v\|$, and, $J^* = \text{argmin}_{\widehat{t}_v \in \mathbb{R}^n}\{J(\widehat{t}_v)$, such that $\|K_v^{-1/2}\widehat{t}_v\| \leq 1\}$. Then the solution to Equation (7) is zero if and only if $J^* \leq \gamma$.
Case 2. If $\widehat{\theta}_v \neq 0$, then we have $\widehat{s}_v = K_v^2\widehat{\theta}_v/\|K_v\widehat{\theta}_v\|$, and $\widehat{t}_v = K_v\widehat{\theta}_v/\|K_v^{1/2}\widehat{\theta}_v\|$ fulfilling Equation (7), $2K_v(R_v - K_v\widehat{\theta}_v) = \sqrt{n}\gamma\frac{K_v^2\widehat{\theta}_v}{\|K_v\widehat{\theta}_v\|_2} + n\mu\frac{K_v\widehat{\theta}_v}{\|K_v^{1/2}\widehat{\theta}_v\|}$, that is,

$$\widehat{\theta}_v = (K_v + \frac{\sqrt{n}\gamma}{2\|K_v\widehat{\theta}_v\|}K_v + \frac{n\mu}{2\|K_v^{1/2}\widehat{\theta}_v\|}I_n)^{-1}R_v \quad \text{if} \quad \widehat{\theta}_v \neq 0. \tag{8}$$

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

Proposition 2 *(Proposition 8.4 in Huet and Taupin (2017)) For $\rho_1, \rho_2 > 0$, let $\theta(\rho_1, \rho_2) = (K_v + \rho_1K_v + \rho_2I_n)^{-1}R_v$. If $\mu > 0$, there exists a non zero solution to Equation (8) if and only if there exists $\rho_1, \rho_2 > 0$ such that $\gamma = \frac{2\rho_1}{\sqrt{n}}\|K_v\theta(\rho_1, \rho_2)\|$, and $\mu = \frac{2\rho_2}{n}\|K_v^{1/2}\theta(\rho_1, \rho_2)\|$. Then $\widehat{\theta}_v = \theta(\rho_1, \rho_2)$.*

Proof The proof is given in Huet and Taupin (2017). □

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 $\widehat{\rho}_1, \widehat{\rho}_2$, the complexity for $\widehat{\theta}_v$, $v \in \mathcal{P}$, is given by the backsolving of $(K_v + \rho_1K_v + \rho_2I_n)\widehat{\theta}_v = R_v$ to get $\widehat{\theta}_v$, which is equal to n^2. The computation of $\widehat{\rho}_1, \widehat{\rho}_2$, 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_multiroot_fdfsolver_gnewton`, `gsl_multiroot_fsolver_hybrids`, and `gsl_multiroot_fsolver_broyden` from Galassi et al. (2018), respectively. These methods converge fast in general, and for n big enough their complexity is dominated by n^2.

Overview of the RKHSMetaMod functions

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

```
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 et al., 2010) and **GSL** (Galassi et al., 2018) via **RcppEigen** (Bates and Eddelbuettel, 2013) and **RcppGSL** (Eddelbuettel and Francois, 2019) packages are used in the **RKHSMetaMod** package. For different examples of usage of **RcppEigen** and **RcppGSl** 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.

Main RKHSMetaMod functions

Let us begin by introducing some notations. For a given Dmax$\in\mathbb{N}$, let $\mathcal{P}_{\text{Dmax}}$ be the set of parts of $\{1,...,d\}$ with dimension 1 to Dmax. The cardinal of $\mathcal{P}_{\text{Dmax}}$ is denoted by vMax $=\sum_{j=1}^{\text{Dmax}}\binom{d}{j}$.

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

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

- `mu`: value of the tuning parameter μ if $\gamma>0$, or $\mu_g=\sqrt{n}\times\mu$ if $\gamma=0$.
- `gamma`: value of the tuning parameter γ.
- `Meta-Model`: a RKHS ridge group sparse or RKHS group lasso object associated with the tuning parameters `mu` and `gamma`. Table 2 gives a summary of all arguments of the output `Meta-Model` of `RKHSMetMod` function.

`RKHSMetMod_qmax` **function:** For a given value of Dmax and a chosen kernel, this function calculates the Gram matrices K_v, $v\in\mathcal{P}_{\text{Dmax}}$; determines μ, referred to as μ_{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 $\widehat{f}$ associated with the tuning parameter μ_{qmax} and a grid of values of the tuning parameter γ. All the estimators $\widehat{f}$ produced by this function have at most $qmax$ groups in their support. This function has the following input 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 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 γ 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 μ: $\mu = \mu_{\max}/(\sqrt{n} \times \text{frc})$. The value $\mu_{\max}$ is calculated inside the program.
verbose	Logical. Set as TRUE to print: the group v for which the correction of the Gram matrix K_v is done, and for each pair of the tuning parameters (μ, γ): the number of current iteration, active groups and convergence criteria. It is set as FALSE by default.

Table 1: List of the input arguments of the `RKHSMetMod` 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 θ_v for $v = 1, ...,$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 $\Vert Y - f_0 - \sum_v K_v\theta_v \Vert^2$.
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. $\Vert \theta_{lastIter} - \theta_{lastIter-1}/\theta_{lastIter-1} \Vert^2$.
RelDiffPar	Scalar, value of the second convergence criterion at the last iteration, i.e. $crit_{lastIter} - crit_{lastIter-1}/crit_{lastIter-1}$.

Table 2: List of the arguments of the output `Meta-Model` of `RKHSMetMod` function.

- `Y`, `X`, `kernel`, `Dmax`, `gamma`, `verbose` (see Table 1).
- `qmax`: integer, the maximum number of groups in the support of the obtained estimator.
- `rat`: positive scalar, to restrict the minimum value of μ considered in "Algorithm 3",

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

  where $\mu_{\max}$ is calculated inside the program.
- `Num`: integer, to restrict the number of different values of the tuning parameter μ to be evaluated in the RKHS group lasso algorithm until it achieves μ_{qmax}. For example, if

`Num` equals 1, the program is implemented for three different values of $\mu \in [\mu_{\min}, \mu_{\max})$:

$$\mu_1 = \frac{(\mu_{\min} + \mu_{\max})}{2},$$
$$\mu_2 = \begin{cases} \frac{(\mu_{\min}+\mu_1)}{2} & \text{if} \quad |\widehat{S}_{\widehat{f}(\mu_1)_{\text{Group Lasso}}}| < qmax, \\ \frac{(\mu_1+\mu_{\max})}{2} & \text{if} \quad |\widehat{S}_{\widehat{f}(\mu_1)_{\text{Group Lasso}}}| > qmax, \end{cases}$$
$$\mu_3 = \mu_{\min},$$

where $|\widehat{S}_{\widehat{f}(\mu_1)_{\text{Group Lasso}}}|$ is the number of groups in the support of the solution of the RKHS group lasso problem associated with μ_1.

If `Num` > 1, the path to cover the interval $[\mu_{\min}, \mu_{\max})$ is detailed in "Algorithm 3".

The `RKHSMetMod_qmax` function returns a list of three components `mus`, `qs`, and `MetaModel`:

- `mus`: vector of all values of μ_i in "Algorithm 3".
- `qs`: vector with the same length as `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 `mus`.
- `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`:
  - `mu`: value of μ_{qmax}.
  - `gamma`: element of the input vector `gamma` associated with the estimated `Meta-Model`.
  - `Meta-Model`: a RKHS ridge group sparse or RKHS group lasso object associated with the tuning parameters `mu` and `gamma` (see Table 2).

Companion functions

`calc_Kv` **function:** For a given value of Dmax and a chosen kernel, this function calculates the Gram matrices K_v, $v \in \mathcal{P}_{\text{Dmax}}$, and returns their associated eigenvalues and eigenvectors. This function has,

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

The `calc_Kv` function returns a list of two components `kv` and `names.Grp`:

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

`RKHSgrplasso` **function:** For a given value of the tuning parameter μ_g, this function fits the solution to the RKHS group lasso optimization problem. This function has,

- three mandatory input arguments:
  - `Y` (see Table 1).
  - `Kv`: list of the eigenvalues and the eigenvectors of the positive definite Gram matrices K_v for $v = 1, ...,$vMax and their associated group names (output of the function `calc_Kv`).
  - `mu`: positive scalar indicates the value of the tuning parameter $\mu_g = \sqrt{n}\mu$.
- two facultative input arguments:
  - `maxIter`: integer, to set the maximum number of loops through all groups. It is set as 1000 by default.
  - `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 μ_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 2).

`mu_max` **function:** This function calculates the value $\mu_{\max}$. 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 $v = 1, ...,$vMax. This function returns the μ_{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 μ, γ. Each RKHS meta-model in the sequence is the solution to the RKHS ridge group sparse optimization problem associated with a pair of values of (μ, γ) in the grid of values of μ, γ. This function has,

- seven mandatory input arguments:
  - `Y` (see Table 1).
  - `gamma`: vector of positive scalars. Values of the penalty parameter γ in decreasing order.
  - `Kv`: list of the eigenvalues and the eigenvectors of the positive definite Gram matrices K_v for $v = 1, ...,$vMax and their associated group names (output of the function `calc_Kv`).
  - `mu`: vector of positive scalars. Values of the tuning parameter μ 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. 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 (μ, γ), 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 (μ, γ). 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 2).

`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 1).
– `XT`: matrix of observations of the testing dataset with n^{test} rows and d columns.
– `YT`: vector of response observations of the testing dataset of size n^{test}.
– `mu`: vector of positive scalars. Values of the tuning parameter μ in decreasing order.
– `res`: list of the estimated RKHS meta-models for the learning dataset associated with the tuning parameters (μ, γ) (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 1).
– `Xnew`: matrix of new observations with n^{new} rows and d columns.
– `res`: list of the estimated RKHS meta-models for a learning dataset associated with the tuning parameters (μ, γ) (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.

`SI_emp` **function:** For each RKHS meta-model $\widehat{f}$, this function calculates the empirical Sobol indices for all the groups that are active in the support of $\widehat{f}$. 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`.

Bibliography

N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950. URL http://dx.doi.org/10.2307/1990404. [p1]

D. Bates and D. Eddelbuettel. Fast and Elegant Numerical Linear Algebra Using the RcppEigen Package. *Journal of Statistical Software*, 52(5):1–24, 2013. URL http://www.jstatsoft.org/v52/i05/. [p6]

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer US, 2003. ISBN 9781402076794. doi: https://doi.org/10.1007/978-1-4419-9096-9. [p1]

G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of computational Physics*, 230:2345–2367, 2011. [p2]

N. Durrande, D. Ginsbourger, O. Roustant, and L. Carraro. ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57 – 67, 2013. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2012.08.016. URL http://www.sciencedirect.com/science/article/pii/S0047259X1200214X. [p1]

D. Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer Publishing Company, Incorporated, 2013. ISBN 1461468671, 9781461468677. [p6]

D. Eddelbuettel and R. Francois. *RcppGSL: 'Rcpp' Integration for 'GNU GSL' Vectors and Matrices*, 2019. URL https://cran.r-project.org/web/packages/RcppGSL/. R package version 0.3.7. [p6]

M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich. GNU Scientific Library Reference Manual, 2018. URL http://www.gnu.org/software/gsl/. [p4, 6]

C. Giraud. *Introduction to High-Dimensional Statistics*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2014. ISBN 9781482237948. URL https://books.google.fr/books?id=qRuVoAEACAAJ. [p1]

G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. [p6]

S. Huet and M.-L. Taupin. Metamodel construction for sensitivity analysis. *ESAIM: Procs*, 60:27–69, 2017. doi: 10.1051/proc/201760027. URL https://doi.org/10.1051/proc/201760027. [p5]

H. Kamari. *RKHSMetaMod: Ridge Group Sparse Optimization Problem for Estimation of a Meta Model Based on Reproducing Kernel Hilbert Spaces*, 2019. URL https://CRAN.R-project.org/package=RKHSMetaMod. R package version 1.1. [p6]

S. Saitoh. *Theory of reproducing kernels and its applications*. Pitman research notes in mathematics series. Longman Scientific & Technical, 1988. ISBN 9780582035645. URL https://books.google.fr/books?id=YzxPAQAAIAAJ. [p1]

Halaleh Kamari
UMR 8071 Laboratoire de Mathématiques et Modélisation d'Évry (LaMME)
Bâtiment IBGBI de l'Université d'Evry Val d'Essonne, 23 Bd de France, 91037 Evry CEDEX
France
halala.kamari@gmail.com

Sylvie Huet
UR 1404 Mathématiques et Informatique Appliquées du Génome à l'Environnement (MaIAGE)
Bâtiment 210 de INRAE, Domaine de Vilvert, 78352 JOUY-EN-JOSAS Cedex
France
shuet.inra@gmail.com

Marie-Luce Taupin
UMR 8071 Laboratoire de Mathématiques et Modélisation d'Évry (LaMME)
Bâtiment IBGBI de l'Université d'Evry Val d'Essonne, 23 Bd de France, 91037 Evry CEDEX
France
marie-luce.taupin@univ-evry.fr