A specific kriging kernel for dimensionality reduction: Isotropic by group kernel

Christophette Blanchet-Scalliet, Céline Helbert, Mélina Ribaud, Céline Vial

Four algorithms to construct a sparse kriging kernel for dimensionality reduction

Christophette Blanchet-Scalliet[1], Céline Helbert[1], Mélina Ribaud*[1], and Céline Vial[2,3]

[1]Univ Lyon, École centrale de Lyon, CNRS UMR 5208, Institut Camille Jordan, 36 avenue Guy de Collonge, F-69134 Ecully Cedex, France
[2]Univ Lyon, Université Claude Bernard Lyon 1, CNRS UMR 5208, Institut Camille Jordan, 43 blvd. du 11 novembre 1918, F-69622 Villeurbanne cedex, France
[3]INRIA, Villeurbanne, France

April 26, 2017

Abstract

In the context of computer experiments, metamodels are largely used to represent the output of computer codes. Among these models, Gaussian process regression (kriging) is very efficient see e.g Snelson (2008). In high dimension that is with a large number of input variables, but with few observations the classical *anisotropic* kriging becomes inefficient and sometimes completely wrong. One way to overcome this drawback is to use the *isotropic* kernel which is more robust because it estimates not as many parameters. However this model is too restrictive. The aim of this paper is twofold. Our first objective is to propose a smooth kernel with as few parameters as warranted. We propose a kernel which is a tensor product of few isotropic kernels built on well-chosen subgroup of variables. The main difficulty is to find the number and the composition of groups. Our second objective is to propose algorithmic strategies to overcome the difficulty of finding the number and the composition of the groups. Four forward strategies are proposed. They all start with the simplest isotropic kernel and stop when the best model according to BIC criterion is found. They all show very good accuracy results on simulation test cases. But one of them is the most efficient. Tested on a real data set, our kernel shows very good prediction results.

Keywords. kriging, metamodel, kernel, isotropic, anisotropic, algorithms, dimension reduction, clustering

1 Introduction

Complex physical phenomena are more and more studied through numerical simulations. These numerical models are able to mimic with a high accuracy the real experiments so they predict the physical measures of interest (outputs) very precisely. Then, we can use them as a

*melina.ribaud@doctorant.ec-lyon.fr

replacement for real experiments because the numerical simulations are less costly in primary materials. However, these simulations stay often time-consuming. The idea to overcome this drawback is to replace the costly numerical model by a metamodel. A metamodel is a less expensive model adjusted on a few well-chosen simulations. It can be shown that among all the possible metamodels, Gaussian process regression (kriging) is a very efficient metamodel. Lots of examples of the use of a metamodel can be found in litterature, see e.g. Marrel et al. (2008), Antoniadis et al. (2012) and Sudret (2012). A detailed example of kriging is the helicopter test displayed in Booker et al. (1998). Furthermore Villa-Vialaneix et al. (2012) shows a comparative study of eight metamodeling techniques for the simulation of N_20 fluxes and N (Nitrogen) leaching from corn crops. In this context, Splines and kriging have the best performances for small and medium training datasets. In addition, kriging is able to model highly complex data, see e.g. Santner et al. (2003) and Rasmussen and Williams (2006). More precisely kriging is a spatial interpolation technique which aims at predicting the outputs using an adapted underlying correlation function between design points. In fact we assume that the output of interest is a realization of a Gaussian process (GP) constructed as a sum of a deterministic part (often called the *trend*) and a stochastic part assumed to be a zero-mean stationary GP, see Roustant et al. (2012). In this paper, we focus on the stochastic part and more precisely on the choice of the covariance kernel structure. In general, people use an *anisotropic* kernel that is a tensor product of as many 1D kernels as the number of inputs. Each kernel being parameterized by a spatial correlation length, called the *range parameter*. In high dimension with a very restricted number of data points the estimation of range parameters becomes quite difficult. That's why in this context, an *isotropic* kernel could be a good alternative. This kernel is a function of the euclidean distance defined on the entire input space, so it depends only on one range parameter. However *isotropic* kernels are too restrictive, given that spatial variations are controlled by only one range parameter. The idea of this paper is to automatically construct a data-driven kernel that is intermediate between these two extremal choices.

A review of the literature shows that one way to improve performance of kriging is to adapt the covariance structure to each specific case, see e.g. Durrande (2001) and Ginsbourger et al. (2016). For example Paciorek and Schervish (2006) create a new class of covariance functions (kernels) allowing the model to adapt itself to spatial surface whose variability changes with location. Likewise, Padonou and Roustant (2016) in a microelectronic framework define a GP model which inserts the geometry of the wafer in the kernel. That's why they introduce the *polar GP* defined with respect to polar coordinates: the covariance function is a sum of a product kernel of radius and a product kernel of polar angles. In the case of multiple outputs Fricker et al. (2013) define a nonseparable covariance structure for GP with the aim of well representing the different simulator outputs and the joint uncertainty. In our case, the idea is to use a covariance structure adapted to high dimension

In general the covariance function only depends on the range parameters and on the sill parameter (variance) which have to be estimated. But range parameters estimation becomes very tricky in high dimension. To overcome this difficulty, Yi (2009) proposed a penalized kriging that mimics the penalization techniques used in linear regression. In fact, he penalizes the likelihood by the norm of the range parameters vector. This methodology selects the variables that have no effect on the covariance function and the associate range parameters are blowing up to infinity. In order to solve the dimensionality of the problem, Binois et al. (2015) proposes different sparse and specific covariance kernels based on expert information or on the work of Muehlenstaedt et al. (2012) and Durrande et al. (2012). In these articles, they use the ANOVA

kernels that are a sum of sub-groups of variables that interact together, see more details in Stitson et al. (1997) and Gao et al. (2002). Muehlenstaedt et al. (2012) found these groups through a sensitivity analysis computed from the prediction function of an *anisotropic* kriging. A first problem appears here. Because the number of points is often too restrictive, the quality of the anisotropic kriging can be poor and the result of the sensitivity analysis such as the proposed partition can be totally wrong. A second problem of the method proposed by Muehlenstaedt and coauthors is that the number of parameters of the final kernel can be higher than that of the *anisotropic* kriging. In Binois et al. (2015) they test different FANOVA constraining them to have fewer range parameters than the *anisotropic* one and choosing the best kernel among them. But this choice of the covariance structure can not be done automatically.

Finally, we can also find in literature the work of Welch et al. (1992) that aims at identifying the active factors. Welch et al. (1992) proposes a likelihood-based forward algorithm to determine the most important factors and to build the predictor. In this study a tensor product of power exponential kernels is chosen to catch the function regularity. The dimension of the kernel, i.e. two parameters (range and power) in each direction, is stepwise reduced by making some range parameters equal. In our paper the structure of the kernel is different. It is a tensor product of isotropic kernels by taking an euclidean norm for the distance in each subspace. We propose to use a Matern kernel of order $5/2$ that needs only one parameter (range), that models intermediate regularity functions and that gives efficient and good results in industrial cases, see Cornford et al. (2002).

The aim of our article is to propose four algorithms that automatically construct from data a sparse kernel adapted to high dimension. The article is structured as follows. In section 2, we introduce the kriging metamodel and the different kernels. In section 3, we present the four algorithms that automatically select the structure of the kernel according to the data. In section 4, we study the behavior of our algorithms on simulation test cases.

2 Statistical models

This section introduce the kriging, *anisotropic* and *isotropic* kernels and finally the new general class of *isotropic by group* kernel.

2.1 Kriging

Let p be the number of input variables. We consider n observations $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ where $\mathbf{x}_i$ is the ith input vector with p coordinates, such that $\mathbf{x} \in \mathbb{R}^p$ and where y_i is the corresponding output ($y_i \in \mathbb{R}$). In the following we denote by $\mathbf{y} = (y_1, \dots, y_n)'$ the vector of the outputs. In kriging we assume that $\mathbf{y}$ is a realization of a Gaussian process $(Y(\mathbf{x}))_{\mathbf{x}\in\mathbb{R}^p}$ at points $(\mathbf{x}_1, \dots, \mathbf{x}_n)'$ such that for each $x \in \mathbb{R}^p$:

$$Y(\mathbf{x}) = m + \epsilon(\mathbf{x}), \tag{1}$$

where $m \in \mathbb{R}$ is the trend, the process $(\epsilon(\mathbf{x}))_{\mathbf{x}\in\mathbb{R}^p}$ is a centered stationary Gaussian process with covariance function $Cov(\epsilon(\mathbf{x}),\epsilon(\mathbf{x}')) = \sigma^2 R(\mathbf{x}, \mathbf{x}') = \sigma^2 \mathbf{r}(\mathbf{x} - \mathbf{x}')$, $\forall\, (\mathbf{x}, \mathbf{x}') \in \mathbb{R}^p \times \mathbb{R}^p$. In this paper the trend m and the variance σ^2 are assumed to be constant.
In this context we want to construct a linear prediction that minimizes the mean-squared prediction error and that guarantees uniform unbiasedness. Under these two constrains, the prediction, see Cressie (1993)), at a point $\mathbf{x_0} \in \mathbb{R}^p$ is given by :

$$\widehat{Y}(\mathbf{x_0}) = m + \mathbf{r}(\mathbf{x_0})'\boldsymbol{R}^{-1}(\mathbf{y} - m\mathbb{1}_n) \tag{2}$$

and the Mean Square Error (MSE) at point $\mathbf{x_0} \in \mathbb{R}^p$ is given by :

$$\widehat{s}(\mathbf{x_0}) = \sigma^2(1 - \boldsymbol{r}(\mathbf{x_0})'\boldsymbol{R}^{-1}\boldsymbol{r}(\mathbf{x_0})') \tag{3}$$

where $\mathbb{1}_n = (1, \ldots, 1)' \in \mathbb{R}^n$, $\boldsymbol{R} \in \mathcal{M}_{n\times n}$ is the correlation matrix of the process (Y) at the observation points, $\mathbf{r}(\mathbf{x_0}) \in \mathcal{M}_{n\times 1}$ the correlation vector between $Y(\mathbf{x_0})$ and the random vector $(Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n))'$.

In the following we will focus on the definition of the correlation function $\mathbf{r}$. First, we introduce the *anisotropic* kernel :

$$\mathbf{r}(\mathbf{x} - \mathbf{x}') = \prod_{j=1}^{p} \rho_{\theta_j}\left(|x_j - x'_j|\right), \ \boldsymbol{\theta} = (\theta_1, \ldots, \theta_p) \in \mathbb{R}^p_+ \tag{4}$$

where ρ_{θ_j} is a correlation function which only depends on the one dimensional range parameter θ_j, see e.g Santner et al. (2003) and Stein (1999). In the following, we denote the model (1) with kernel (4) as $\mathcal{M}_a$. The parameters m, σ^2 and $\boldsymbol{\theta}$ are unknown and will be estimated by maximum likelihood.

In our industrial case study the output is supposed to be two times continuously differentiable this is why we use a Matern $5/2$ kernel usual in that case, see e.g. Cornford et al. (2002) . This kernel function is defined by :

$$\forall \theta \in \mathbb{R}^+, \forall h \in \mathbb{R}^+, \rho_\theta(h) = \left(1 + \frac{\sqrt{5}|h|}{\theta} + \frac{5h^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}|h|}{\theta}\right).$$

Other examples can be found in Rasmussen and Williams (2006). However, *anisotropic* kernels contain as many parameters as the number of variables p. In high dimension the estimation of these parameters becomes unstable. To stabilize the estimation an idea is to reduce the number of range parameters. The extremal case is the *isotropic* kernel, defined as follows for $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^p$:

$$r(\mathbf{x} - \mathbf{x}') = \rho_\theta\left(\|\mathbf{x} - \mathbf{x}'\|_2\right), \ \theta \in \mathbb{R}_+ \tag{5}$$

where $\|.\|_2$ is the euclidian norm on $\mathbb{R}^p$. In the following, we denote the model (1) with kernel (5) as $\mathcal{M}_i$. In model $\mathcal{M}_i$, spatial variations are controlled by only one range parameter. If the value of that parameter is small the process Y fluctuates a lot in all directions. On the contrary if the value of θ is large, the process varies in a very slow way in all directions. That's why, one unique parameter is often too restrictive to characterize the underlying process. In many cases spatial behavior varies from a direction to another. Therefore we introduce in the next section a kernel which is a compromise between the too restrictive *isotropic* kernel and the too flexible *anisotropic* one. We call it the *isotropic by group* kernel.

2.2 *Isotropic by group* kernel

The isotropic by group kernel is defined as follows:

$$r(\mathbf{x} - \mathbf{x}') = \prod_{\ell=1}^{q} \rho_{\theta_\ell}\left(\|\mathbf{x} - \mathbf{x}'\|_{\mathcal{I}_\ell}\right), \ \boldsymbol{\theta} = (\theta_1, \ldots, \theta_q) \in \mathbb{R}^q_+ \tag{6}$$

where, $q < p$ is the number of groups. Let $\mathcal{I}_\ell$ be the set of indexes in the ℓth group with cardinal $|\mathcal{I}_\ell| = p_\ell$, $p_\ell \in \{1, \ldots, p\}$ such that $p_1 + \ldots + p_q = p$. We define the norm of the subvector $(x_j)_{j\in\mathcal{I}_\ell}$ of the vector $\mathbf{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$ by $\|\mathbf{x}\|_{\mathcal{I}_\ell} = \sqrt{\left(\sum_{j\in\mathcal{I}_\ell} x_j^2\right)}$. In the following, we

denote model (1) with kernel (6) as $\mathcal{M}_q$. The *anisotropic* and *isotropic* kernel are two particular cases of *isotropic by group* kernel with respectively p groups $\mathcal{I}_\ell = \{\ell\}$ for $\ell = \{1, \ldots, p\}$, and one unique group $\mathcal{I}_1 = \{1, \ldots, p\}$, ie $\mathcal{M}_p = \mathcal{M}_a$ and $\mathcal{M}_1 = \mathcal{M}_i$.

In appendix, we simulate with the same seed a Gaussian Process in 2 dimensions with an anisotropic and an isotropic kernel ($\mathcal{M}_a$ and $\mathcal{M}_i$) to visualize the impact of the choice of the kernel .

3 Methodology

In this article we propose four strategies to find simultaneously and automatically the number of groups and the composition of each group. In Yi (2009) and Muehlenstaedt et al. (2012) the two proposed methodologies to reduce the number of parameters in kriging start with a fully *anisotropic* kernel. To gain in robustness in high dimensional problems, in the four procedures we introduce, we always start with an *isotropic* kernel that depends on a unique range parameter and we finish with an *isotropic by group* kernel. At each stage we compare different models and we choose the best model under a specific criterion. The model parameters are estimated by maximum likelihood, that is:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[l(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}) \right] \tag{7}$$

where $l = -\log(\mathcal{L})$ and $\mathcal{L}$ is the likelihood function defined by:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}) = \frac{1}{(2\pi\hat{\sigma}^2)^{\frac{n}{2}} |\boldsymbol{R}|^{\frac{1}{2}}} e^{-(\mathbf{y}-\hat{m}\mathbb{1}_n)^t \boldsymbol{R}^{-1} (\mathbf{y}-\hat{m}\mathbb{1}_n)} \tag{8}$$

where $\hat{m}$ and $\hat{\sigma}^2$ could be writen as functions of $\boldsymbol{\theta}$:

$$\hat{m} = (\mathbb{1}_n^t \boldsymbol{R}^{-1} \mathbb{1}_n)^{-1} \mathbb{1}_n^t \boldsymbol{R}^{-1} \mathbf{y}$$
$$\hat{\sigma}^2 = \frac{1}{n} \left(\mathbf{y} - m\mathbb{1}_n\right)^t \boldsymbol{R}^{-1} \left(\mathbf{y} - m\mathbb{1}_n\right)$$

Equation (7) shows that the estimation of the range parameter $\boldsymbol{\theta}$ is done by minimizing l. The minimization of l is very difficult due to the nonlinearity of the objective function. We use a classical numerical method names Limited-memory BFGS to handle optimization. A well known drawback of the optimization algorithms is the dependence of the solution on the initialization point. That's why we create the **Algorithm** 0 that finds a good initial value for $\boldsymbol{\theta}$ to minimize l and that gives an estimation of $\hat{\boldsymbol{\theta}}$. The principle of **Algorithm** 0 is to first select 10 vectors of parameters $\boldsymbol{\theta}$ ($\boldsymbol{\theta} \in \mathbb{R}^q$) that cover space as well as possible and to choose the set of parameters, says $\boldsymbol{\theta}_{opt\_init}$, that has the lowest l-value. Then, we minimize l using $\boldsymbol{\theta}_{opt\_init}$ as initial point and we obtain the estimation of $(\theta_\ell)_{\ell=1,\ldots,q}$, denoted by $\boldsymbol{\theta}_{opt}$.

Algorithm 0 Likelihood Optimization

$crit\_vrais = \mathbf{0}_{10}$;
$\boldsymbol{\theta}_{\boldsymbol{Mat},init} = \mathbf{0}_{q\times 10}$, where $\mathbf{0}_{q\times 10}$ is the zero matrix of size $q \times 10$;
for $k = 1$ to 10 **do**
    $\boldsymbol{\theta}_{\boldsymbol{Mat},init}(k) \rightsquigarrow \mathcal{U}([0;+\infty]^q)$;
    $crit\_vrais(k) \leftarrow l_{\mathcal{M}_2}(\boldsymbol{\theta}_{\boldsymbol{Mat},init}(k); \mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n)$;
end for
$k_{opt} \leftarrow \underset{k}{\text{argmin}}\{crit\_vrais(k)\}$;
$\boldsymbol{\theta}_{opt\_init} \leftarrow \boldsymbol{\theta}_{\boldsymbol{Mat},init}(k_{opt})$;
$\boldsymbol{\theta}_{opt} \leftarrow \underset{\boldsymbol{\theta}\in\mathcal{D}}{\text{argmin}}\, l_{\mathcal{M}_2}(\boldsymbol{\theta}_{opt\_init}; \mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n)$ with the initialization at $\boldsymbol{\theta}_{opt\_init}$.

As we estimate parameters by maximum likelihood, it could then be natural to select the best model under likelihood considerations. However the likelihood is an increasing function of the number of parameters. As a trade off between estimation qualities and sparsity, a penalized likelihood criterion is preferred. In our work, we consider the BIC criterion (see e.g. Schwarz (1978)), that is for a model $\mathcal{M}_q$ with q parameters to estimate $\boldsymbol{\theta}_q$:

$$BIC(\boldsymbol{\theta}_q, \mathcal{M}_q) = l_{\mathcal{M}_q}(\boldsymbol{\theta}_q; \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}) + q\log(n). \tag{9}$$

In the four strategies, the best model corresponds to the one minimizing the BIC criterion. Another usual criterion for selecting the best model is based on predictive considerations, called the cross-validation criterion. But this criterion is very time consuming as the estimation in the context of high dimensional data. So we decide to not consider this criterion.

3.1 Algorithm 1

General loop:
At each step k, if the previous model is composed of $k-1$ groups, the next one will have k groups. Let $\mathcal{M}_{k-1}$ be the model at step $k-1$, we note $\mathcal{I}_1 = \{i_{1,1}, \ldots, i_{1,p_1}\}; \ldots; \mathcal{I}_k = \{i_{k-1,1}, \ldots, i_{k-1,p_{k-1}}\}$, with $p_1 + \ldots + p_{k-1} = p$, the sets of indexes of the variables included in each group. One of these $k-1$ groups is then divided into two groups. We have J_k ways of choosing this group, with $J_k = \sum_{\ell=1}^{k-1} J_{k,\ell}$ and $J_{k,\ell}$ is the number of ways of dividing into two groups the set of indexes $\mathcal{I}_\ell$, $\ell = 1, \ldots, k-1$. We note $\mathcal{M}_k^1, \ldots, \mathcal{M}_k^{J_k}$ the J_k new models. And among these J_k models we choose the one minimizing the BIC, says $\mathcal{M}_k^{\ell_{opt}}$.

Algorithm 1:
We start with the estimation of the full *isotropic* model, noted $\mathcal{M}_1$. We follow the previous procedure and we obtain J_2 models noted $\mathcal{M}_2^\ell$, $\ell = 1, \ldots, J_2$. We estimate the parameter $\boldsymbol{\theta}$ of each model and we keep the model $\mathcal{M}_2^{\ell_{opt}}$ that minimizes the BIC. If the BIC of the model $\mathcal{M}_1$ is smallest than the one of $\mathcal{M}_2$, we stop the algorithm and we choose $\mathcal{M}_1$ as the best model. Otherwise, we set down $\mathcal{M}_2 = \mathcal{M}_2^{\ell_{opt}}$ and we build the models $\mathcal{M}_3^\ell$ $l = 1, \ldots, J_3$ and so forth at step $4, 5, \ldots$. We stop the algorithm when $\mathcal{M}_k$ has a larger BIC than $\mathcal{M}_k$. At the end of the algorithm, we obtain the model $\mathcal{M}_q$ that minimizes the BIC.

Algorithm 1 Find $\mathcal{M}_q$

We choose θ_{opt} thanks to algorithm 0 for model $\mathcal{M}_1$.
$opt\_BIC \leftarrow BIC(\theta_{opt}, \mathcal{M}_1)$;
$k \leftarrow 2$;
while $k \leq p$ **do**
    $vect\_BIC \leftarrow \mathbf{0}_{J_k}$
    for ℓ de 1 à J_k **do**
        We choose θ_{opt} thanks to algorithm 0 with the model $\mathcal{M}_k^\ell$;
        $vect\_BIC(\ell) \leftarrow BIC(\boldsymbol{\theta}_{opt}, \mathcal{M}_{k+1}^\ell)$;
    end for
    $\ell_{opt} \leftarrow \underset{\ell}{\text{argmin}}\{vect\_BIC\}$;
    $\mathcal{M}_k \leftarrow \mathcal{M}_k^{\ell_{opt}}$;
    if $vect\_BIC(\ell_{opt}) \geq opt\_BIC$ **then**
        $\mathcal{M}_q \leftarrow \mathcal{M}_{k-1}$;
        return $\mathcal{M}_q$;
    else
        $opt\_BIC \leftarrow vect\_BIC(\ell_{opt})$;
        $k \leftarrow k + 1$;
    end if
end while

Algorithm 1 is a quasi-exhaustive algorithm since at each step it browses all the possibilities to divide. But the calculation time grows really fast with the number of inputs p. For example, the first separation, that consists in separating one group in two, estimates 500 models for $p = 10$ and 524287 for $p = 20$. That's why we propose several alternative algorithms that take much less time.

3.2 Algorithm 2

We propose a faster algorithm close to the forward **Algorithm** W introduced by Welch et al. (1992) that is described in appendix. The direction of **Algorithm** 2 is both because it has a backward step contrary to **Algorithm** W. **Algorithm** W is included in **Algorithm** 2.
We note $\mathcal{M}_{s_{k-1}}$ the model at step $k-1$ and s_{k-1} the number of groups. Let $\mathcal{I}_1 = \{i_{1,1}, \ldots, i_{1,p_1}\}$; ... ; $\mathcal{I}_{s_{k-1}} = \{i_{s_{k-1},1}, \ldots, i_{s_{k-1},p_{s_{k-1}}}\}$, with $p_1 + \ldots + p_{s_{k-1}} = p$, the variables' indexes of each group.

Checking step:
At step k, there are 2 possibilities :

(i) If we have grouped some variables at the step $k-1$ or if the current step is $k < 3$ (aggregation is not allowed before $k = 3$), we create an additional group by taking an index $i_{1,m} \in \mathcal{I}_1$, $m = 1, \ldots, p_1$ out of the group 1. We have $J_k = J_k^+ = p_1$ possibilities and so we build J_k models.

(ii) If we have separated at the step $k-1$ and $k \geq 3$, we have 2 possibilities. The first one consists in taking an index $i_{1,s} \in \mathcal{I}_1$, $s = 1, \ldots, p_1$ out of the first group, we have $J_k^+ = p_1$ possible indexes. The second consists in grouping the last taken out variable $i_{s_{k-1},1}$ ($p_{s_{k-1}} = 1$) with an other group, we have $J_k^- = s_{k-1} - 2$ possibilities. Then we build a total of $J_k = J_k^+ + J_k^- = p_1 + s_{k-1} - 2$ models.

In the two cases, we create J_k models noted $\mathcal{M}^1_{s_{k_1}}, \ldots, \mathcal{M}^{J_k}_{s_{k_{J_k}}}$.

General loop:
We follow the general procedure in introducing a boolean that represents the choice at step $k-1$ of grouping or separating. Then the value of the boolean will be TRUE if we grouped at the previous step and FALSE if not.

Algorithm 2 Find $\mathcal{M}_q$

We choose θ_{opt} with algorithm 0 in considering model $\mathcal{M}_1$.
$opt\_BIC \leftarrow BIC(\theta, \mathcal{M}_1)$;
$k \leftarrow 2$;
while $k \leq p$ **do**
  if $reg_{bool} = false$ and $k \geq 3$ **then**
    We construct $J_k = p_1 + s_{k-1} - 2$ models $\mathcal{M}^\ell_{s_{k_\ell}}$, $\ell = 1, \ldots, J_k$;
  else
    We construct $J_k = p_1$ models $\mathcal{M}^\ell_{s_{k_\ell}}$, $\ell = 1, \ldots, J_k$;
  end if
  $vect\_BIC \leftarrow \mathbf{0}_J$
  for ℓ from 1 to J_k **do**
    We choose θ_{opt} with algorithm 0 in considering model $\mathcal{M}^\ell_{s_{k_\ell}}$;
    $vect\_BIC(k) \leftarrow BIC(\theta, \mathcal{M}^\ell_{s_{k_\ell}})$;
  end for
  $\ell_{opt} \leftarrow \underset{\ell}{\text{argmin}}\{vect\_BIC\}$;
  $\mathcal{M}_{s_k} \leftarrow \mathcal{M}^{\ell_{opt}}_{s_{k_\ell}}$;
  if $vect\_BIC(\ell_{opt}) \geq opt\_BIC$ **then**
    $\mathcal{M}_q \leftarrow \mathcal{M}_{s_k}$;
    return $\mathcal{M}_q$;
  else
    $opt\_BIC \leftarrow vect\_BIC(\ell_{opt})$;
  end if
end while

Algorithm 2 divides more than **Algorithm** 1 and browses less possibilities. This algorithm is largely identical to **Algorithm** W introduced by Welch Welch et al. (1992) except that we add a backward step.

3.3 Algorithm 3

Because **Algorithm** 2 generates a number of clusters often too high, we propose to add a backward step at the end of the procedure. This step naturally reduces the complexity of the model.
General loop:
Let $\mathcal{M}_q$ the best model found by **Algorithm** 2, such that $\mathcal{I}_1 = \{i_{1,1}, \ldots, i_{1,p_1}\}; \ldots; \mathcal{I}_q = \{i_{q,1} \ldots, i_{q,p_q}\}$, with $p_1 + \ldots + p_q = p$ the indexes of variables in each groups. We build a new model noted $\mathcal{M}_{clust}$ with a clustering method that groups variables with close ranges (see Everitt et al. (2011)). However, in the *isotropic by group* kernel the ranges are not comparable since they are linked to variable sets of different sizes. So we introduce the following kernel ,

Algorithms	alg1	alg2	alg3	alg4	algW
Strategy	Quasi exhaustive	Separate	Separate + Cluster at the end	Separate + Cluster at each step	Separate
Direction	Forward	Both	Both	Both	Forward

Table 1: Strategy and direction of the 4 algorithms and algorithm proposed by Welch et al. (1992)

called the *product* kernel, where range parameters can be compared :

$$r_{\boldsymbol{\theta}}(\mathbf{x}-\mathbf{x}') = \prod_{\ell=1}^{q}\prod_{j=1}^{p_\ell} \rho_{\theta_\ell}\left(|x_j - x'_j|\right) \tag{10}$$

It is a tensor product kernel with some equal range parameters. In the following, we denote the model (10) with kernel (4) as model $\mathcal{M}_q^{prod}$. Then we estimate the model $\mathcal{M}_q^{prod}$ with the same groups than the model $\mathcal{M}_q$. We build the model $\mathcal{M}_{clust}$ with a clustering method that groups variables with close range parameter values. Then, if the BIC of the model $\mathcal{M}_{clust}$ is smaller that the one of $\mathcal{M}_q$ we set down $\mathcal{M}_q = \mathcal{M}_{clust}$. We replace the step 19 in **Algorithm** 2 by:

Algorithm 3 Find $\mathcal{M}_q$
Step 19bis:

We obtain $\mathcal{M}_{clust}$ from $\mathcal{M}_q$ with a clustering method;
if $BIC(\theta, \mathcal{M}_{clust}) < BIC(\theta, \mathcal{M}_q)$ **then**
  $\mathcal{M}_q \leftarrow \mathcal{M}_{clust}$;
end if
return $\mathcal{M}_q$;

3.4 Algorithm 4

Algorithm 4 combines **Algorithm** 2 with a clustering method at each step k of the algorithm.
General loop:
At each step k we compute $\mathcal{M}_{s_k}$ by **Algorithm** 2. Then, with the same groups composition we estimate $\mathcal{M}_{s_k}^{prod}$ to which we apply a classification step to produce $\mathcal{M}_{k,clust}$. If the BIC of model $\mathcal{M}_{k,clust}$ is smaller that the one of $\mathcal{M}_{s_k}$ we set down $\mathcal{M}_{s_k} = \mathcal{M}_{k,clust}$. We replace the step 16 in **Algorithm** 2 by:

Algorithm 4 Find $\mathcal{M}_q$
Step 16bis:

$\mathcal{M}_{s_k} \leftarrow \mathcal{M}_{s_{k_\ell}}^{\ell_{opt}}$;
We obtain $\mathcal{M}_{k,clust}$ from $\mathcal{M}_{s_k}$ with a clustering method;
if $BIC(\theta, \mathcal{M}_{k,clust}) < BIC(\theta, \mathcal{M}_{s_k})$ **then**
  $vect\_BIC(\ell_{opt}) \leftarrow BIC(\theta, \mathcal{M}_{k,clust})$;
  $\mathcal{M}_{s_k} \leftarrow \mathcal{M}_{k,clust}$;
end if

3.5 Conclusion and summary

The Table 1 summarizes the general characteristics of the algorithms.

4 Application

4.1 Analytical examples

In this section we compare the algorithms and we observe the behavior of the best one according to the number of observations and to the nugget effect.

4.1.1 Algorithms' Comparison

We simulate a Gaussian process with $p = 8$ parameters, the simulated model is :

$$Y(\mathbf{x}) = 1 + \epsilon(\mathbf{x})$$

where ϵ is a Gaussian process with a standard deviation $\sigma^2 = 1$ and an *isotropic by group kernel*:

$$r_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{x}') = \prod_{\ell=1}^{3} \rho_{\theta_\ell} \left(\|x - x'\|_{\mathcal{I}_\ell} \right)$$

where $\mathcal{I}_\ell = \{1\}$, $\mathcal{I}_2 = \{2, 3\}$, $\mathcal{I}_3 = \{4, 5, 6, 7, 8\}$ and ρ_θ is a Matern5_2. $\boldsymbol{\theta} = (0.5, 0.9, 0.8)$. The learning set is an optimized Latin Hypercube with n observations and the test set is a Sobol sequences of 1000 points.

We simulate 100 different trajectories of the model with $n = 800$ points of observations. For each trajectory we estimate the model with the algorithms. Table 2 shows that each algorithm finds the correct number and composition of groups for the 100 trajectories except **Algorithm** W. This result is not surprising because **Algorithm** W separates variable by variable and cannot propose several groups with more than one variable. Among the four algorithms, Table

	good groups	too separate
alg1	100	0
alg2	100	0
alg3	100	0
alg4	100	0
algW	0	100

Table 2: Gathering errors done by the algorithms for 100 trajectories with one learning set of size 800.

3 shows that **Algorithm** 4 is the fastest, it takes 1h21 to find the best model. In the next sections we only use **Algorithm** 4.

alg	Calculation time by processor
alg1	6h38
alg2	1h52
alg3	1h50
alg4	1h21
algW	1h20

Table 3: Calculation time done by the four algorithms for 100 trajectories with one learning set of size 800.

4.1.2 Evolution of the prediction quality with the learning set size

We test 7 learning set sizes : $n = \{80, 280, 480, 680, 880, 1080, 1280, \}$. For each size we simulate 1 learning set and 25 trajectories. For a learning set of size 80, the algorithm makes lot of bad

Sample size	good groups	bad gathering	too separate
80	7	17	1
280	23	2	0
480	25	0	0
680	25	0	0
880	25	0	0
1080	25	0	0
1280	25	0	0

Table 4: Gathering errors done by the **Algorithm** 4 for 25 trajectories with 9 learning set of size $\{80, 280, 480, 680, 880, 1080, 1280\}$

gathering (see Table 4). These errors are serious because they cause a loss of prediction quality. Figure 1 shows that the estimation time grows with the number of experiments. For a size of

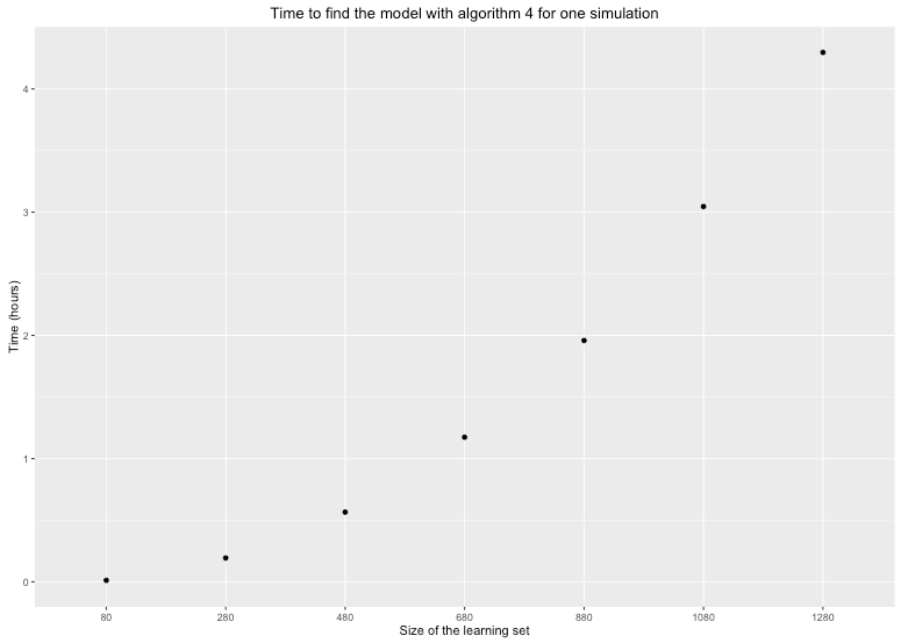

Figure 1: Time in hours to find the best model with the **Algorithm** 4. 25 trajectories are simulated for the each size $\{80, 280, 480, 680, 880, 1080, 1280\}$.

1280 the estimation takes more than 4 hours (see Table 3). The prediction quality also increases with the number of experiments. From $n = 680$, the prediction quality becomes very good (see Figure 2).

Whatever the size of the learning set kriging with an *Isotropic by group* kernel stays the best (see Figure 3).

4.1.3 Evolution of the prediction quality with an error on simulation parameters

We simulate a Gaussian process with $p = 8$ parameters, the simulated model is :

$$Y(\mathbf{x}) = 1 + \epsilon(\mathbf{x})$$

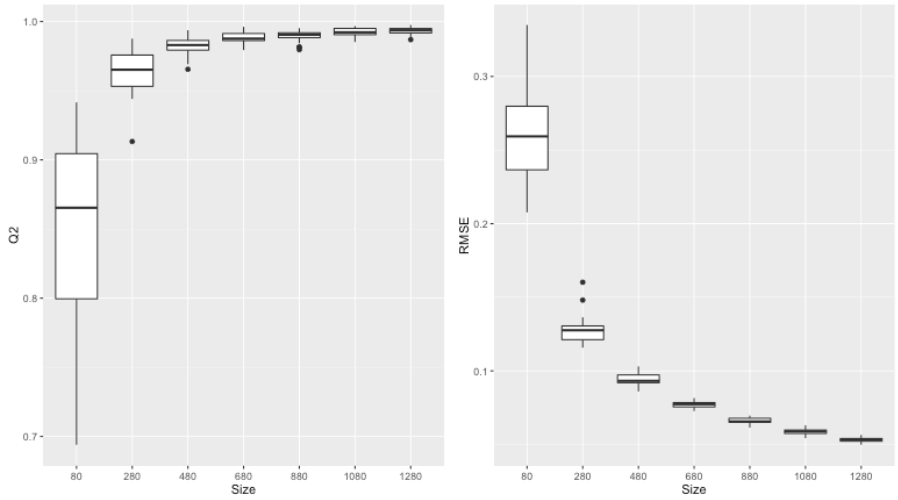

Figure 2: Prediction quality of the model with an *isotropic by group* kernel found by **Algorithm** 4 according to the sample size. 25 trajectories are simulated for each size $\{80, 280, 480, 680, 880, 1080, 1280\}$. The criteria of prediction quality are Q2 and RMSE.

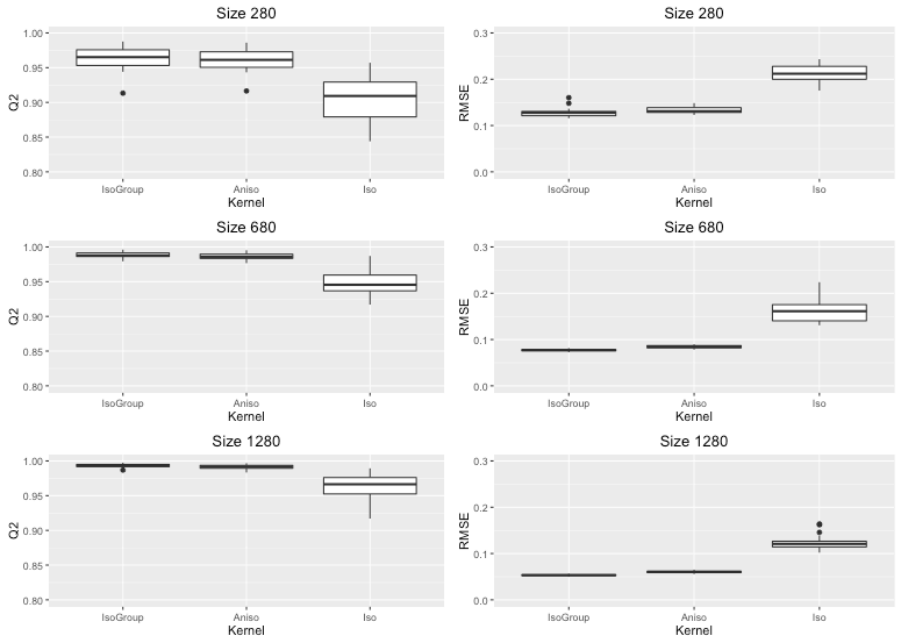

Figure 3: Prediction quality of the model with an *isotropic by group* kernel found by **Algorithm** 4 on the left, *anisotropic* kernel in the middle and *isotropic* kernel on the right according to the sample size. 25 trajectories are simulated for each learning set size $\{280, 680, 1280\}$. The criteria of prediction quality are Q2 and RMSE.

where ϵ is a Gaussian process with standard deviation $\sigma^2 = 1$ and an *anisotropic* kernel:

$$r_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{x}') = \prod_{j=1}^{p} \rho_{\theta_j^*}\left(|x_j - x_j'|\right)$$

where ρ_θ is a Matern5_2 and $\boldsymbol{\theta} = (0.5, 0.9, 0.9, 0.8, 0.8, 0.8)$. We simulate the range parameters with an error $\xi = (0, 2, 5, 10, 15, 40)$ such that $\theta_j^* \sim \mathcal{U}\left(\left[\theta_j - \frac{\xi_k}{100}\theta_j; \theta_j - \frac{\xi_k}{100}\theta_j\right]\right)$, $k = 1, \ldots, 6$ and $j = 1, \ldots, p$. The learning set is an optimized Latin Hypercube with $n = 400$ observations and the test set is a Sobol sequence of 1000 points. For each value of ξ we simulate 25 trajectories.

Adding an error to range parameters doesn't influence the prediction quality of $\mathcal{M}_q$ and it

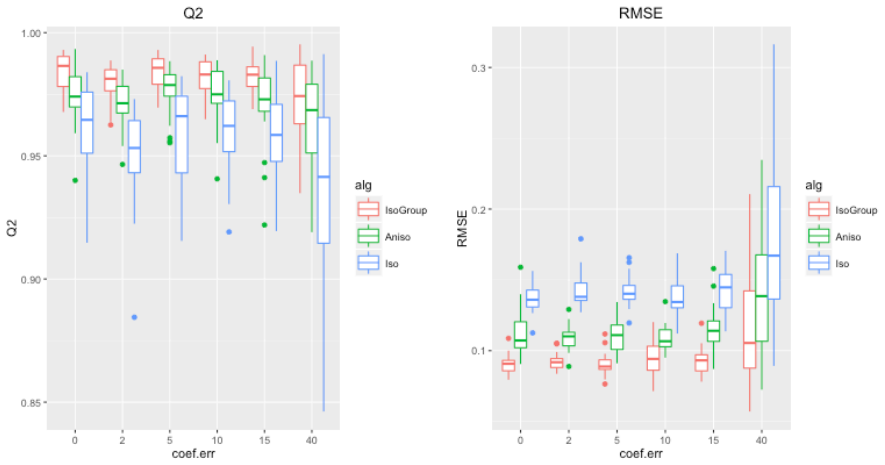

Figure 4: Prediction quality of the model with an *isotropic by group* kernel found by **Algorithm** 4 in orange *anisotropic* kernel in green and *isotropic* kernel in blue according to an error added on the range parameters in the simulated model. 25 trajectories are simulated for a learning set of size 400. The prediction quality criteria are Q2 and RMSE.

stays the best compared to $\mathcal{M}_a$ and $\mathcal{M}_i$, see Figure 4.

4.1.4 Evolution of the prediction quality with a nugget effect

We take the same Gaussian process as previously but with a nugget effect:

$$r_{\boldsymbol{\theta}}(\mathbf{x}-\mathbf{x}') = \prod_{j=1}^{p} \rho_{\theta_j}\left(|x_j - x'_j|\right) + \tau^2 \delta_0(\mathbf{x}-\mathbf{x}')$$

where ρ_θ is a Matern5_2. $\boldsymbol{\theta} = (0.5, 0.9, 0.9, 0.8, 0.8, 0.8)$. The learning set is an optimized Latin Hypercube with n observations and the test set is a Sobol sequence of 1000 points.

$$\delta_0(\mathbf{x}-\mathbf{x}') = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}' \\ 0 & \text{ifnot} \end{cases}$$

We test 6 different values of nugget effect $\tau = (0, 0.02, 0.03, 0.05, 0.1, 0.3)$.

Figure 5 shows a loss of prediction quality for each model. *Isotropic by group* kernel is not a good solution in the presence of a nugget effect. It stays comparable to the other models (*anisotropic* and *isotropic*) estimated a without nugget effect. In this context, whatever the kernel, allowing the estimation of a nugget effect is the correct solution.

4.2 Test function

To motivate the construction of our kriging model, let us consider the function::

$$f(\mathbf{x}) = \prod_{j=1}^{15} \frac{4x^j + a_i}{1 + a_i}$$

with $a^j = (0, 1, 1, 4.5, 10, 10, 99, 99, 99, 99, 99, 99, 99, 99, 99)$. $y_1, ..., y_n$ are the observations such that $y_i = f(\mathbf{x}_i)$ where f is the function and $\mathbf{x}_i \in [0; 1]^{15}$. Considering $750 = 50 \times 15$ runs of the function, we aim to construct a predictive model based on kriging, with three different kernels: *isotropic* ($\mathcal{M}_i$), *anisotropic* ($\mathcal{M}_a$) and *isotropic by group* ($\mathcal{M}_q$ is found automatically by the **Algorithm** 4) based on tree different correlation functions:

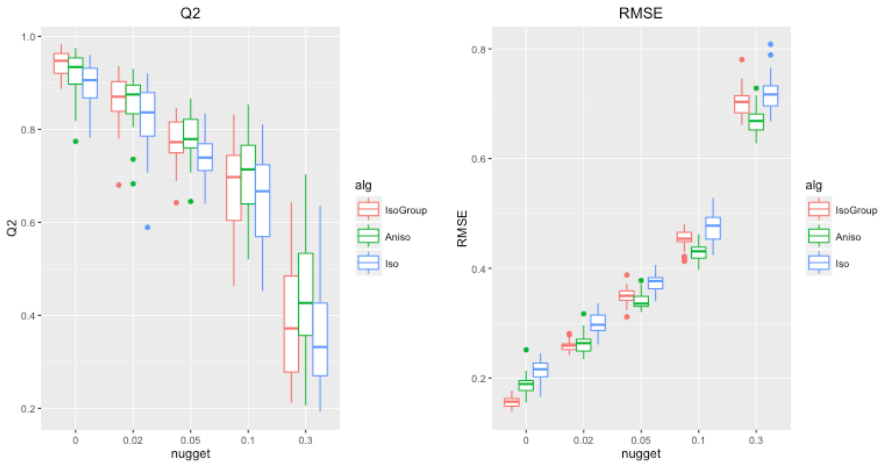

Figure 5: Prediction quality of the model with an *isotropic by group* kernel found by **Algorithm** 4 in orange *anisotropic* kernel in green and *isotropic* kernel in blue according to the value of a nugget effect added in the simulated model. 25 trajectories are simulated for a learning set of size {160}. The prediction quality criteria are Q2 and RMSE.

- *Anisotropic* ($\mathcal{M}_a$):

$$\mathbf{r}_{\boldsymbol{\theta}}(\mathbf{x}-\mathbf{x}') = \prod_{j=1}^{15} \rho_{\theta_j}\left(|x_j - x'_j|\right)$$

- *Isotropic* ($\mathcal{M}_i$):

$$\mathbf{r}_{\boldsymbol{\theta}}(\mathbf{x}-\mathbf{x}') = \rho_{\theta}\left(\|\mathbf{x}-\mathbf{x}'\|_2\right)$$

- *Isotropic by group* ($\mathcal{M}_4$):

$$\mathbf{r}_{\boldsymbol{\theta}}(\mathbf{x}-\mathbf{x}') = \rho_{\theta_1}\left(\sqrt{\sum_{j=1}^{3}(x_j-x'_j)^2}\right) \times \rho_{\theta_2}\left(|x_4-x'_4|\right) \times \rho_{\theta_3}\left(\sqrt{\sum_{j=5}^{6}(x_j-x'_j)^2}\right) \times \rho_{\theta_4}\left(\sqrt{\sum_{j=7}^{15}(x_j-x'_j)^2}\right)$$

The parameters m, σ^2 and $\boldsymbol{\theta}$ of different models have to be estimated from the sample. ρ_θ is the Matern 5_2 correlation function. In Figure 6, we compare the predictive power of three models : kriging with an *isotropic* kernel is poor, kriging with an *isotropic by group* kernel improves prediction power compared to kriging with an *anisotropic* kernel. Model $\mathcal{M}_4$ estimates 4 range parameters instead of 15 for model $\mathcal{M}_a$ and 1 for model $\mathcal{M}_i$. The estimates of $\boldsymbol{\theta}$, m and σ^2 are in Table 5. Table 5 shows that the range parameter groups correspond to close values of a in the Sobol function. Thus, the *isotropic by group* kernel allows a rise of the number of inputs.

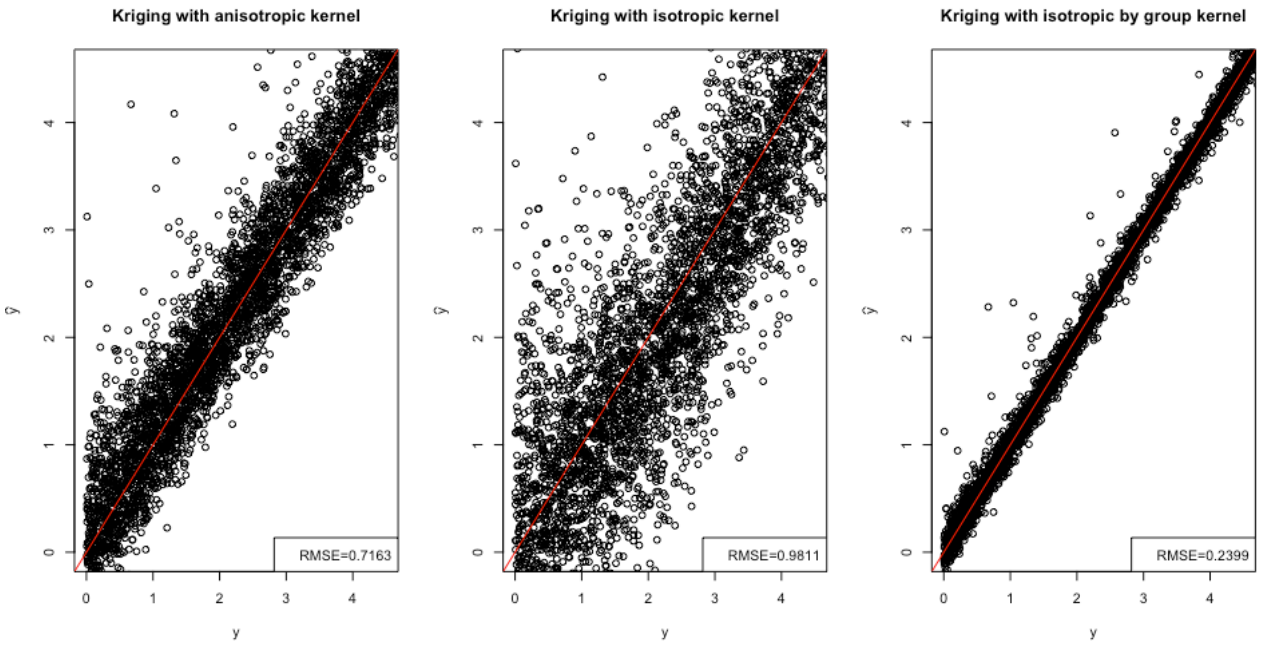

Figure 6: Kriging prediction plots for the function : *anisotropic* kernel (left), *isotropic* kernel (middle), *isotropic by group kernel* (right).

Anisotropic kernel

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
0.646	0.880	0.850	1.47	2.00	1.99	1.99	2.00	1.991
θ_{10}	θ_{11}	θ_{12}	θ_{13}	θ_{14}	θ_{15}	m	σ^2	
1.99	2.00	1.99	1.99	1.99	1.99	7.11	10.1	

***Isotropic* kernel**

θ	m	σ^2
2.88	7.11	268

***Isotropic by group* kernel**

$\theta_1(x_1, x_2, x_3)$	$\theta_2(x_4)$	$\theta_3(x_5, x_6)$	$\theta_4(x_7, \ldots, x_{15})$	m	σ^2
1.23	1.91	3.227334	15.0	31.0	6.40

Table 5: Kriging parameters for an *anisotropic* kernel (top), an *isotropic* kernel (middle) and an *isotropic by group* kernel (bottom). The learning set is an optimized Latin Hypercube of size 250. The test set is a Sobol sequence of 8000 points.

5 Conclusion

In high dimension, kriging model with classical kernels provide poor predictions of the response. Yet, with a sparse kernel, predictions are much better. After a study of the algorithms' behavior on their prediction quality and their time of estimation. It results that **Algorithm** 4 is the most efficient. The estimation time does not grow too fast with the number of experiments and the quality of prediction is always the best. The prediction quality increases with the number of experiments. Adding an error on the simulate range parameters influences the groups composition but not the prediction quality of the *isotropic by group* model. On the other hand adding a nugget effect to the simulation model degrade the quality of the *isotropic by group* kernel predictor but it stays close to the quality of the *anisotropic*. The comparison of the predictive power on the test function shows that kriging with the *isotropic* kernel is poor. Kriging with an *isotropic by group* kernel improves the predictive power compared to an *anisotropic* kernel. To conclude, the proposed methods enable to improve the prediction quality in the context of time-consuming simulation in high dimension.

In this paper, the algorithms are used for an *isotropic by group* kernel but these methods are generic and could be used for other kernels. Our current work consists in adapting **Algorithm** 4 to group *product* kernels, *additive* kernels presented in Muehlenstaedt et al. (2012) and *anova* kernels described in Durrande et al. (2013). The clustering method used to group the variables is hierarchical and only focus on the value of the range parameters. An idea is to find a specific method in the case of range parameters and that takes into account the number of inputs in the groups. The estimation of the range parameters needs an optimization of the log likelihood function. At each step of the algorithm, the chosen parameter set for initialization of the optimization is the point that maximizes the log likelihood function among a space filling design. The idea is to initialize the optimization with the value of range parameters estimated at the previous step. In the paper of Welch et al. (1992), they propose a modification in their algorithm to accelerate the optimization. They optimize on only one range parameter. This modification could be apply in our algorithms.

6 Acknowledgments

This work benefited from the financial support of the ANR project "PEPITO" (ANR-14-CE23-0011)

Bibliography

Antoniadis, A., Helbert, C., Prieur, C., and Viry, L. (2012). Spatio-temporal metamodeling for West African monsoon. Environmetrics, 23(1):24–36.

Binois, M., Ginsbourger, D., and Roustant, O. (2015). Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations. European J. Oper. Res., 243(2):386–394.

Booker, A. J., Dennis, Jr., J. E., Frank, P. D., Serafini, D. B., and Torczon, V. (1998). Optimization using surrogate objectives on a helicopter test example. In Computational methods for optimal design and control (Arlington, VA, 1997), volume 24 of Progr. Systems Control Theory, pages 49–58. Birkhäuser Boston, Boston, MA.

Cornford, D., Nabney, I. T., and Williams, C. K. I. (2002). Modelling frontal discontinuities in wind fields. J. Nonparametr. Stat., 14(1-2):43–58. Statistical models and methods for discontinuous phenomena (Oslo, 1998).

Cressie, N. A. C. (1993). Statistics for spatial data. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., New York. Revised reprint of the 1991 edition, A Wiley-Interscience Publication.

Durrande, N. (2001). Étude de classes de noyaux adaptées à la simplification et à l'interprétation des modèles d'approximation. Une approche fonctionnelle et probabiliste. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne.

Durrande, N., Ginsbourger, D., and Roustant, O. (2012). Additive covariance kernels for high-dimensional Gaussian process modeling. Ann. Fac. Sci. Toulouse Math. (6), 21(3):481–499.

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

Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). Cluster analysis. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., Chichester, fifth edition.

Fricker, T. E., Oakley, J. E., and Urban, N. M. (2013). Multivariate Gaussian process emulators with nonseparable covariance structures. Technometrics, 55(1):47–56.

Gao, J., Gunn, S., and Kandola, J. (2002). Adapting kernels by variational approach in SVM. In AI 2002: Advances in artificial intelligence, volume 2557 of Lecture Notes in Comput. Sci., pages 395–406. Springer, Berlin.

Ginsbourger, D., Roustant, O., Schuhmacher, D., Durrande, N., and Lenz, N. (2016). On ANOVA decompositions of kernels and Gaussian random field paths. In Monte Carlo and quasi-Monte Carlo methods, volume 163 of Springer Proc. Math. Stat., pages 315–330. Springer, [Cham].

Marrel, A., Iooss, B., Van Dorpe, F., and Volkova, E. (2008). An efficient methodology for modeling complex computer codes with Gaussian processes. Comput. Statist. Data Anal., 52(10):4731–4744.

Muehlenstaedt, T., Roustant, O., Carraro, L., and Kuhnt, S. (2012). Data-driven Kriging models based on FANOVA-decomposition. Stat. Comput., 22(3):723–738.

Paciorek, C. J. and Schervish, M. J. (2006). Spatial modelling using a new class of nonstationary covariance functions. Environmetrics, 17(5):483–506.

Padonou, E. and Roustant, O. (2016). Polar Gaussian processes and experimental designs in circular domains. SIAM/ASA J. Uncertain. Quantif., 4(1):1014–1033.

Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian processes for machine learning. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.

Roustant, O., Ginsbourger, D., and Deville, Y. (2012). Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). The design and analysis of computer experiments. Springer Series in Statistics. Springer-Verlag, New York.

Schwarz, G. (1978). Estimating the dimension of a model. Ann. Statist., 6(2):461–464.

Snelson, E. L. (2008). Flexible and efficient Gaussian process models for machine learning. ProQuest LLC, Ann Arbor, MI. Thesis (Ph.D.)–University of London, University College London (United Kingdom).

Stein, M. L. (1999). Interpolation of spatial data. Springer Series in Statistics. Springer-Verlag, New York. Some theory for Kriging.

Stitson, M., Gammerman, A., Vapnik, V., Vovk, V., Watkins, C., and Weston, J. (1997). Support vector regression with anova decomposition kernels. Advances in kernel methodsSupport vector learning, pages 285–292.

Sudret, B. (2012). Meta-models for structural reliability and uncertainty quantification. arXiv preprint arXiv:1203.2062.

Villa-Vialaneix, N., Follador, M., Ratto, M., and Leip, A. (2012). A comparison of eight metamodeling techniques for the simulation of n 2 o fluxes and n leaching from corn crops. Environmental Modelling & Software, 34:51–66.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992). Screening, predicting, and computer experiments. Technometrics, 34(1):15–25.

Yi, G. (2009). Variable selection with penalized Gaussian process regression models. PhD thesis, University of Newcastle Upon Tyne.

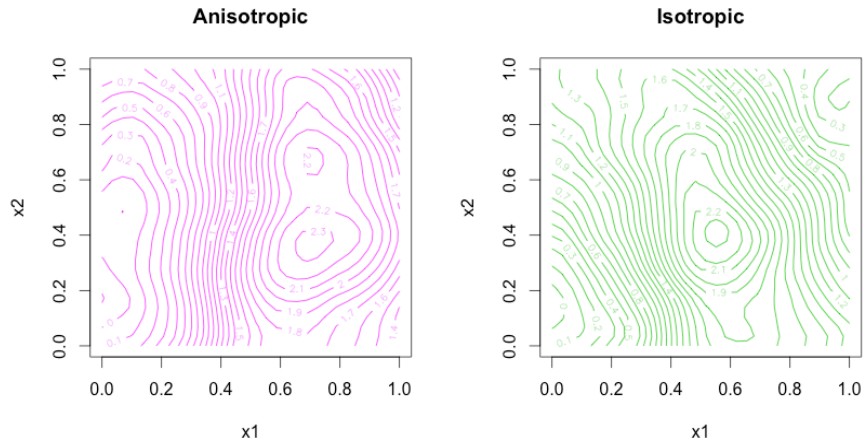

Figure 7: Simulation of a Gaussian process trajectory with 2 different kernels. *Anisotropic* on the left and *isotropic* on the right.

A Visualization of *isotropic* and *anisotropic* kernels

We simulate in the Figure 7 a Gaussian process with $p = 2$ parameters, the simulated model is :

$$Y(\mathbf{x}) = 1 + \epsilon(\mathbf{x})$$

where ϵ is a Gaussian process with a standard deviation $\sigma^2 = 1$.
With an *anisotropic* kernel:

$$r_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{x}') = \rho_\theta\left(|x_1 - x_1'|\right) \times \rho_\theta\left(|x_2 - x_2'|\right)$$

and an *isotropic* kernel :

$$r_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{x}') = \rho_\theta\left(\|\mathbf{x} - \mathbf{x}'\|_2\right)$$

where ρ_θ is a Matern5_2 and $\theta = 0.5$.

B Algorithm W

This algorithm is inspired by Welch et al. (1992). At each step k, if the previous model has $s_{k-1} = k - 1$ groups, the next estimated models will have $s_k = k$ groups.
We note $\mathcal{M}_{s_k}$ the model at step k with $s_k = k$ groups. Let $\mathcal{I}_1 = \{i_{1,1}, \ldots, i_{1,p_1}\}; \mathcal{I}_2 = \{i_{2,1}\}; \ldots; \mathcal{I}_{s_k} = \{i_{s_k,1}\}$, with $p_1 + s_k - 1 = p$ where $(s_k - 1 = k)$.

Checking step:
We create an additional group in taking an index $i_{1,m} \in \mathcal{I}_1$, $m = 1, \ldots, p_1$ out of the group 1. We have $J_k = p_1$ possibilities and we obtain $s_k = s_{k-1} + 1$ groups. we create J_k models noted $\mathcal{M}_{s_k}^1, \ldots, \mathcal{M}_{s_k}^{J_k}$ and we follow the **General step**.
In the **Algorithm** 2 steps 5-9 are replace by :

Algorithm W Find $\mathcal{M}_q$
Step 5bis-9bis:

We construct $J_k = p_1$ models $\mathcal{M}_{s_k}^\ell$, $\ell = 1, \ldots, J_k$;
