# Nested Kriging estimations for datasets with large number of observations

Didier Rullière, Nicolas Durrande, François Bachoc, Clément Chevalier

# Nested Kriging estimations for datasets with large number of observations

Didier Rullière[*], Nicolas Durrande[†], François Bachoc[‡] and Clément Chevalier[§]

Wednesday 21st December, 2016

**Abstract**

This work falls within the context of predicting the value of a real function $f$ at some input locations given a limited number of observations of this function. Kriging interpolation technique (or Gaussian process regression) is often considered to tackle such problem but the method suffers from its computational burden when the number of observation points $n$ is large. We introduce in this article nested Kriging estimators which are constructed by aggregating sub-models based on subsets of observation points. This approach is proven to have better theoretical properties than other aggregation methods that can be found in the literature. In particular, contrary to some other methods which are shown inconsistent, we prove the consistency of our proposed aggregation method. Finally, the practical interest of the proposed method is illustrated on simulated datasets and on an industrial test case with $10^4$ observations in a 6-dimensional space.

## 1 Introduction

Gaussian process regression models have proven to be of great interest in many fields when it comes to predict the output of a function $f : D \to \mathbb{R}$, $D \subset \mathbb{R}^d$, based on the knowledge of $n$ input-output tuples $(x_i, f(x_i))$ for $1 \leq i \leq n$ [Stein, 2012, Santner et al., 2013, Williams and Rasmussen, 2006]. One asset of this method is to provide not only a mean predictor but also a quantification of the model uncertainty. The Gaussian process regression framework uses a (centered) real-valued Gaussian process $Y$ over $D$ as a prior distribution for $f$ and approximates it by the conditional distribution of $Y$ given the observations $Y(x_i) = f(x_i)$ for $1 \leq i \leq n$. In this framework, we denote by $k : D \times D \to \mathbb{R}$ the covariance function (or kernel) of $Y$: $k(x, x') = \mathrm{Cov}\left[Y(x), Y(x')\right]$, and by $X \in D^n$ the vector of observation points with entries $x_i$ for $1 \leq i \leq n$.

In the following, we use classical vectorial notations: for any functions $f : D \to \mathbb{R}$, $g : D \times D \to \mathbb{R}$ and for any vectors $A = (a_1, \ldots, a_n) \in D^n$ and $B = (b_1, \ldots, b_m) \in D^m$, we denote by $f(A)$ the $n \times 1$ real valued vector with components $f(a_i)$ and by $g(A, B)$ the $n \times m$ real valued matrix with components $g(a_i, b_j)$, $i = 1, \ldots, n$, $j = 1, \ldots, m$. With such notations, the conditional distribution of $Y$ given the $n \times 1$ vector of observations $Y(X)$ is Gaussian with mean, covariance and variance:

$$\begin{cases} M_{full}(x) = \mathrm{E}\left[Y(x)|Y(X)\right] = k(x, X)k(X, X)^{-1}Y(X)\,, \\ c_{full}(x, x') = \mathrm{Cov}\left[Y(x), Y(x')|Y(X)\right] = k(x, x') - k(x, X)k(X, X)^{-1}k(X, x')\,, \\ v_{full}(x) = c_{full}(x, x)\,. \end{cases} \tag{1}$$

Since we do not specify yet the values taken by $Y$ at $X$, the "mean predictor" $M_{full}(x)$ is random so it is denoted by an upper-case letter $M$. The approximation of $f(x)$ given the observations

[*]Université de Lyon, Université Claude Bernard Lyon 1, ISFA, Laboratoire SAF, EA2429, 50 avenue Tony Garnier, 69366 Lyon, France, didier.rulliere@univ-lyon1.fr.

[†]*Corresponding author*, Mines Saint-Étienne, H. Fayol Institute, 158 cours Fauriel, Saint-Étienne, France and CNRS LIMOS, UMR 5168, durrande@emse.fr.

[‡]Institut de Mathématiques de Toulouse, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse, France, francois.bachoc@math.univ-toulouse.fr.

[§]Institute of Statistics, University of Neuchâtel, avenue de Bellevaux 51, 2000 Neuchâtel, Switzerland, clement.chevalier@unine.ch.

$f(X)$ is thus given by $m_{full}(x) = \mathrm{E}\left[Y(x)|Y(X) = f(X)\right] = k(x, X)k(X, X)^{-1}f(X)$. This method is quite general since an appropriate choice of the kernel allows to recover the models obtained from various frameworks such as linear regression and splines models [Wahba, 1990].

One limitation of such models is the computational time required for building models based on a large number of observations. Indeed, these models require computing and inverting the $n \times n$ covariance matrix $k(X, X)$ between the observed values $Y(X)$, which leads to a $O(n^2)$ complexity in space and $O(n^3)$ in time. In practice, this computational burden makes Gaussian process regression difficult to use when the number of observation points is in the range $[10^3, 10^4]$ or greater.

Many methods have been proposed in the literature to overcome this limit. Let us first mention that, when the observations are recorded on a regular grid, choosing a separable covariance function $k$ enables to drastically simplify the inversion of the covariance matrix $k(X, X)$, since the latter can be written as a Kronecker product.

For irregularly spaced data, a common approach in machine learning relies on inducing points. It consists in introducing a set $W$ of pseudo input points and in approximating the full conditional distribution $Y(x)|Y(X)$ by $Y(x)|Y(W)$. The challenge here is to find the best locations for the inducing inputs and to decide which values should be assigned to the outputs at $W$. Various methods are suggested in the literature to answer these questions [Hensman et al., 2013]. One drawback of this kind of approximation is that the predictions do not interpolate the observation points any more. Note that this method has recently been combined with the Kronecker product method in [Nickson et al., 2015].

Among popular classes of methods dedicated to large datasets, one can also cite low rank approximations (see [Stein, 2014] and the references therein for a review). They typically assume that the covariance matrix $k(X, X)$ is a sum of a diagonal matrix and a low rank matrix. While these techniques can be very computationally efficient, they may poorly reproduce the underlying process small scale dependence [Stein, 2008, Stein, 2014]. See also the experiments conducted in [Datta et al., 2016].

Another possibility is to rely on compactly supported covariance functions. The latter can be done by tapering an initial arbitrary covariance function [Kaufman et al., 2008, Furrer et al., 2006], or by simply working directly with a compactly supported covariance function [Kaufman et al., 2011, Stein, 2013]. This approach benefits from the sparse covariance matrix $k(X, X)$, for which efficient factorization algorithms exist. Furthermore, it has been combined with low rank approximation in [Sang and Huang, 2012]. The main criticism against these methods is that, by definition, they tend not to capture large scale dependencies. [Maurya, 2016] argues however that the issue can be mitigated by the choice of appropriate trend functions; a case in which the independence assumption between the (zero-mean) residuals at distant locations becomes more realistic. Another drawback – which to the best of our knowledge is little discussed in the literature – is the difficulty to use these methods when the dimension of the input space is large (say larger than 10, which is frequent in computer experiments or machine learning). In this case, stationary compactly supported covariance functions are known to decrease quickly to zero (see e.g. [Gneiting, 2002] for more details about parametric families of such covariance functions) so that their effective range, i.e. the range at which the correlation equals 5%, is in fact much lower than the actual range. This issue leads to covariance matrices where non-zero off diagonal terms will be widely dominated by the diagonal ones, i.e. where the underlying process behaves like a white-noise. In application, these techniques have been applied in dimension up to 4 [Kaufman et al., 2011]. See also the remarks of [Xue et al., 2012, Bai et al., 2012, Maurya, 2016] about the performances of sparse Cholesky algorithms when the dimension of the input set becomes larger.

Other methods based on Gaussian Markov Random Fields (GMRF) assume that the precision matrix, i.e. the inverse of $k(X, X)$ is a sparse matrix [Rue and Held, 2005]. A drawback of these methods is the difficulty to compute predictions at arbitrary locations, since nodes should be placed at both observation and prediction locations (see also the recent work of [Datta et al., 2016]). There is also a relative lack of flexibility when the Markov assumption about the random field is unnatural.

Let us also mention that the computation of $k(X,X)^{-1}y$, for an arbitrary vector $y \in R^n$ can be performed using iterative algorithms, like the preconditionned conjugate gradient algorithm [Golub and Van Loan, 2012]. Unfortunately, the algorithms need to be run many times when a posterior variance – involving the computation of $k(X,X)^{-1}k(X,x_i)$ – needs to be computed for a large set of prediction points.

Finally, we mention inference methods based on composite likelihood, see [Vecchia, 1988] and [Bevilacqua and Gaetan, 2015] or on iterative algorithms solving the score equations [Stein et al., 2013, Anitescu et al., 2016] which enable to efficiently find maximum likelihood estimates for the covariance hyperparameters. While these techniques are very promising for inference with large dataset, they do not give a process interpretation which enables interpolation.

The method proposed in this paper belongs to the a so-called "mixture of experts" family. The latter relies on the aggregation of sub-models based on subsets of the data which make them easy to compute. This kind of methods offers a great flexibility since it can be applied with any covariance function and in large dimension while retaining the interpolation property. Some existing "mixture of experts" methods are product of experts [Hinton, 2002], and the (robust) Bayesian committee machine [Tresp, 2000, Deisenroth and Ng, 2015]. All these methods are based on a similar approach: for a given point $x$, each sub-model provides its own prediction (a mean and a variance) and these predictions are then merged into one single mean and prediction variance. The differences between these methods lie in how to aggregate the predictions made by each sub-model. It shall be noted that aggregating expert opinions is the topic of *consensus statistical methods* (sometimes referred to as *opinion synthesis* or *averaging methods*), where probability distributions representing expert opinions are joined together. Early references are [Winkler, 1968, Winkler, 1981]. A detailed review and an annotated bibliography is given in [Genest and Zidek, 1986] (see also [Satopää et al., 2015, Ranjan and Gneiting, 2010] for recent related developments). From a probabilistic perspective, usual mixture of experts methods assume that there is some (conditional) independence between the sub-models. Although this kind of hypothesis leads to efficient computations, it is often violated in practice and may lead to poor predictions as illustrated in [Samo and Roberts, 2016]. Furthermore, these methods only provide pointwise confidence intervals instead of a full Gaussian process posterior distribution.

The new aggregation method we develop in this article is part of the mixture of experts framework, so it will benefit from the properties of this family: it does not require the data to be on a grid, the predictions can interpolate the observations and it can be applied to data with small or large scale dependencies regardless of the input space dimension. Compared to other mixtures of experts, we relax the usually made independence assumption so that the prediction takes into account all $n^2$ pairwise cross-covariances between observations. We show this addresses two main pitfalls of usual mixture of experts: the predictions are more accurate and the theoretical consistency is ensured (we prove it is not the case for product of experts and Bayesian committee machine). Furthermore, the proposed method remains computationally affordable: predictions are performed in a few seconds for $n = 10^4$ and a few minutes for $n = 10^5$ using standard laptop and the proposed online implementation. Finally, the prediction method comes with a naturally associated inference procedure, which is based on cross validation errors.

The proposed method is presented in Section 2. In particular, we detail a pointwise approach and a process based approach such that the proposed aggregation method can be seen as an optimal method for a modified prior process. In Section 3, we introduce an iterative scheme for nesting the estimators derived previously. A procedure for estimating the parameters of models is then given in Section 4. Finally, Section 5 compares the method with state of the art aggregation methods on both a simulated dataset and an industrial case study.

# 2 Proposed aggregation

## 2.1 Pointwise aggregation of experts

Let us now address in more details the framework of this article. The method is based on the aggregation of sub-models defined on smaller subsets of points. Let $X_1, \ldots, X_p$ be subvectors of the vector of observations input points $X$, it is thus possible to define $p$ associated sub-models (or

*experts*) $M_1, \ldots, M_p$. For example, the sub-model $M_i$ can be a Gaussian process regression model based on a subset of the data

$$M_i(x) = \mathrm{E}\left[Y(x)|Y(X_i)\right] = k(x, X_i)k(X_i, X_i)^{-1}Y(X_i)\,, \tag{2}$$

however, we make no Gaussian assumption in this section. For a given prediction point $x \in D$, the $p$ sub-models predictions are gathered into a $p \times 1$ vector $M(x) = (M_1(x) \ldots, M_p(x))^t$. The random column vector $(M_1(x), \ldots, M_p(x), Y(x))^t$ is supposed to be centered with finite first two moments and we consider that both the $p \times 1$ covariance vector $k_M(x) = \mathrm{Cov}\left[M(x), Y(x)\right]$ and the $p \times p$ covariance matrix $K_M(x) = \mathrm{Cov}\left[M(x), M(x)\right]$ are given. Sub-models aggregation (or mixture of experts) aims at merging all the pointwise sub-models $M_1(x), \ldots, M_p(x)$ into one unique pointwise estimator $M_{\mathcal{A}}(x)$ of $Y(x)$. We propose the following aggregation:

**Definition 1** (Sub-models aggregation)**.** *For a given point $x \in D$, let $M_i(x)$, $i \in \mathcal{A} = \{1, \ldots, p\}$ be sub-models with covariance matrix $K_M(x)$. Then, when $K_M(x)$ is invertible, we define the sub-model aggregation as:*

$$M_{\mathcal{A}}(x) = k_M(x)^t K_M(x)^{-1} M(x). \tag{3}$$

In practice, the invertibility condition on $K_M(x)$ can be avoided by using matrices pseudo-inverses. Given the vector of observations $M(x) = m(x)$, the associated prediction is

$$m_{\mathcal{A}}(x) = k_M(x)^t K_M(x)^{-1} m(x). \tag{4}$$

Notice that we are here aggregating random variables rather than their distributions. For dependent non-elliptical random variables, expressing the probability density function of $M_{\mathcal{A}}(x)$ as a function of each expert density $M_i(x)$ is not straightforward. This difference in the approaches implies that the proposed method differs from usual consensus aggregations. For example, aggregating random variables allows to specify the correlations between the aggregated prediction and the experts whereas aggregating expert distributions into a univariate prediction distribution does not characterize uniquely these correlations.

**Proposition 1** (BLUE)**.** *$M_{\mathcal{A}}(x)$ is the best linear unbiased estimator (BLUE) of $Y(x)$ that writes $\sum_{i\in\mathcal{A}} \alpha_i(x) M_i(x)$. The mean squared error $v_{\mathcal{A}}(x) = \mathrm{E}\left[(Y(x) - M_{\mathcal{A}}(x))^2\right]$ writes*

$$v_{\mathcal{A}}(x) = k(x, x) - k_M(x)^t K_M(x)^{-1} k_M(x)\,. \tag{5}$$

*The coefficients $\{\alpha_i(x), i \in \mathcal{A}\}$ are given by the vector $\alpha = k_M(x)^t K_M(x)^{-1}$.*

*Proof.* The standard proof applies: The square error writes $\mathrm{E}\left[(Y(x) - \alpha^t M(x))^2\right] = k(x, x) - 2\alpha^t k_M(x) + \alpha^t K_M(x)\alpha$. The value of $\alpha^*$ minimising it can be found by differentiation: $-2k_M(x) + 2\alpha^* K_M(x) = 0$ which leads to $\alpha^* = K_M(x)^{-1} k_M(x)$. Then, $v_{\mathcal{A}}(x) = k(x, x) - 2\alpha^{*t} k_M(x) + \alpha^{*t} K_M(x)\alpha^*$ and the result holds. □

**Proposition 2** (Basic properties)**.** *Let $x$ be a given prediction point in $D$.*

*(i) Linear case: if $M(x)$ in linear in $Y(X)$, i.e. if there exists a $p \times n$ deterministic matrix $\Lambda(x)$ such that $M(x) = \Lambda(x)Y(X)$ and if $\Lambda(x)k(X, X)\Lambda(x)^t$ is invertible, then $M_{\mathcal{A}}(x)$ in linear in $Y(X)$ with*

$$\left\{ \begin{array}{rcl} M_{\mathcal{A}}(x) & = & \lambda_{\mathcal{A}}(x)^t Y(X)\,, \\ v_{\mathcal{A}}(x) & = & k(x, x) - \lambda_{\mathcal{A}}(x)^t k(X, x)\,. \end{array} \right. \tag{6}$$

*where $\lambda_{\mathcal{A}}(x)^t = k(x, X)\Lambda(x)^t\big(\Lambda(x)k(X, X)\Lambda(x)^t\big)^{-1}\Lambda(x)$.*

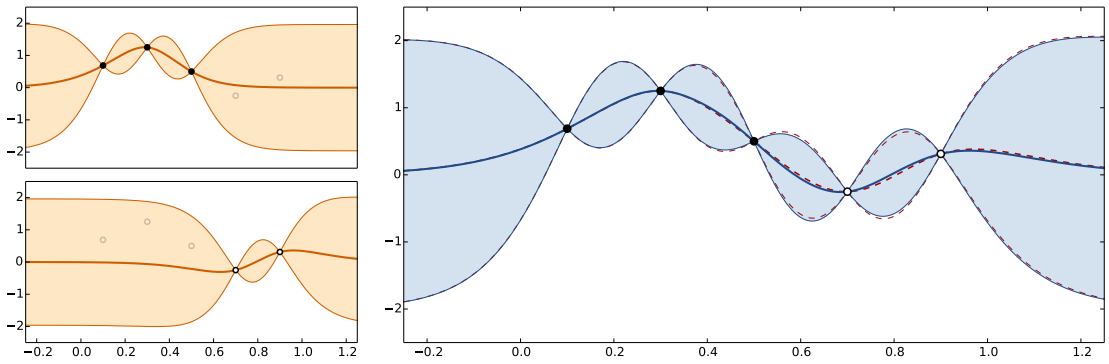
*(ii) Interpolation case: if $M$ interpolates $Y$ at $X$, i.e. if for any component $x_k$ of the vector $X$ there is at least one index $i_k \in \mathcal{A}$ such that $M_{i_k}(x_k) = Y(x_k)$, and if $K_M(x_k)$ is invertible for any component $x_k$ of $X$, then $M_{\mathcal{A}}$ is also interpolating, i.e.*

$$\left\{ \begin{array}{rcl} M_{\mathcal{A}}(X) & = & Y(X)\,, \\ v_{\mathcal{A}}(X) & = & 0_n\,, \end{array} \right. \tag{7}$$

*where $0_n$ is a $n \times 1$ vector with entries 0. This property can be extended when some $K_M(x_k)$ are not invertible by using pseudo-inverse in place of matrix inverse in Definition 1.*

*(iii) Gaussian case: if the joint distribution $(M(x), Y(x))$ is multivariate normal, then the conditional distribution of $Y(x)$ given $M(x)$ is normal with moments*

$$\left\{ \begin{array}{l} \mathrm{E}\left[Y(x)|M_i(x), i \in \mathcal{A}\right] = M_{\mathcal{A}}(x)\,, \\ \mathrm{V}\left[Y(x)|M_i(x), i \in \mathcal{A}\right] = v_{\mathcal{A}}(x)\,. \end{array} \right. \tag{8}$$



(a) sub-models to aggregate (b) aggregated model (solid lines) and full model (dashed lines)

Figure 1: Example of aggregation of two Gaussian process regression models. For each model, we represent the predicted mean and 95% confidence intervals.

*Proof.* Linearity directly derives from $k_M(x) = \Lambda(x)k(X,x)$ and $K_M(x) = \Lambda(x)K(X,X)\Lambda(x)^t$.
Interpolation: Let $k \in \{1,\dots,n\}$, and $i \in \mathcal{A}$ be an index such that $M_i(x_k) = Y(x_k)$. As $K_M(x_k) = \mathrm{Cov}\left[M(x_k), M(x_k)\right]$, the $i^{th}$ line of $K_M(x_k)$ is equal to $\mathrm{Cov}\left[M_i(x_k), M(x_k)\right] = \mathrm{Cov}\left[Y(x_k), M(x_k)\right] = k_M(x_k)^t$. Setting $e_i$ the $p$ dimensional vector having entries 0 except on its $i^{th}$ component, it is thus clear that ${e_i}^t K_M(x_k) = k_M(x_k)^t$. As $K_M(x_k)$ is assumed to be invertible, then ${e_i}^t = k_M(x_k)^t K_M(x_k)^{-1}$, so that $M_{\mathcal{A}}(x_k) = k_M(x_k)^t K_M(x_k)^{-1} M(x_k) = {e_i}^t M(x_k) = M_i(x_k) = Y(x_k)$. This result can be plugged into the definition of $v_{\mathcal{A}}$ to obtain the second part of Eq. (7): $v_{\mathcal{A}}(x_k) = \mathrm{E}\left[(Y(x_k) - M_{\mathcal{A}}(x_k))^2\right] = 0$.
Finally the Gaussian case can be proved directly by applying the usual multivariate normal conditioning formula. □

**Example 1** (Gaussian process regression aggregation)**.** *In this example, we set $D = \mathbb{R}$ and we approximate the function $f(x) = \sin(2\pi x) + x$ based on a set of five observation points in $D$: $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. These observations are gathered in two column vectors $X_1 = (0.1, 0.3, 0.5)$ and $X_2 = (0.7, 0.9)$. We use as prior a centered Gaussian process $Y$ with squared exponential covariance $k(x,x') = \exp\left(-12.5(x-x')^2\right)$ in order to build two Kriging sub-models, for $i \in \{1,2\}$:*

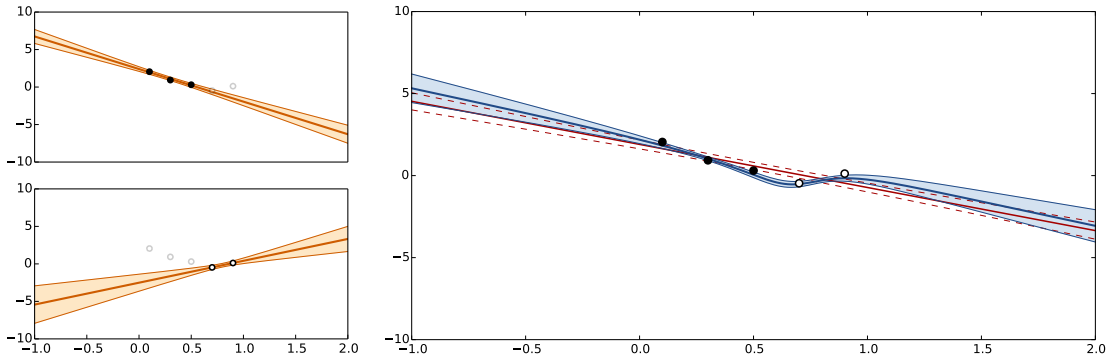
$$\begin{cases} M_i(x) = \mathrm{E}\left[Y(x)|Y(X_i)\right] = k(x,X_i)k(X_i,X_i)^{-1}Y(X_i)\,, \\ m_i(x) = \mathrm{E}\left[Y(x)|Y(X_i) = f(X_i)\right] = k(x,X_i)k(X_i,X_i)^{-1}f(X_i)\,. \end{cases} \tag{9}$$

*The expressions required to compute $M_{\mathcal{A}}$ as defined in Eq. (3) are for $i,j \in \{1,2\}$:*

$$\begin{cases} \left(k_M(x)\right)_i = \mathrm{Cov}\left[M_i(x), Y(x)\right] = k(x,X_i)k(X_i,X_i)^{-1}k(X_i,x)\,, \\ \left(K_M(x)\right)_{i,j} = \mathrm{Cov}\left[M_i(x), M_j(x)\right] = k(x,X_i)k(X_i,X_i)^{-1}k(X_i,X_j)k(X_j,X_j)^{-1}k(X_j,x)\,. \end{cases} \tag{10}$$

*Recall $m_{full}(x) = \mathrm{E}\left[Y(x)|Y(X) = f(X)\right]$ and $v_{full}(x) = \mathrm{V}\left[Y(x)|Y(X) = f(X)\right]$, as it can de seen in Figure 1, the resulting model $m_{\mathcal{A}}$ appears to be a very good approximation of $m_{full}$ and there is only a slight difference between prediction variances $v_{\mathcal{A}}$ and $v_{full}$ on this example.*

**Example 2** (Linear regression aggregation)**.** *In this distribution-free example, we set $D = \mathbb{R}$ and we consider the process $Y(x) = \varepsilon_1 + \varepsilon_2 x$ where $\varepsilon_1$ and $\varepsilon_2$ are independent centered random variables with unit variance. $Y$ is thus centered with covariance $k(x,x') = 1 + xx'$. Furthermore, we consider that $Y$ is corrupted by some observation noise $Y_{obs}(x) = Y(x) + \varepsilon_3(x)$ where $\varepsilon_3(x)$ is an independent white noise process with covariance $k_3(x,x') = \mathbf{1}_{\{x=x'\}}$. Note that we only make assumptions on the first two moments of $\varepsilon_1$, $\varepsilon_2$ or $\varepsilon_3(x)$ but not on their laws. We introduce five observation points gathered in two column vectors: $X_1 = (0.1, 0.3, 0.5)^t$ and $X_2 = (0.7, 0.9)^t$ and their associated outputs $y_1 = (2.05, 0.93, 0.31)^t$ and $y_2 = (-0.47, 0.12)^t$. The linear regression sub-models, obtained by square error minimization, are $M_i(x) = k(x,X_i)(k(X_i,X_i) + Id)^{-1}Y_{obs}(X_i)$, $i \in \{1,2\}$. Resulting covariances $\mathrm{Cov}\left[M_i(x), Y(x)\right]$, $\mathrm{Cov}\left[M_i(x), M_j(x)\right]$ and aggregated model $M_{\mathcal{A}}(x)$, $v_{\mathcal{A}}(x)$ of Eq. (8) are then easily obtained. The resulting model is illustrated in Figure 2.*



(a) sub-models to aggregate (b) aggregated model (solid lines) and full model (dashed lines)

Figure 2: Example of aggregation of two linear regression sub-models. Exhibited confidence bands correspond to a difference to mean value of two standard deviations.

## 2.2 Random process perspective

In this section, we develop an alternative construction where the process $Y$ is replaced by an alternative prior $Y_{\mathcal{A}}$ for which $M_{\mathcal{A}}(x)$ and $v_{\mathcal{A}}(x)$ correspond exactly to the conditional expectation and variance of $Y_{\mathcal{A}}(x)$ given $Y_{\mathcal{A}}(X)$. As discussed in [Quinonero-Candela and Rasmussen, 2005], this point of view allows us to see the proposed aggregation not only as an approximation of the full model but also as an exact method for a slightly different prior (as illustrated in the further commented Figure 3). As a consequence, it also provides conditional cross-covariances (which were not available in the pointwise approach) and posterior samples can be associated to the aggregated models. Furthermore, all the methods developed in the literature based on Kriging predicted covariances, such as [Marrel et al., 2009] for sensitivity analysis and [Chevalier and Ginsbourger, 2013] for optimization, may be applied to our aggregated models.

Here, we consider that $(M_1, \dots, M_p, Y)^t$ is a centered process with finite variance on the whole input space $D$. We define the $p \times 1$ cross-covariance vector $k_M(x, x') = \mathrm{Cov}\left[M(x), Y(x')\right]$ and $p \times p$ cross-covariance matrix $K_M(x, x') = \mathrm{Cov}\left[M(x), M(x')\right]$, for all $x, x' \in D$. Notice that using notations of subsection 2.1, $K_M(x) = K_M(x, x)$ and $k_M(x) = k_M(x, x)$. We now define an aggregated process $Y_{\mathcal{A}}$ based on $M_{\mathcal{A}}$ that aims at reproducing the behaviour of the process $Y$:

**Definition 2** (Aggregated process)**.** *We define the process $Y_{\mathcal{A}}$ as $Y_{\mathcal{A}} = M_{\mathcal{A}} + \varepsilon'_{\mathcal{A}}$ where $\varepsilon'_{\mathcal{A}}$ is an independent replicate of $Y - M_{\mathcal{A}}$.*

As $Y = M_{\mathcal{A}} + (Y - M_{\mathcal{A}})$, the difference between $Y$ and $Y_{\mathcal{A}}$ is that $Y_{\mathcal{A}}$ neglects the covariances between $M_{\mathcal{A}}$ and the residual $Y - M_{\mathcal{A}}$. The process $Y_{\mathcal{A}}$ is centered with a kernel given for all $x, x' \in D$ by

$$\begin{aligned} k_{\mathcal{A}}(x, x') = k(x, x') &+ 2k_M(x)^t K_M^{-1}(x) K_M(x, x') K_M^{-1}(x') k_M(x') \\ &- k_M(x)^t K_M^{-1}(x) k_M(x, x') - k_M(x')^t K_M^{-1}(x') k_M(x', x)\,, \end{aligned} \tag{11}$$

The main interest of introducing $Y_{\mathcal{A}}$ is that it corresponds to a prior for which $M_{\mathcal{A}}$ is actually an optimal model : given a prior process $Y_{\mathcal{A}}$, aggregated values $M_{\mathcal{A}}$ and $v_{\mathcal{A}}$ can be seen as respective mean and variance of a posterior process $Y_{\mathcal{A}}|Y_{\mathcal{A}}(X)$:

**Proposition 3** (Bayesian perspective)**.** *If $M_{\mathcal{A}}$ is a deterministic and interpolating function of $Y(X)$, i.e. if for any $x \in D$ there exists a deterministic function $g_x : \mathbb{R}^n \to \mathbb{R}$ such that $M_{\mathcal{A}}(x) = g_x(Y(X))$ and if $M_{\mathcal{A}}(X) = Y(X)$, then*

$$\begin{cases} M_{\mathcal{A}}(x) = \mathrm{E}\left[Y_{\mathcal{A}}(x)|Y_{\mathcal{A}}(X)\right]\,, \\ v_{\mathcal{A}}(x) = \mathrm{V}\left[Y_{\mathcal{A}}(x)|Y_{\mathcal{A}}(X)\right]\,. \end{cases} \tag{12}$$

*Proof.* The interpolation hypothesis $M_{\mathcal{A}}(X) = Y(X)$ ensures $\varepsilon'_{\mathcal{A}}(X) = 0$ so we have

$$\begin{aligned} \mathrm{E}\left[Y_{\mathcal{A}}(x)|Y_{\mathcal{A}}(X)\right] &= \mathrm{E}\left[Y_{\mathcal{A}}(x)|M_{\mathcal{A}}(X) + 0\right] \\ &= \mathrm{E}\left[M_{\mathcal{A}}(x)|M_{\mathcal{A}}(X)\right] + \mathrm{E}\left[\varepsilon'_{\mathcal{A}}(x)|M_{\mathcal{A}}(X)\right] \\ &= \mathrm{E}\left[g_x(Y(X))|Y(X)\right] + 0 \\ &= M_{\mathcal{A}}(x). \end{aligned} \tag{13}$$

The proof that $v_{\mathcal{A}}$ is a conditional variance follows the same pattern:

$$\begin{aligned}\mathrm{V}\left[Y_{\mathcal{A}}(x)|Y_{\mathcal{A}}(X)\right] &= \mathrm{V}\left[Y_{\mathcal{A}}(x)|M_{\mathcal{A}}(X)\right] \\ &= \mathrm{V}\left[M_{\mathcal{A}}(x)|M_{\mathcal{A}}(X)\right] + \mathrm{V}\left[\varepsilon'_{\mathcal{A}}(x)|M_{\mathcal{A}}(X)\right] \\ &= 0 + \mathrm{V}\left[\varepsilon'_{\mathcal{A}}(x)\right] \\ &= v_{\mathcal{A}}(x).\end{aligned} \tag{14}$$

□

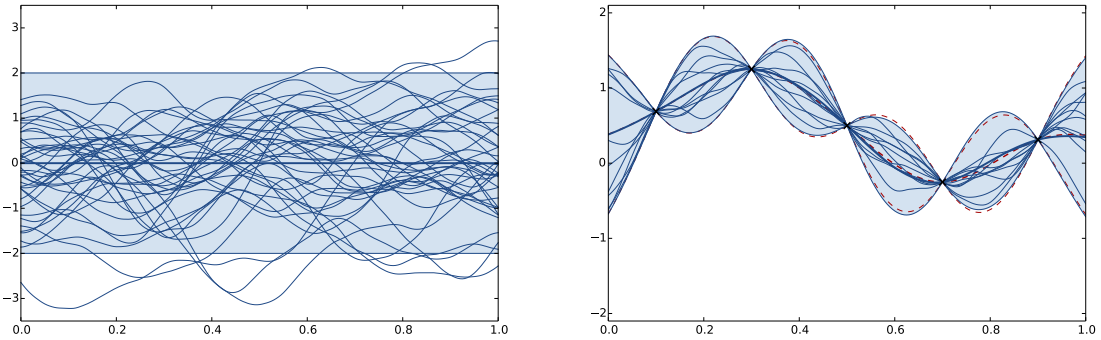
One great advantage of this Bayesian framework is to introduce the posterior conditional covariance:

$$c_{\mathcal{A}}(x, x') = \mathrm{Cov}\left[Y_{\mathcal{A}}(x), Y_{\mathcal{A}}(x')|Y_{\mathcal{A}}(X)\right]. \tag{15}$$

In the case where $(M, Y)$ is Gaussian, then $Y_{\mathcal{A}}$ is also Gaussian and Eq. (15) writes

$$c_{\mathcal{A}}(x, x') = k_{\mathcal{A}}(x, x') - k_{\mathcal{A}}(x, X)k_{\mathcal{A}}(X, X)^{-1}k_{\mathcal{A}}(X, x'). \tag{16}$$

This point of view thus allows to define conditional sample paths which was not possible in the previous section. Figure 3 is based on the same settings as in Example 1 and illustrate that: a. The mean and variance obtained by conditioning $Y_{\mathcal{A}}$ are the same as in the pointwise illustration (Figure 1) and b. The knowledge of a full posterior Gaussian Process allows to sample from the conditional distribution $Y_{\mathcal{A}}|Y_{\mathcal{A}}(X) = f(X)$. However, computational issues arise when calculating $c_{\mathcal{A}}$ or conditional samples of $Y_{\mathcal{A}}$ for large $n$. This exact conditional interpretation of the aggregation aims at explaining the approximation that is done when using $M_{\mathcal{A}}$ instead of the full model, but does not provide any computational gain.



(a) Prior samples.

(b) Posterior samples.

Figure 3: Interpretation of the results from Example 1 as a posterior Gaussian process distribution. (a) Samples from modified prior process $Y_{\mathcal{A}}$, which is normally distributed with mean 0 and covariance $k_{\mathcal{A}}$. (b) conditional sample paths of $Y_{\mathcal{A}}$ given $Y_{\mathcal{A}}(X) = f(X)$, which is normally distributed with mean $m_{\mathcal{A}}$ and covariance $c_{\mathcal{A}}$.

The new covariance $k_{\mathcal{A}}$ of the prior process $Y_{\mathcal{A}}$ can be shown to coincide with the one of the process $Y$ at several locations, as detailed in the following proposition.

**Proposition 4** (Covariance interpolation)**.** *For all $x \in D$, $Y(x)$ and $Y_{\mathcal{A}}(x)$ have the same variance: $k_{\mathcal{A}}(x, x) = k(x, x)$. Furthermore, if $M_{\mathcal{A}}$ is interpolating $Y$ at $X$, i.e. if $M_{\mathcal{A}}(X) = Y(X)$ then $k_{\mathcal{A}}(X, X) = k(X, X)$.*

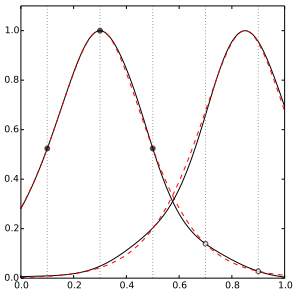
*Proof.* The first property of this proposition is a direct consequence of Eq. (11). The second one relies on the fact that $Y_{\mathcal{A}}(X) = Y(X)$ under the interpolation assumption. □

Figure 4 illustrates the difference between the covariance kernels of $k$ and $k_{\mathcal{A}}$, using the settings of Example 1. It shows that

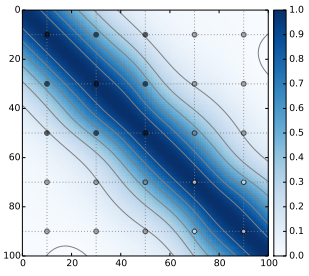
(a) the absolute difference between the two covariance functions $k$ and $k_{\mathcal{A}}$ is quite small. Furthermore, it illustrates the identity of Prop. 4 which states $k_{\mathcal{A}}(X, X) = k(X, X)$: as 0.3 is a component of $X$, $k_{\mathcal{A}}(0.3, x_k) = k(0.3, x_k)$ for any of the five components $x_k$ of $X$.

(b) the contour lines for $k_{\mathcal{A}}$ are not straight lines, as it is the case for covariance matrices of stationary processes evaluated on regular grids. In this example, $Y$ is stationary whereas $Y_{\mathcal{A}}$ is not. However, the latter only departs slightly from the stationary assumption.

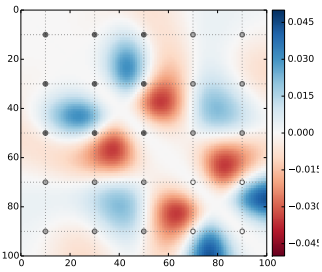
(c) the difference $k_{\mathcal{A}} - k$ vanishes at some places, among which are the places of the bullets points and the diagonal which correspond respectively to $k_{\mathcal{A}}(X, X) = k(X, X)$ and $k_{\mathcal{A}}(x, x) = k(x, x)$. Furthermore, the absolute differences between the two covariances functions are again quite small. It also shows that the pattern of the difference is quite complex.



(a) kernel functions $k_{\mathcal{A}}$ (solid lines) and $k$ (dashed lines) with one variable fixed to $0.3 \in X$ and $0.85 \notin X$.



(b) contour plot of the modified prior covariance matrix $k_{\mathcal{A}}(X_p, X_p)$.



(c) image plot of the difference between prior covariance matrices $k_{\mathcal{A}}(X_p, X_p) - k(X_p, X_p)$.

Figure 4: Comparisons of the modified covariance $k_{\mathcal{A}}$ and the initial covariance $k$. The vector of prediction points $X_p$ corresponds to 100 regularly spaced points spanning $[0, 1]$. The horizontal and vertical dotted lines correspond to locations of observed points $x_i$ for $i \in \{1, \ldots, 5\}$. The bullets indicate locations where $k_{\mathcal{A}}(x_i, x_j) = k(x_i, x_j)$.

## 2.3 Consistency

This section gives results on the consistency of the proposed aggregation method. Furthermore, we show that some other aggregation methods developed in the literature are not consistent.

**Proposition 5** (Consistency)**.** *Let $D$ be a compact subset of $\mathbb{R}^d$. Let $Y$ be a Gaussian process on $D$ with mean zero and continuous covariance function $k$. Let $(x_{ni})_{1 \leq i \leq n, n \in \mathbb{N}}$ be a triangular array of observation points so that $x_{ni} \in D$ for all $1 \leq i \leq n, n \in \mathbb{N}$ and so that for all $x \in D$, $\lim_{n \to \infty} \min_{i=1,\ldots,n} ||x_{ni} - x|| = 0$.*

*For $n \in \mathbb{N}$, let $\mathcal{A}_n = \{1, ..., p_n\}$ be the set of sub-model indexes and let $M_1(x), ..., M_{p_n}(x)$ be any collection of $p_n$ Kriging predictors based on respective design points $X_1, \ldots, X_{p_n}$. Assume that each component of $X = (x_{n1}, ..., x_{nn})$ is a component of at least one $X_i$, $1 \leq i \leq p_n$. Then we have*

$$\sup_{x \in D} \mathbb{E}\left( (Y(x) - M_{\mathcal{A}_n}(x))^2 \right) \to_{n \to \infty} 0. \tag{17}$$

The proof is given in Appendix A.

In the next proposition, we consider aggregations of Kriging predictors based on weighted sums of the conditional means of the predictors, which weights are functions of only their conditional variances. For certain conditions on the weight functions, we show that such aggregations methods can lead to mean square prediction errors that do not go to zero as $n \to \infty$, even in cases where the triangular array of observation points is dense in $D$.

**Proposition 6** (Non-consistency of variance based aggregations)**.** *Let $D$ be a compact non-empty subset of $\mathbb{R}^d$. Let $Y$ be a Gaussian process on $D$ with mean zero and stationary covariance function $k$. Assume that $k$ is defined on $\mathbb{R}^d$, continuous and satisfies $0 < k(x, y)$ for two distinct points $x, y \in D$ so that $D$ contains two open balls with strictly positive radii and centres $x$ and $y$. Assume also that $k$ has a positive spectral density (defined by $\hat{k}(\omega) = \int_{\mathbb{R}^d} k(x) \exp(Jx'\omega) dx$ with $J^2 = -1$ and for $\omega \in \mathbb{R}^d$). Assume that there exists $0 \leq A < \infty$ and $0 \leq T < \infty$ so that $1/\hat{k}(\omega) \leq A||\omega||^T$.*

*For each number $n$ of observation points, let $p_n$ be the number of Kriging predictors, let $X$ be the vector of the $n$ observation points, and let $X_1, ..., X_{p_n}$ be the vectors of observation points for the $p_n$ Kriging predictors. Let $\mathcal{A}_n = \{1, ..., p_n\}$. Let $v_k(x) = \mathbb{V}[Y(x)|Y(X_k)]$ and let $v_{prior}(x) = k(x,x)$. Consider an aggregated predictor of the form*

$$\bar{M}_{\mathcal{A}_n}(x) = \sum_{k=1}^{p_n} \alpha_{k,n}(v_1(x), ..., v_{p_n}(x), v_{prior}(x)) M_k(x)$$

*where*

$$\alpha_{k,n}(v_1(x), ..., v_{p_n}(x), v_{prior}(x)) \leq \frac{a(v_k(x), v_{prior}(x))}{\sum_{l=1}^{p_n} b(v_l(x), v_{prior}(x))},$$

*where $a$ and $b$ are given deterministic continuous functions from $\Delta = \{(x,y) \in (0,\infty)^2; x \leq y\}$ to $[0,\infty)$, with $a$ and $b$ positive on $\mathring{\Delta} = \{(x,y) \in (0,\infty)^2; x < y\}$.*

*Then, there exists a triangular array of observation points $(x_{ni})_{1 \leq i \leq n; n \in \mathbb{N}}$ so that $\lim_{n\to\infty} \sup_{x \in D} \min_{i=1,...,n} ||x_{ni} - x|| = 0$, a triangular array of subvectors $X_1, ..., X_{p_n}$ so that $X = (X_1, ..., X_{p_n})$, with $p_n \to_{n\to\infty} \infty$ and $p_n/n \to_{n\to\infty} 0$, and so that there exists $x_0 \in D$ so that*

$$\liminf_{n\to\infty} \mathrm{E}\left[\left(Y(x_0) - \bar{M}_{\mathcal{A}_n}(x_0)\right)^2\right] > 0. \tag{18}$$

The detailed proof is given in Appendix B. Its intuitive explanation is that the aggregation methods for which the proposition applies ignore the correlations between the different Kriging predictors. Hence, for prediction points around which the density of observation points is smaller than on average, too much weight can be given to Kriging predictors based on distant observation points.

As detailed in Section 5, examples of aggregation methods for which Prop. 6 applies are (generalized) products of experts and (robust) Bayesian committee machines, described in [Deisenroth and Ng, 2015, van Stein et al., 2015] and references therein. Furthermore, the assumptions made on $k$ in this proposition are satisfied by many stationary covariance functions, including those of the Matérn model, with the notable exception of the Gaussian covariance function (Proposition 1 in [Vazquez and Bect, 2010]).

## 2.4 Bounds on aggregation errors

We recall that, even in a non-Gaussian setting, $M_{full}(x) = k(x,X)k(X,X)^{-1}Y(X)$ and $v_{full}(x) = k(x,x) - k(x,X)k(X,X)^{-1}k(X,x)$. $M_{full}(x)$ corresponds to an optimal model but it cannot be used because of its computational burden so it is approximated by the aggregated model $M_{\mathcal{A}}(x)$. Similarly, $v_{\mathcal{A}}(x)$ aims at approximating $v_{full}(x)$ that can also be computationally intractable. This section aims at studying the differences between the aggregated model and full one.

For the analysis of these approximation errors we focus on the case where $M(x)$ is linear in $Y(X)$, i.e. there exists a $p \times n$ deterministic matrix $\Lambda(x)$ such that $M(x) = \Lambda(x)Y(X)$. Under this assumption, the differences write

$$\left\{\begin{array}{lcl} M_{\mathcal{A}}(x) - M_{full}(x) & = & -k(x,X)\Delta(x)Y(X)\,, \\ v_{\mathcal{A}}(x) - v_{full}(x) & = & k(x,X)\Delta(x)k(X,x)\,. \end{array}\right. \tag{19}$$

where $\Delta(x) = K^{-1} - \Lambda(x)^t\big(\Lambda(x)k(X,X)\Lambda(x)^t\big)^{-1}\Lambda(x)$, as soon as $\Lambda(x)k(X,X)\Lambda(x)^t$ is invertible.

**Proposition 7** (Bounds for maximal errors)**.** *Let $x \in D$. if $M(x)$ is linear in $Y(X)$, then for any norm $\|.\|$, there exists some constants $\lambda, \mu \in \mathbb{R}^+$ such that*

$$\left\{\begin{array}{lcl} |M_{\mathcal{A}}(x) - M_{full}(x)| & \leq & \lambda \|k(X,x)\| \|Y(X)\|\,, \\ |v_{\mathcal{A}}(x) - v_{full}(x)| & \leq & \mu \|k(X,x)\|^2\,. \end{array}\right. \tag{20}$$

*This implies that, if one can choose a prediction point $x$ far enough to observations in $X$, in the sense $\|k(X,x)\| \leq \epsilon$ for any given $\epsilon > 0$, $|M_{\mathcal{A}}(x) - M_{full}(x)|$ and $|v_{\mathcal{A}}(x) - v_{full}(x)|$ can be as small as desired. Furthermore, if $M$ is interpolates $Y$ at $X$ (see definition in Prop. 2), then since $v_{full}(x) = \mathrm{E}\left[(Y(x) - M_{full}(x))^2\right]$:*

$$0 \leq v_{\mathcal{A}}(x) - v_{full}(x) \leq \min_{k \in \{1,...,n\}} \mathrm{E}\left[(Y(x) - M_k(x))^2\right] - v_{full}(x)\,. \tag{21}$$

*Proof.* For the first part of the proposition, $\Delta(x)$ is the difference of two positive semi-definite matrices. After expanding Eq. (19) both terms can thus be interpreted as differences of inner products. We can thus conclude using successive application of triangular inequality, Cauchy-Schwartz inequality, and equivalence of norms for finite-dimensional real vector spaces. Regarding the second part, the upper bound comes from the fact that $M_{\mathcal{A}}(x)$ is the best linear combination of $M_k(x)$ for $k \in \{1, \ldots, n\}$. The positivity of $v_{\mathcal{A}} - v_{full}$ can be proved similarly: $M_{\mathcal{A}}(x)$ is a linear combination of $Y(x_k)$, $k \in \{1, \ldots, n\}$, whereas $M_{full}(x)$ is the best linear combination. □

One consequence of previous proposition is that when the covariances between the prediction point $x$ and the observed ones $X$ become small, both models tend to predict the unconditional distribution of $Y(x)$. This is a natural property that is desirable for any aggregation method but it is not always fulfilled (see for instance PoE in [Deisenroth and Ng, 2015]).

We have seen in Figure 4 that the kernels $k$ and $k_{\mathcal{A}}$ are very similar. The following proposition gives a link between the aggregation errors and the kernel differences.

**Proposition 8** (Errors as kernel differences)**.** *Assume that for all $x \in D$, $M(x)$ is a linear function of $Y(X)$ and that $M$ interpolates $Y$ at $X$, i.e. if for any component $x_k$ of the vector $X$ there is at least one index $i_k \in \mathcal{A}$ such that $M_{i_k}(x_k) = Y(x_k)$, then the differences between the full and aggregated models write as differences between kernels :*

$$\begin{cases} \mathrm{E}\left[(M_{\mathcal{A}}(x) - M_{full}(x))^2\right] = \|k(X,x) - k_{\mathcal{A}}(X,x)\|_K^2 \,, \\ v_{\mathcal{A}}(x) - v_{full}(x) = \|k(X,x)\|_K^2 - \|k_{\mathcal{A}}(X,x)\|_K^2 \,, \end{cases} \tag{22}$$

*where $\|u\|_K^2 = u^t k(X,X)^{-1} u$. Assuming the the smallest eigenvalue $\lambda_{\min}$ of $k(X,X)$ is non zero, this norm can be bounded by $\|u\|_K^2 \leq \frac{1}{\lambda_{\min}} \|u\|^2$ where $\|u\|$ denotes the Euclidean norm.*

*Proof.* The first equality comes from $k(X,X) = k_{\mathcal{A}}(X,X)$ under interpolation assumption, which leads to $M_{\mathcal{A}}(x) - M_{full}(x) = (k(x,X) - k_{\mathcal{A}}(x,X))k(X,X)^{-1}Y(X)$. The second equality uses both $k(X,X) = k_{\mathcal{A}}(X,X)$ and $k(x,x) = k_{\mathcal{A}}(x,x)$ which leads to $v_{\mathcal{A}}(x) = k(x,x) - k_{\mathcal{A}}(x,X)k(X,X)^{-1}k_{\mathcal{A}}(X,x)$. The result is then obtained by subtracting $v_{full}(x)$. Finally, the classical inequality between $\|.\|_K$ and $\|.\|$ derives from the diagonalization of $k(X,X)$. □

The difference between the full model and the aggregated one of Example 1 is illustrated in Figure 5. Various remarks can be made on this figure. First, the difference between the aggregated and full model is small, both on the predicted means and variances. Second, the error tends toward 0 when the prediction point $x$ is far away from the observations $X$. This illustrates Prop. 7 in the case where $\|k(X,x)\|$ is small. Third, it can be seen that the bounds on the left panel are relatively tight on this example, and that both the errors and their bounds vanish at observation points. At last, the right panel shows $v_{\mathcal{A}}(x) \geq v_{full}(x)$. This is because the estimator $M_{\mathcal{A}}$ is expressed as successive optimal linear combinations of $Y(X)$, which have a quadratic error necessarily greater or equal than $M_{full}$ which is the optimal linear combination of $Y(X)$. Panel (b) also illustrates that the bounds given in Eq. (21) are relatively loose. This means that the nested aggregation is more informative than the most accurate sub-model.

At last, the following result gives another optimality property that is often not satisfied by other aggregating methods: if the sub-models contain enough information, the aggregated one corresponds to the full model.

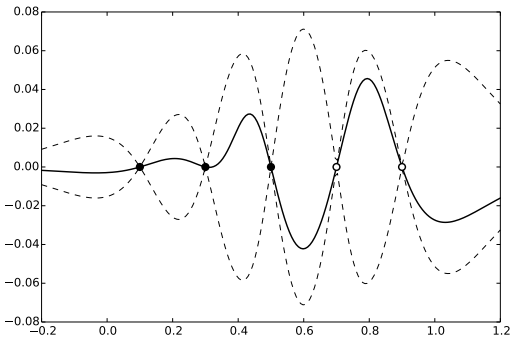
**Proposition 9** (Fully informative sub-models)**.** *Assume $M(x)$ is linear in $Y(X)$: $M(x) = \Lambda(x)Y(X)$ and that $\Lambda(x)$ is a $n \times n$ matrix with full rank, then*

$$\begin{cases} M_{\mathcal{A}}(x) &= M_{full}(x) \,, \\ v_{\mathcal{A}}(x) &= v_{full}(x) \,. \end{cases} \tag{23}$$

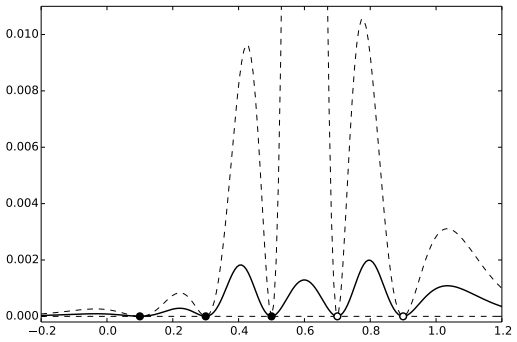
*Furthermore, if $Y$ and $Y_{\mathcal{A}}$ are Gaussian processes then*

$$Y_{\mathcal{A}} \overset{law}{=} Y \quad \text{and thus} \quad Y_{\mathcal{A}}|Y_{\mathcal{A}}(X) \overset{law}{=} Y|Y(X). \tag{24}$$

*In other words, there is no difference between the full and the approximated models when $\Lambda(x)$ is invertible. This property can be extended if $\Lambda(x)$ is $p \times n$ with $p > n$ (more sub-models than observation points) but it requires replacing matrix inverses by pseudo-inverses in Definition 1.*



(a) differences between predicted means $m_{\mathcal{A}}(x) - m_{full}(x)$.



(b) differences between predicted variances $v_{\mathcal{A}}(x) - v_{full}(x)$.

Figure 5: Comparisons of the full and aggregated model. The dashed lines correspond to the bounds given in Prop. 8: $\pm\lambda_{\min}^{-1/2}\|k(X,x) - k_{\mathcal{A}}(X,x)\|$ on panel (a) and bounds of Eq.(21) on panel (b).

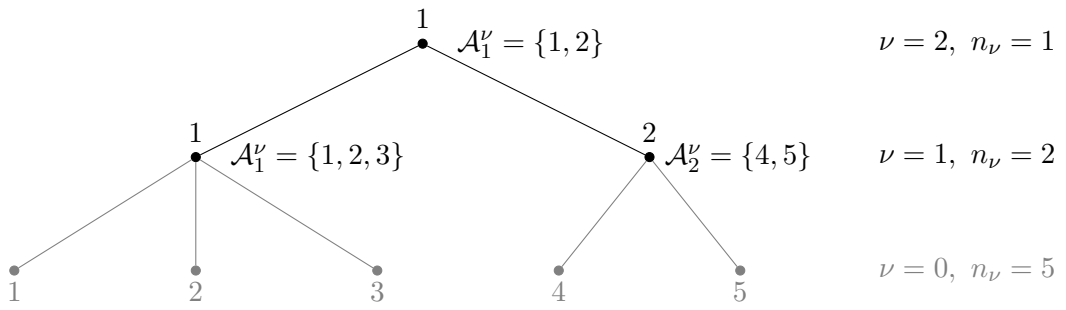



Figure 6: One aggregation tree with height $\bar{\nu} = 2$, $n_0 = 5$ initial leave nodes (observation points) and $n_1 = 2$ sub-models.

*Proof.* As $\Lambda(x)$ is invertible, the expression of $\lambda_{\mathcal{A}}(x)$ from Prop. 2 *(i)* is $\lambda_{\mathcal{A}}(x)^t = k(x,X)k(X,X)^{-1}$ which leads to Eq. (23). As $M_{\mathcal{A}} = M_{full}$, we have $Y_{\mathcal{A}} = M_{full} + \varepsilon$ where $\varepsilon$ is an independent copy of $Y - M_{full}$. Furthermore $Y = M_{full} + Y - M_{full}$ where $M_{full}$ and $Y - M_{full}$ are independent in the Gaussian case so $Y_{\mathcal{A}} \overset{law}{=} Y$. □

Note that there is of course no computational interest in building and merging fully informative sub-models since it requires computing and inverting a matrix that has the same size as $k(X,X)$ so there is no complexity gain compared to the full model.

# 3 Iterative scheme

In the previous sections, we have seen how to aggregate sub-models $M_1, \ldots, M_p$ into one unique aggregated value $M_{\mathcal{A}}$. Now, starting from the same sub-models, one can imagine creating several aggregated values, $M_{\mathcal{A}_1}, \ldots, M_{\mathcal{A}_s}$, each of them based on a subset of $\{M_1, \ldots, M_p\}$. One can show that these aggregated values can themselves be aggregated. This makes possible the construction of an iterative algorithm that merges sub-models at successive steps, according to a tree structure. Such tree-based schemes are sometimes used to reduce the complexity of models, see e.g. [Tzeng et al., 2005], or to allow parallel computing [Wei et al., 2015].

The aim of this section is to give a generic algorithm for aggregating sub-models according to a tree structure and to show that the choice of the tree structure helps partially reducing the complexity of the algorithm. It also aims at giving perspectives for further large reduction of the global complexity.

Let us introduce some notations. The total height (i.e number of layers) of the tree is denoted $\bar{\nu}$ and the number of node of a layer $\nu \in \{1, \dots, \bar{\nu}\}$ is $n_\nu$. We associate to each node (say node $i$ in layer $\nu$) a sub-model $M_i^\nu$ corresponding to the aggregation of its child node sub-models. In other words, $M_i^\nu$ is the aggregation of $\{M_k^{\nu-1},\, k \in \mathcal{A}_i^\nu\}$ where $\mathcal{A}_i^\nu$ is the set of childs of node $i$ in layer $\nu$. These notations are summarized in Figure 6 which details the tree associated to Example 1. In practice, there will be one *root node* ($n_{\bar{\nu}} = 1$) and each node will have at least one parent: $\cup_{i=1,\dots,n_\nu} \mathcal{A}_i^\nu = \{1, \dots, n_{\nu-1}\}$. Typically, the sets $\mathcal{A}_i^\nu$, $i = 1, \dots, n_\nu$, are a partition of $\{1, \dots, n_{\nu-1}\}$ but this assumption is not required and a child node may have several parents (which can generate a lattice rather than a tree).

## 3.1 Two-Layer aggregation

We discuss in this section the tree structure associated to the case $\bar{\nu} = 2$ as per the previous examples. With such settings, the first step consists in calculating the initial sub-models $M_1^1, \dots, M_p^1$ of the layer $\nu = 1$ and the second one is to aggregate all sub-models of layer $\nu = 1$ into one unique estimator $M_1^2$ (see for example Figure 6). This aggregation is obtained by direct application of Definition 1.

In practice the sub-models can be any covariates, like gradients, non-Gaussian underlying factors or even black-box responses, as soon as cross-covariances and covariances with $Y(x)$ are known. When sub-models are calculated from direct observations $Y(x_1), \dots, Y(x_n)$, the number of leave nodes at layer $\nu = 0$ is $n_0 = n$. In further numerical illustrations of Section 5, the sub-models $M_i^1$ are simple Kriging predictors of $Y(x)$, with for $i = 1, \dots, p$,

$$\left\{ \begin{array}{lcl} M_i^1(x) & = & k(x, X_i) k(X_i, X_i)^{-1} Y(X_i)\,, \\ \mathrm{Cov}\left[M_i^\nu(x), Y(x)\right] & = & k(x, X_i) k(X_i, X_i)^{-1} k(X_i, x)\,, \\ \mathrm{Cov}\left[M_i^\nu(x), M_j^\nu(x)\right] & = & k(x, X_i) k(X_i, X_i)^{-1} k(X_i, X_j) k(X_j, X_j)^{-1} k(X_j, x)\,. \end{array} \right. \tag{25}$$

With these particular simple Kriging initial sub-models, the layer $\nu = 1$ corresponds to the aggregation of covariates $M_i^0(x) = Y(x_i)$ at the previous layer $\nu = 0$, $i = 1, \dots, n$.

## 3.2 Multiple Layer aggregation

In order extend the two-layer setting, one needs to compute covariances among aggregated sub-models. The following proposition gives covariances between aggregated models of a given layer.

**Proposition 10** (aggregated models covariances)**.** *Let us consider a layer $\nu \geq 1$ and given aggregated models $M_1^\nu(x), \dots, M_{n_\nu}^\nu(x)$. Assume that the following covariances $(k^\nu(x))_i = \mathrm{Cov}\left[M_i^\nu(x), Y(x)\right]$ and $(K^\nu(x))_{ij} = \mathrm{Cov}\left[M_i^\nu(x), M_j^\nu(x)\right]$ are given, $i, j \in \{1, \dots, n_\nu\}$. Let $n_{\nu+1} \geq 1$ be a number of new aggregated values. Consider subsets $\mathcal{A}_i^{\nu+1}$ of $\{1, \dots, n_\nu\}$, $i = 1, \dots, n_{\nu+1}$, and assume that $M_i^{\nu+1}(x)$ is the aggregation of $M_k^\nu(x)$, $k \in \mathcal{A}_i^{\nu+1}$. Then*

$$\left\{ \begin{array}{lcl} (M^{\nu+1}(x))_i & = & \alpha_i^{\nu+1}(x)^t \left(M^\nu(x)_{[\mathcal{A}_i^{\nu+1}]}\right)\,, \\ \mathrm{Cov}\left[M_i^{\nu+1}(x), Y(x)\right] & = & \alpha_i^{\nu+1}(x)^t \left(k^\nu(x)_{[\mathcal{A}_i^{\nu+1}]}\right)\,, \\ \mathrm{Cov}\left[M_i^{\nu+1}(x), M_j^{\nu+1}(x)\right] & = & \alpha_i^{\nu+1}(x)^t \left(K^\nu(x)_{[\mathcal{A}_i^{\nu+1}, \mathcal{A}_j^{\nu+1}]}\right) \alpha_j^{\nu+1}(x)\,, \end{array} \right. \tag{26}$$

*where the vectors of optimal weights are $\alpha_i^{\nu+1}(x) = \left(K^\nu_{[\mathcal{A}_i^{\nu+1}, \mathcal{A}_i^{\nu+1}]}\right)^{-1} \left(k^\nu(x)_{[\mathcal{A}_i^{\nu+1}]}\right)$ and where $k^\nu(x)_{[\mathcal{A}_i^{\nu+1}]}$ corresponds to the sub-vector of $k^\nu(x)$ of indices in $\mathcal{A}_i^{\nu+1}$ and similarly for $M^\nu(x)_{[\mathcal{A}_i^{\nu+1}]}$ and the submatrix $K^\nu(x)_{[\mathcal{A}_i^{\nu+1}, \mathcal{A}_i^{\nu+1}]}$, which is assumed to be invertible.*
*Furthermore,* $\mathrm{Cov}\left[M_i^{\nu+1}(x), Y(x)\right] = \mathrm{Cov}\left[M_i^{\nu+1}(x), M_i^{\nu+1}(x)\right]$.

*Proof.* This follows immediately from Definition 1: as the aggregated values are linear expressions, the calculation of their covariances is straightforward. The last equality is simply obtained by inserting the value of $\alpha_i^{\nu+1}(x)$ into the expression of $\mathrm{Cov}\left[M_i^{\nu+1}(x), M_i^{\nu+1}(x)\right]$. □

The following algorithm, which is a generic algorithm for aggregating sub-models according to a tree structure, is based on an iterative use of the previous proposition. It is given for one prediction point $x \in D$ and it assumes that the sub-models are already calculated, starting directly from layer

1. This allows a large variety of sub-models, and avoids storage the storage of the possibly large covariance matrix $K^0(x)$. Its outputs are the final scalar aggregated model, $M_{\bar{\nu}}(x)$, and the scalar covariance $K_{\bar{\nu}}(x)$ from which one deduces the prediction error $\mathrm{E}\left[(Y(x) - M_{\bar{\nu}}(x))^2\right] = k(x,x) - K_{\bar{\nu}}(x)$.

In order to give dimensions in the algorithm and to ease the calculation of complexities, we define $c_i^\nu$ as the number of childs of the sub-model $M_i^\nu$, $c_i^\nu = \operatorname{card} \mathcal{A}_i^\nu$. We also denote $c_{\max} = \max_{\nu,i} c_i^\nu$ the maximal number of childs.

**Algorithm 1:** Nested Kriging algorithm

**inputs** : $M_1$, vector of length $n_1$ (sub-models evaluated at $x$)
$k_1$, vector of length $n_1$ (covariance between $Y(x)$ and sub-models at $x$)
$K_1$, matrix of size $n_1 \times n_1$ (covariance between sub-models at $x$)
$\mathcal{A}$, a list describing the tree structure
**outputs**: $M_{\bar{\nu}}$, $K_{\bar{\nu}}$

Create vectors $M$, $k$ of size $c_{\max}$ and matrix $K$ of size $c_{\max} \times c_{\max}$
**for** $\nu = 2, \ldots, \bar{\nu}$ **do**
  Create vectors $M_\nu$ of size $n_\nu$ and matrix $K_\nu$ of size $n_\nu \times n_\nu$
  **for** $i = 1, \ldots, n_\nu$ **do**
    Create vector $\alpha_i$ of size $c_i^\nu$
    $M \leftarrow$ subvector of $M_{\nu-1}$ on $\mathcal{A}_i^\nu$
    $K \leftarrow$ submatrix of $K_{\nu-1}$ on $\mathcal{A}_i^\nu$
    **if** $\nu = 2$ **then** $k \leftarrow k_1$ **else** $k \leftarrow \mathrm{Diag}(K)$
    $\alpha_i \leftarrow K^{-1}k$
    $M_\nu[i] \leftarrow (\alpha_i)^t M$
    $K_\nu[i,i] \leftarrow (\alpha_i)^t k$
    **for** $j = 1, \ldots, i-1$ **do**
      $K \leftarrow$ submatrix of $K_{\nu-1}$ on $\mathcal{A}_i^\nu \times \mathcal{A}_j^\nu$
      $K_\nu[i,j] \leftarrow (\alpha_i)^t K \alpha_j$
      $K_\nu[j,i] \leftarrow K_\nu[i,j]$
  $M_{\nu-1}$, $K_{\nu-1}$ and all $\alpha_i$ can be deleted

Notice that Algorithm 1 uses the result $(K^{\nu+1}(x))_{ii} = (k^{\nu+1}(x))_i$ from Prop. 10: when we consider aggregated models ($\nu \geq 2$), we do not need to store and compute the vector $k^\nu(x)$ any more. When $\nu = 1$, depending on the initial covariates, $\mathrm{Cov}\left[M_i^1(x), Y(x)\right]$ is not necessarily equal to $\mathrm{Cov}\left[M_i^1(x), M_i^1(x)\right]$ (this is however the case when $M_i^1(x)$ are simple Kriging predictors).

For the sake of clarity, some improvements have been omitted in the algorithm above. For instance, covariances can be stored in triangular matrices, one can store two couples $(M_\nu, K_\nu)$ instead of $\bar{\nu}$ couples by using objects $M_{(\nu \mod 2)}$ and $K_{(\nu \mod 2)}$. Furthermore, it is quite natural to adapt this algorithm to parallel computing, but this is out of the scope of this article.

### 3.3 Complexity

We study here the complexity of Algorithm 1 in space (storage footprint) and in time (execution time). For the sake of clarity we consider in this paragraph a simplified tree where $n_\nu$ is decreasing in $\nu$ and each child has only one parent. This corresponds to the most common structure of trees, without overlapping. Furthermore, at any given level $\nu$, we consider that each node has the same number of childs: $c_i^\nu = c_\nu$ for all $i = 1, \ldots, n_\nu$. Such a tree will be called *regular*. In this setting, one easily sees that $n_\nu = \frac{n_{\nu-1}}{c_\nu} = \frac{n}{c_1 \ldots c_\nu}$, $\nu \in \{1, \ldots, \bar{\nu}\}$. Complexities obviously depend on the choice of sub-models, we give here complexities for Kriging sub-models as in Eq. (25), but this can be adapted to other kind of sub-models.

For one prediction point $x \in D$, we denote by $\mathcal{S}$ the storage footprint of Algorithm 1, and by $\mathcal{C}$ its complexity in time, including sub-models calculation. One can show that in a particular two-layers setting with $\sqrt{n}$ sub-models ($\bar{\nu} = 2$ and $c_1 = c_2 = \sqrt{n}$), a reachable global complexity for $q$ prediction points is (see assumptions below and expression details in the proof of Proposition 11)

$$\mathcal{S} = O(n) \quad \text{and} \quad q\mathcal{C} = O(n^2 q)\,. \tag{27}$$

This is to be compared with $O(n^3) + O(n^2 q)$ for the same prediction with the full model. The aggregation of sub-models can be useful when the number of prediction points is smaller than the number of observations. Notice that the storage needed for $q$ prediction points is the same as for one prediction point, but in some cases (as for leave-one-out errors calculation), it is worth using a $O(nq)$ storage to avoid recalculations of some quantities.

We now detail chosen assumptions on the calculation of $\mathcal{S}$ and $\mathcal{C}$, and study the impact of the tree structure on these quantities. For one prediction point $x \in D$, including sub-models calculation, the complexity in time can be decomposed into $\mathcal{C} = \mathcal{C}_{\text{cov}} + \mathcal{C}_\alpha + \mathcal{C}_\beta$, where

- $\mathcal{C}_{\text{cov}}$ is the complexity for computing all cross covariances among initial design points, which does not depend on the tree structure (neither on the number of prediction points).
- $\mathcal{C}_\alpha$ is the complexity for building all aggregation predictors, i.e. the sum over $\nu, i$ of all operations in the $i$-loop in the Algorithm 1 (excluding operations in the $j$-loop).
- $\mathcal{C}_\beta$ is the complexity for building the covariance matrices among these predictors, i.e. the sum over $\nu, i, j$ of all operations in the $j$-loop in the Algorithm 1.

We assume here that there exists two constants $\alpha > 0$ and $\beta > 0$ such that the complexity of operations inside the $i$-loop (excluding those of the $j$-loop) is $\alpha c_\nu^3$, and the complexity of operations inside the $j$-loop is $\beta c_\nu^2$. Despite perfectible, this assumption follows from the fact that one usually considers that the complexity of $c_\nu \times c_\nu$ matrix inversion is $O(c_\nu^3)$ and the complexity of matrix-vector multiplication is $O(c_\nu^2)$. We also assume that the tree height $\bar{\nu}$ is finite, and that all numbers of childs $c_\nu$ tend to $+\infty$ as $n$ tends to $+\infty$. This excludes for example binary trees, but makes assumptions on complexities more reliable. Under these assumptions, the following proposition details how the tree structure affects the complexities.

**Proposition 11** (Complexities)**.** *The following storage footprint $\mathcal{S}$ and complexities $\mathcal{C}_\alpha$, $\mathcal{C}_\beta$ hold for the respective tree structures, when the number of observations $n$ tend to $\infty$.*

*(i) The two-layer equilibrated $\sqrt{n}$-tree, where $p = c_1 = c_2 = \sqrt{n}$, $\bar{\nu} = 2$, is the optimal storage footprint tree, and*

$$\mathcal{S} = O(n)\,, \qquad \mathcal{C}_\alpha \sim \alpha n^2\,, \qquad \mathcal{C}_\beta \sim \frac{\beta}{2} n^2\,. \tag{28}$$

*(ii) The $\bar{\nu}$-layer equilibrated $\sqrt[\bar{\nu}]{n}$-tree, where $c_1 = \cdots = c_{\bar{\nu}} = \sqrt[\bar{\nu}]{n}$, $\bar{\nu} \geq 2$, is such that*

$$\mathcal{S} = O(n^{2-2/\bar{\nu}})\,, \qquad \mathcal{C}_\alpha \sim \alpha n^{1+\frac{2}{\bar{\nu}}}\,, \qquad \mathcal{C}_\beta \sim \frac{\beta}{2} n^2\,. \tag{29}$$

*(iii) The optimal complexity tree is defined as the regular tree structure that minimizes $\mathcal{C}_\alpha$, as it is not possible to reduce $\mathcal{C}_\beta$ to lower orders than $O(n^2)$. This tree is such that*

$$\mathcal{S} = O\left(n^{2-\frac{1}{\delta^{\bar{\nu}}-1}}\right)\,, \qquad \mathcal{C}_\alpha \sim \gamma\alpha n^{1+\frac{1}{\delta^{\bar{\nu}}-1}}\,, \qquad \mathcal{C}_\beta \sim \frac{\beta}{2} n^2\,, \tag{30}$$

*with $\delta = \frac{3}{2}$ and $\gamma = \frac{27}{4}\delta^{-\frac{\bar{\nu}}{\delta^{\bar{\nu}}-1}}(1-\delta^{-\bar{\nu}})$. This tree is obtained for $c_\nu = \delta\,(\delta^{-\bar{\nu}} n)^{\frac{\delta^{(\nu-1)}}{2(\delta^{\bar{\nu}}-1)}}$, $\nu = 1, \ldots, \bar{\nu}$. In a particular two-layers setting one gets $c_1 = \left(\frac{3}{2}\right)^{1/5} n^{2/5}$ and $c_2 = \left(\frac{3}{2}\right)^{-1/5} n^{3/5}$, which leads to $\mathcal{C}_\alpha = \gamma\alpha n^{9/5}$ and $\mathcal{C}_\beta = \frac{\beta}{2}n^2 - \frac{\beta}{2}\left(\frac{3}{2}\right)^{\frac{1}{5}} n^{\frac{7}{5}}$, where $\gamma = (\frac{2}{3})^{-2/5} + (\frac{2}{3})^{3/5} \simeq 1.96$.*

*Proof.* The details of the proof are given in Appendix C. □

We have seen that for $q$ prediction points and $n$ observations, a reachable complexity of the algorithm is $O(n^2 q)$, which is less than $O(n^3) + O(n^2 q)$ for the same prediction with the full model, when $q < n$.

More precisely, we have shown that the choice of the tree structure helps partially reducing the complexity of the algorithm. Indeed, a large tree height $\bar{\nu}$ largely reduces the complexity $\mathcal{C}_\alpha$ of matrix inversions in the algorithm. However, $\mathcal{C}_\beta$ cannot be reduced and one can expect a maximal complexity reduction factor of $\frac{\beta}{2\alpha+\beta}$ when using an optimal tree, compared to the equilibrated two-layers $\sqrt{n}$-tree. One shall however keep in mind that a lower complexity can lead to larger prediction errors or larger storage footprint.

As a perspective, approximating cross-covariances between aggregated models would allow to reduce $\mathcal{C}_\beta$ to the same order than $\mathcal{C}_\alpha$, which approach $O(n)$ when $\bar{\nu}$ is large. This thus give perspectives for further large reduction of the global complexity, which are let to future work.

At last, several parts of the algorithm can be computed in parallel execution threads. This is an interesting feature since sub-models computation at any layer can also be distributed.

# 4 Parameter estimation

Consider a set of covariance functions $\{\sigma^2 k_\theta, \sigma^2 \geq 0, \theta \in \Theta\}$ where $k_\theta$ is a correlation function from $D \times D$ into $[-1, 1]$ depending on some parameters $\theta$ such as length-scales. In this section, we address the problem of selecting the value of $\sigma^2$ and $\theta$ from the input observation points in $X$ and the observation vector $f(X)$. The mean predictor $m_{\mathcal{A}}$ depends only on $\theta$ so it will be written $m_{\mathcal{A},\theta}$. The prediction variance is a function of both $\theta$ and $\sigma^2$. Since it is linear in the latter, the prediction variance is written $\sigma^2 v_{\mathcal{A},\theta}$.

Let, for $1 \leq i \leq n$, the leave-one-out means $m_{\mathcal{A},\theta,-i}(x_i)$ be computed as $m_{\mathcal{A},\theta}(x_i)$, but with $X, f(X)$ replaced by $X_{-i}, f(X_{-i})$, where $X_{-i}$ is obtained by removing the $i$th line of $X$. Note that the input division $X_1, ..., X_p$ is left unchanged, apart from removing $x_i$ when it appears in $X_1, ..., X_p$. Similarly, the tree structure $\{\mathcal{A}_i^\nu\}$ is left unchanged. We define $\sigma^2 v_{\mathcal{A},\theta,-i}(x_i)$ similarly from $\sigma^2 v_{\mathcal{A},\theta}(x_i)$.

We estimate $\sigma^2$ and $\theta$ with a two-step leave-one-out procedure similar to that of [Bachoc, 2013]. We first select $\theta$ as minimizing the leave-one-out mean square error. We let

$$\widehat{\theta} \in \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \left( f(x_i) - m_{\mathcal{A},\theta,-i}(x_i) \right)^2 . \tag{31}$$

Second, we set $\sigma^2$ so that the leave-one-out errors have variance one:

$$\widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{\left( f(x_i) - m_{\mathcal{A},\widehat{\theta},-i}(x_i) \right)^2}{v_{\mathcal{A},\widehat{\theta},-i}(x_i).} \tag{32}$$

We implemented an algorithm which computes, for a given covariance parameter $\theta$, the quantities $m_{\mathcal{A},\theta,-i}(x_i)$ and $v_{\mathcal{A},\theta,-i}(x_i)$ for $q$ different values of $x_i$. If the proper storage and precomputations are made, the computational cost is of the order $O(qn^2)$, which is similar to the cost for predicting in $q$ new locations using the model aggregation procedure presented in this paper. However using precomputations, the algorithm also has a storage cost of $O(nq)$ which excludes using $q = n$ in the case where $n$ is large and prevents computing the right-hand side of Equation (31) exactly. Finally, one may notice that when $q$ points are chosen uniformly, without replacement, in the set of all $n$ points, averaging $q$ leave-one-out mean square error yields an unbiased estimate of the leave-one out mean square error, and can be seen as an approximation of the latter. We thus propose to solve the optimization problem (31) with a stochastic gradient descent algorithm described in Chapter 5 of [Bhatnagar et al., 2013]. At each step of the gradient descent, the projection of the gradient of (31) on a random direction is approximated by a finite difference. The algorithm is as follows.

**Algorithm 2:** Stochastic gradient descent

**inputs** : $\theta_0$, initial value of $\theta$
$(a_i)_{i \in \mathbb{N}}$, sequence of increment terms for the gradient descent
$(\delta_i)_{i \in \mathbb{N}}$, sequence of step sizes for the finite differences
$q$, number of leave-one-out predictions
$n_{iter}$, maximal number of iterations

**outputs**: $\hat{\theta}$

**for** $i = 1, ..., n_{iter}$ **do**

Sample a subset $\mathcal{I}_i$ of $\{1, ..., n\}$, uniformly over all the subsets of $\{1, ..., n\}$ with cardinality $q$.
Sample a $m$-dimensional vector $h_i$ from a $m$-dimensional random vector with independent components, each of them taking the values 1 and $-1$ with probabilities $1/2$.
Let

$$\Delta_i = \frac{1}{2\delta_i} \left( \frac{1}{q} \sum_{j \in \mathcal{I}_i} \left( f(x_j) - m_{\mathcal{A},-j,\theta_{i-1}+\delta_i h_i}(x_j) \right)^2 - \frac{1}{q} \sum_{j \in \mathcal{I}_i} \left( f(x_j) - m_{\mathcal{A},-j,\theta_{i-1}-\delta_i h_i}(x_j) \right)^2 \right).$$

Let $\theta_i = \theta_{i-1} - a_i \Delta_i h_i$.

Let $\hat{\theta} = \theta_{n_{iter}}$.

An implementation in R and C++ of both algorithms 1 and 2 is publicly available on the website `http://blinded` (available upon request to the editor). In practice, the computation cost of $q$ leave-one-out predictions is the sum of a fixed cost – involving in particular the computation of the $n^2$ covariances $k_\theta(x_i, x_j)$ – and a marginal cost which is proportional to $q$. When $n = 10,000$, these two summands take comparable values for $q = 100$, which is the setting we use in practice. Following the recommendations in [Bhatnagar et al., 2013], we set $\delta_i = c/(i+1)^\gamma$, with $\gamma = 0.101$. We set $a_i = a/(A+i+1)^\alpha$, with $\alpha = 0.602$ (as suggested in [Bhatnagar et al., 2013]), or $\alpha = 0.2$, or a combination of these two values. Typically we run a first gradient descent with $\alpha = 0.2$, which termination point serves as starting point for a second gradient descent with $\alpha = 0.601$. Good values of $a$, $c$ and $A$ depend of the application case. In practice, satisfactory results are obtained for $n = 10,000$, $d = 10$ and $p = 100$, with $n_{iter} = 500$, in which case the computation time would be around a few hours on a personal computer.

# 5 Numerical applications

## 5.1 Comparison with other aggregation methods

We now compare the predictions obtained with various methods when aggregating 15 Kriging sub-models based on two observations each. The test functions are samples of a centered Gaussian process over $[0, 1]$. The compared models are the nested Kriging model introduced in this article, the full model and other methods developed in the literature:

**Product of expert (PoE)** [Hinton, 2002] is based on the assumption that for a given $x$, the predictions of each sub-model correspond to independent random variables. As a consequence, the aggregated predicted density for $Y(x)$ is equal to the product of the sub-models densities : $f_{poe}(y) \propto \prod_{i=1}^{p} f_i(y)$ where $f_i$ is the predicted density of $Y(x)$ according to the $i$th sub-model. The PoE corresponds to the normal model developed in [Winkler, 1981], in the case of independent experts, when the considered covariance matrix is diagonal (see e.g. section 3.2 in the previously cited article and [van Stein et al., 2015]). Some extensions of this method to consensus Monte-Carlo sampling can be found in [Scott et al., 2016].

**Generalised product of expert (GPoE)**. As discussed in [Deisenroth and Ng, 2015], a major drawback of Kriging based PoE is that the prediction variance of the aggregated model decreases when the number sub-model increases even in regions with no observation points. [Cao and Fleet, 2014] introduced a variant called generalised product of expert where a weighting term is added to overcome this issue. The prediction is then given by

$$f_{gpoe}(y) \propto \prod_{i=1}^{p} (f_i(y))^{\beta_i}. \tag{33}$$

For this benchmark, the parameters $\beta_i$ will be set to $1/p$ as recommended in [Deisenroth and Ng, 2015]. Notice that GPoE corresponds exactly to what consensus literature refers to *logarithmic opinion pool*, see e.g. Eq.(3.11) in [Genest and Zidek, 1986].

**Bayesian Committee Machine (BCM)** has been introduced in [Tresp, 2000] to aggregate Kriging sub-models. It is based on the assumption of conditional independence of the sub-models given the process values at prediction points. The predicted aggregated density is given by

$$f_{bcm}(y) \propto \frac{\prod_{i=1}^{p} f_i(y)}{f_Y(y)^{p-1}}. \tag{34}$$

**Robust Bayesian committee machine (RBCM)** has been introduced in [Deisenroth and Ng, 2015] to correct some supposed flaws from BCM aggregations in the case where there are only few observations in each sub-models. The predicted aggregated density is given by

$$f_{rbcm}(y) \propto \frac{\prod_{i=1}^{p} (f_i(y))^{\beta_i}}{(f_Y(y))^{-1+\sum_i \beta_i}}, \tag{35}$$

where $\beta_i = \frac{1}{2}[\log(\mathrm{V}[Y(x)]) - \log(v_i(x))]$ with $v_i(x)$ the predicted variance of the $i^{th}$ sub-model at $x$.

**Smallest prediction variance (SPV)**. For the sake of comparison, we add another aggregation

method to the benchmark: for a given prediction point $x$, the aggregation returns the prediction of the sub-model with the lowest prediction variance:

$$f_{spv}(y) = f_k(y) \qquad \text{with } k = \underset{i\in\{1,\dots,p\}}{\operatorname{argmin}}\ v_i(x). \tag{36}$$

One advantage of these aggregation methods is their very low complexity. However, these methods are missing basic good behaviour properties. On the one hand (generalised) product of expert and (robust) Bayesian committee machine can be inconsistent. This is is a direct consequence of Prop. 6 with $a(s^2,k) = b(s^2,k) = 1/s^2$ for PoE, $a(s^2,k) = b(s^2,k) = (1/2)[\log(k) - \log(s^2)][1/s^2]$ for GPoE, $a(s^2,k) = 1/s^2$, $b(s^2,k) = 1/s^2 - 1/k$ for BCM and $a(s^2,k) = (1/2)[\log(k) - \log(s^2)][1/s^2]$, $b(s^2,k) = (1/2)[\log(k) - \log(s^2)][1/s^2 - 1/k]$ for RBCM. On the other hand, SPV can be proved to be consistent but the predictions given by this method are not continuous.

The test functions are given by samples over $[0,1]$ of a centered Gaussian process $Y$ with a Matérn 5/2 kernel. The variance and length-scale parameters of the later are fixed to $\sigma^2 = 1$ and $\theta = 0.05$. The vector of observation points $X$ consists in 30 random points uniformly distributed on $[0,1]$ and we consider in this example the aggregation of 15 sub-models based on two points each. Assuming that the observations points are ordered ($x_1 \leq ... \leq x_n$), each sub-model is trained with two consecutive observations points : $\mathcal{A}_1 = \{1,2\}$, ..., $\mathcal{A}_{15} = \{29,30\}$. The variance and length-scale parameters of the sub-models are equal to the values used to generate the process samples.

First of all, we will focus on the aggregated models obtained with the different methods for a given sample path and design of experiments $X$ before looking at the distribution of various criteria when replicating the experiment. Figure 7 shows the aggregated models for the aggregation methods described above. On this example, PoE and GPoE appear respectively to be over- and under-confident in their predictions and show a mean prediction that tends too quickly to zero as the prediction point moves away from the observation points. On the other hand, the predictions from other methods seem more reliable and the best approximation is obtained with the proposed nested estimation approach.

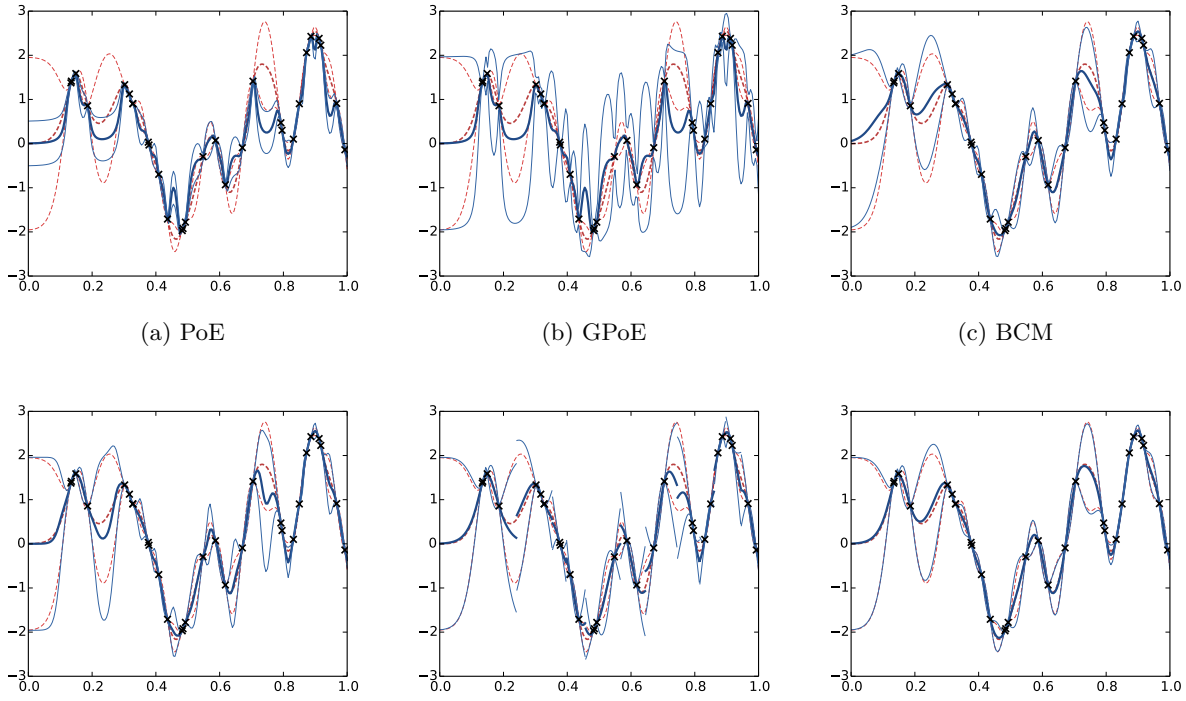
This can confirmed by replicating 50 times the experiment by sampling independently the observation points and the test function. We consider three criteria to quantify the distance between the aggregated model and the full model: the mean square error (MSE) to assess the accuracy of the aggregated mean, the mean variance error MVE for the accuracy of the predicted variance and the mean negative log probability (MNLP) [Williams and Rasmussen, 2006] to quantify the overall distribution fit. Let $m, v$ (resp. $m_{full}, v_{full}$) denote the mean and variance of the model to be tested (resp. the full model) and let $X_t$ be the vector of test points. These criteria are defined as:

$$\begin{cases} MSE(m, m_{full}, X_t) = \dfrac{1}{n_t}\displaystyle\sum_{i=1}^{n_t}(m(x_{t,i}) - m_{full}(x_{t,i}))^2\,, \\ MVE(v, v_{full}, X_t) = \dfrac{1}{n_t}\displaystyle\sum_{i=1}^{n_t}(v(x_{t,i}) - v_{full}(x_{t,i}))\,, \\ MNLP(m, v, f, X_t) = \dfrac{1}{n_t}\displaystyle\sum_{i=1}^{n_t}\left(\frac{1}{2}\log(2\pi v(x_{t,i})) + \frac{(m(x_{t,i}) - f(x_{t,i}))^2}{2v(x_{t,i})}\right). \end{cases} \tag{37}$$

Figure 8 shows the boxplots of these criteria for 50 replications of the experiments. It appears that the proposed approach gives the best approximation of the full model for the three considered criteria.

## 5.2 Application to an industrial case study

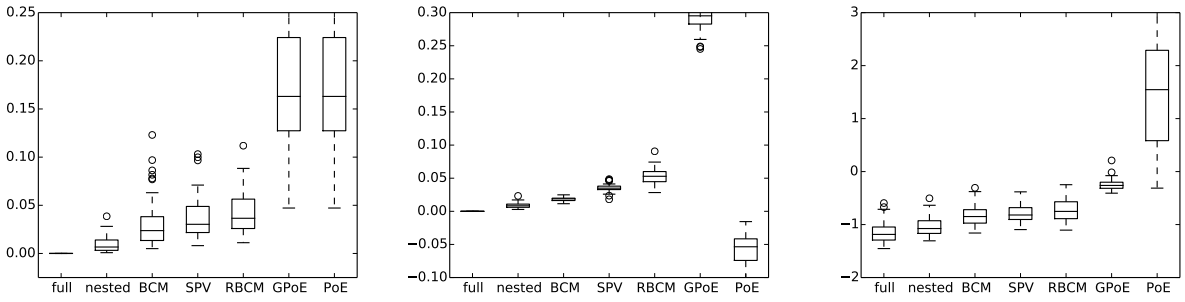
We consider in this section experimental data on the behaviour of a steel test piece subject to cycles of tension-compression. During these cycles, the evolution of the tensile strain in the test piece is monitored over time using two methods: by performing the actual physical experiment and by a numerical simulator based on a Chaboche constitutive equation [Lemaitre and Chaboche, 1994]. The quantity of interest is the misfit between these two experiments. A test piece is described by 6 scalar variables $(E, C_1, C_2, \gamma_1^0, \gamma_2^0, r)$, where $E$ is a logarithm transform of the Young's modulus,



(a) PoE (b) GPoE (c) BCM

(d) RBCM (e) SPV (f) nested GPR

Figure 7: Comparison of various aggregation methods. The solid lines corresponds to aggregated models (mean and 95% prediction intervals) and the dashed lines indicate the full model predictions (mean and 95% prediction intervals).




(a) MSE (b) MVE (c) MNLP

Figure 8: Quality assessment of the aggregated models for 50 test functions. Each test function is a sample from a Gaussian process and in each case 30 observation points are sampled uniformly on $[0, 1]$. The test points vector $X_t$ consists of 101 points regularly spaced from $x_{t,1} = 0$ to $x_{t,101} = 1$.

$C_1$, $C_2$, $\gamma_1^0$ and $\gamma_2^0$ are parameters related to the kinematic hardening and $r$ is the radius of the plastic surface at the stabilized state. The set of admissible inputs is denoted by $D \subset \mathbb{R}^6$.

Hereafter, we focus on modelling the function $f : D \to \mathbb{R}$ that returns the logarithm of the $L^2$ norm of the difference between the curve from the actual experiment and the one from the simulator.

In total, we have at our disposal a set of $10,000$ observations $[X, f(X)]$, from which we randomly extract a learning set $[X_l, f(X_l)]$ of $n = 9,000$ observations and assign the $n_t = 1,000$ remaining observations to a test set $[X_t, f(X_t)]$.

We compare the predictions of $f(X_t)$ obtained from the SPV, PoE, GPoE1, GPoE2, BCM and RBCM aggregation procedures described in Section 5.1 with our nested aggregation procedure. GPoE1 corresponds to (33) with $\beta_i = \frac{1}{2}[\log(\mathrm{V}[Y(x)]) - \log(v_i(x))]$ [Cao and Fleet, 2014] and GPoE2 corresponds to (33) with $\beta_i = 1/p$ [Deisenroth and Ng, 2015]. For all these methods, we consider an aggregation tree of height $\bar{\nu} = 2$ (once submodels have been evaluated at layer 1, they are all directly aggregated into one value at layer 2), so that $p$ Gaussian process models are directly aggregated. The $p$ subsamples form a partition of $[X_l, f(X_l)]$, which is obtained using the $k$-means clustering algorithm.

Three covariance functions have been considered for the sub-models: (tensorized) exponential, Matérn 3/2 and Matérn 5/2 (see [Williams and Rasmussen, 2006, Roustant et al., 2012] for the definition of these functions). For all studied methods, the Matérn 5/2 covariance seemed to be the most appropriate to the problem at hand since we obtained overall more accurate results. The results presented hereafter thus focus on this Matérn 5/2 covariance family. Its parameters are estimated with two different techniques depending on the aggregation method: for the methods from the literature and SPV, we follow the recommended procedure which consists in maximizing the sum of the log likelihoods over the $p$ subsamples of $[X_l, f(X_l)]$ (see [Deisenroth and Ng, 2015]). For the proposed nested aggregation, we carry out the stochastic-gradient based estimation method described in Section 4, with starting points set to the maximiser of the sum of the log likelihoods.

To assess the quality of a model with predicted mean $m$ and variance $v$, we compute three quality criteria using the test set: MSE and MNLP as per Eq. 37 which are small for a good model, and the mean normalized square error (MNSE)

$$MNSE(m, v, f, X_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \frac{(m(x_{t,i}) - f(x_{t,i}))^2}{v(x_{t,i})},$$

which should be close to 1.

The prediction results for a given learning and training test set are given in Table 1 for the aggregation of $p = 20$ sub-models and in Table 2 for $p = 90$. It can be seen that in both cases the proposed method outperforms the other aggregation methods for the MSE and MNLP quality criteria. The MSE has the same order of magnitude for the SPV and our aggregation method, where the prediction errors are small compared the empirical variance of the test outputs $f(x_{t,i})$, $i = 1, ..., n_t$, which is approximately equal to 0.81. In contrasts, the MSE can be significantly larger for all the other aggregation procedures. For the PoE, GPoE1, BCM and RBCM aggregation techniques, the values of MNSE are orders of magnitude greater than the target value one, which indicates that the aggregated models are highly overconfident. The GPoE2 aggregation technique is also overconfident when $p = 90$, where its MNSE is equal to 5.16. The SPV and our aggregation methods provide appropriate predictive variances, and our method provides the best combination of predictions and predictive variances, according to the MNLP criterion.

Tables 1 and 2 also show that aggregating $p = 20$ sub-models gives more accurate models than aggregating $p = 90$ sub-models. This suggests that it is a good practice to aggregate few sub-models based on many points instead of aggregating many sub-models based on few points. Although this would require further testing to be confirmed, it is not surprising since aggregation methods rely on some independence assumptions that are not often met in practice.

| | SPV | PoE | GPoE1 | GPoE2 | BCM | RBCM | Nested |
|---|---|---|---|---|---|---|---|
| MSE | 0.00416 | 0.0662 | 0.0033 | 0.0662 | 0.604 | 0.0625 | **0.00321** |
| MNSE | 1.27 | 20.00 | 4.55 | **1.00** | 219 | 60.8 | 0.846 |
| MNLP | −1.86 | 7.25 | −0.949 | −0.765 | 107 | 27.2 | **−1.97** |

Table 1: Prediction performances of the aggregation of $p = 20$ sub-models for the steel piece constraints cycles data set. The investigated prediction performance criteria are the mean square error (MSE) which should be minimal, mean normalized square error (MNSE) which should be close to 1 and mean negative log probability (MNLP) which should be small. Bold figures indicate each line's best performing aggregation method.

| | SPV | PoE | GPoE1 | GPoE2 | BCM | RBCM | Nested |
|---|---|---|---|---|---|---|---|
| MSE | 0.00556 | 0.811 | 0.0244 | 0.811 | 1.84 | 0.121 | **0.00418** |
| MNSE | 1.20 | 465 | 34.2 | 5.16 | 980 | 148 | **0.84700** |
| MNLP | −1.55 | 230 | 14.1 | 2.13 | 487 | 71 | **−1.7** |

Table 2: Prediction performances of the aggregation of $p = 90$ sub-models for the steel piece constraints cycles data set. All other settings are the same as in Table 1.

Tables 3 and 4 show the values of the quality criteria when the subsamples used for the $p = 20$ or $p = 90$ sub-models are randomly generated into the learning set. They can thus be compared to Tables 1 and 2 to study the influence of the choice of the support points of the sub-models : the criteria values are overall better in Tables 1 and 2 so using $k$-means is beneficial for the aggregation procedures. In addition, our proposed aggregation technique becomes better in comparison to the other methods, and specifically to SPV, when the subsamples are randomly generated.

All previous results have been obtained for a given random choice of the learning and test sets. We now replicate the procedure 20 times, with the same settings as in Tables 1 ($p = 20$; subsamples obtained from the $k$-means algorithm; Matérn 5/2 covariance function) and 4 ($p = 90$; subsamples randomly selected; Matérn 5/2 covariance function), but with different learning and test sets for each replication. The covariance parameters are reestimated for each learning set, by minimizing the sum of log likelihoods for the SPV, PoE, GPoE1, GPoE2, BCM and RBCM aggregation techniques, and with the proposed leave one out estimation procedure for our nested aggregation method. The box plots of the corresponding 20 mean square errors and mean negative log probability are reported in Figures 9 and 10. These replications confirm the results obtained previously on single instances of the learning and test set: the proposed nested aggregation and covariance parameter estimation jointly give better prediction both for the predicted mean and variance than current existing aggregation techniques.

Of course, the improvement brought by our proposed aggregation scheme comes with a higher computational cost: the proposed estimation procedure takes a few hours on a personal computer, against a few tens of minutes for the minimization of the sum of the log likelihoods. Similarly, performing 1000 predictions takes around 30 seconds with our proposed optimal aggregation, against around 1 second for the other simpler aggregation procedures. Nevertheless, we believe that the increased accuracy and robustness of the method we propose is worth the additional computational burden in many situations.

| | SPV | PoE | GPoE1 | GPoE2 | BCM | RBCM | Nested |
|---|---|---|---|---|---|---|---|
| MSE | 0.0086 | 0.00763 | 0.00704 | 0.00763 | 0.338 | 0.274 | **0.00539** |
| MNSE | 1.21 | 9.38 | 16.6 | 0.469 | 178 | 268 | **0.864** |
| MNLP | −1.25 | 1.75 | 5.03 | −1.21 | 86.2 | 130 | **−1.5** |

Table 3: Same settings as in Table 1 but when the subsamples are randomly selected.

| | SPV | PoE | GPoE1 | GPoE2 | BCM | RBCM | Nested |
|---|---|---|---|---|---|---|---|
| MSE | 0.0182 | 0.0293 | 0.0246 | 0.0293 | 0.977 | 0.686 | **0.00575** |
| MNSE | 1.29 | 42.5 | 57.2 | 0.473 | 852 | 988 | **0.867** |
| MNLP | −0.804 | 18.3 | 25.3 | −0.517 | 423 | 491 | **−1.37** |

Table 4: Same settings as in Table 1 but with $p = 90$ submodels and where the subsamples are randomly selected.

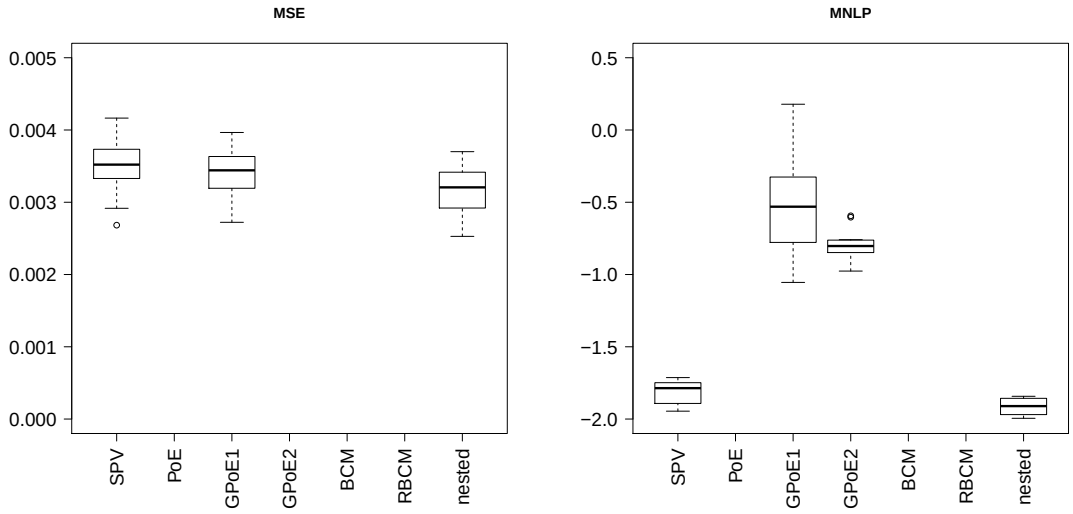



Figure 9: Box plots of 20 values of the mean square error (MSE) prediction criterion and of the logarithm of the mean negative log probability (MNLP) prediction criterion where the learning and test sets are randomly generated. The settings are as in Table 1 ($p = 20$ subsamples obtained from the $k$-means algorithm; Matérn 5/2 covariance function). The covariance parameters are estimated by minimizing the sum of log likelihoods for the SPV, PoE, GPoE1, GPoE2, BCM and RBCM aggregation techniques, and with our proposed leave one out estimation procedure for the nested aggregation procedure. The box plots that are not represented correspond to large MSE or MLNP values. More specifically, the averages (standard deviations) of the 20 MSE values for PoE, GPoE2, BCM and RBCM are respectively 0.057 (0.0107), 0.057 (0.0107), 0.43 (0.095) and 0.047 (0.0099). The averages (standard deviations) of the 20 MNLP values for PoE, BCM and RBCM are respectively 7.04 (1.37), 90.1 (18.5) and 23.2 (4.09).

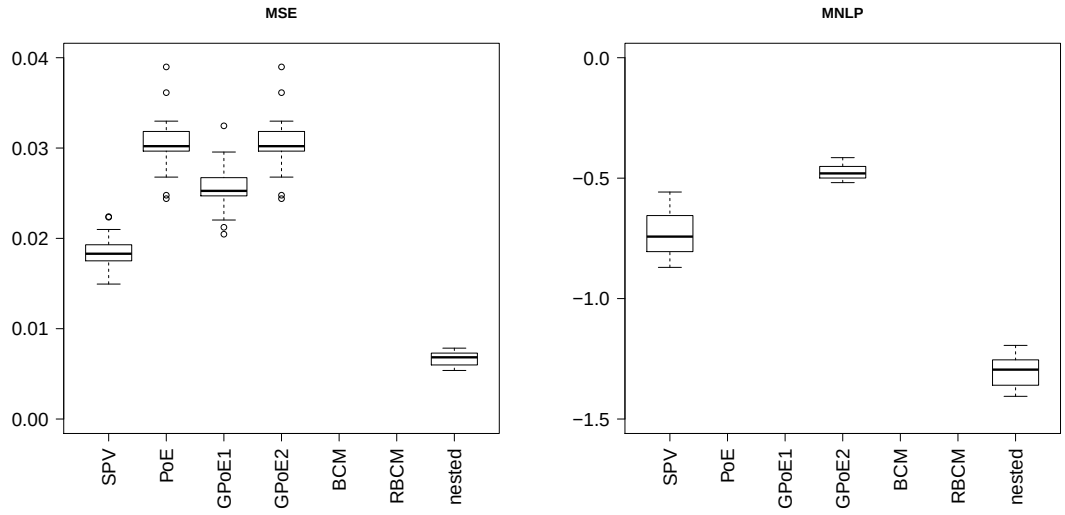



Figure 10: Same settings as in Figure 9 but with $p = 90$ and where the subsamples are randomly selected. The box plots that are not represented correspond to large MSE or MLNP values. More specifically, the averages (standard deviations) of the 20 MSE values for BCM and RBCM are respectively 0.97 (0.11) and 0.68 (0.071). The averages (standard deviations) of the 20 MNLP values for PoE, GPoE1, BCM and RBCM are respectively 20.6 (2.22), 28.9 (3.29), 441.9 (29.0) and 509 (23.5).

# 6 Conclusion

We have proposed a new method for aggregating sub-models based on subsets of observation points, with a particular emphasis on Kriging sub-models. Our method can be seen as an optimal linear weighting of submodels, where the obtained weights are taking into account all pairwise covariances between the submodels, thus avoiding some usual independence assumptions. Compared to current existing aggregation techniques, we find several benefits of our approach.

First, our aggregation procedure has some theoretical benefits. It can be seen as an optimal method based on a slightly different process, which can be simulated. In the Gaussian case, it yields a full conditional Gaussian process distribution, which allows computing conditional covariances and simulating Gaussian process conditional sample paths. Furthermore, our nested aggregation method is proven to provide consistent predictors whereas, as demonstrated, some other classical aggregation techniques can yield inconsistent predictors.

Second, a dedicated covariance parameter estimation procedure is provided, based on a gradient descent minimization of leave-one-out cross validation errors, where the predictions are performed using the proposed nested aggregation. Some code for computing both prediction and covariance parameter estimation is publicly available.

At last, numerical results are encouraging. In both simulated data and industrial application, our method is shown to outperform state of the art aggregation techniques. This improvement comes with an increased computational cost compared to more basic aggregation methods, but the proposed nested remains typically applicable in the domain $n = 10,000 - 100,000$ observation points, while exact Kriging inference becomes intractable around $n = 10,000$ observation points.

We would like to mention two avenues for future research. First, we show that the aggregation method we propose can be applied recursively, yielding a nested aggregation technique with smaller computational cost. It would be interesting to quantify the practical gain one could obtain on real data sets from this recursive aggregation. Second, we find that the stochastic gradient algorithm we propose could be further investigated. In particular, theoretical properties could be derived, the practical implementation could be improved, and the principle could be extended to other criteria for covariance parameter estimation.

## Acknowledgements

The authors would like to warmly thank Dr. Géraud Blatman from EDF R&D for providing us the industrial test case.

## References

[Anitescu et al., 2016] Anitescu, M., Chen, J., and Stein, M. L. (2016). An inversion-free estimating equations approach for Gaussian process models. *Journal of Computational and Graphical Statistics*, just-accepted:1–42.

[Bachoc, 2013] Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model mispecification. *Computational Statistics and Data Analysis*, 66:55–69.

[Bai et al., 2012] Bai, Y., Song, P. X. K., and Raghunathan, T. E. (2012). Joint composite estimating functions in spatiotemporal models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(5):799–824.

[Bevilacqua and Gaetan, 2015] Bevilacqua, M. and Gaetan, C. (2015). Comparing composite likelihood methods based on pairs for spatial Gaussian random fields. *Statistics and Computing*, 25(5):877–892.

[Bhatnagar et al., 2013] Bhatnagar, S., Prasad, H., and Prashanth, L. (2013). *Stochastic recursive algorithms for optimization*, volume 434. New York: Springer.

[Cao and Fleet, 2014] Cao, Y. and Fleet, D. J. (2014). Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions. *ArXiv e-prints*.

[Chevalier and Ginsbourger, 2013] Chevalier, C. and Ginsbourger, D. (2013). Fast computation of the multi-points expected improvement with applications in batch selection. In *Learning and Intelligent Optimization*, pages 59–69. Springer.

[Datta et al., 2016] Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, accepted.

[Deisenroth and Ng, 2015] Deisenroth, M. P. and Ng, J. W. (2015). Distributed Gaussian processes. *Proceedings of the 32nd International Conference on Machine Learning, Lille, France. JMLR: W&CP volume 37*.

[Furrer et al., 2006] Furrer, R., Genton, M. G., and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*.

[Genest and Zidek, 1986] Genest, C. and Zidek, J. V. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, pages 114–135.

[Gneiting, 2002] Gneiting, T. (2002). Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83(2):493–508.

[Golub and Van Loan, 2012] Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

[Hensman et al., 2013] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *Uncertainty in Artificial Intelligence*, pages 282–290.

[Hinton, 2002] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

[Kaufman et al., 2011] Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011). Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *The Annals of Applied Statistics*, 5(4):2470–2492.

[Kaufman et al., 2008] Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555.

[Lemaitre and Chaboche, 1994] Lemaitre, J. and Chaboche, J.-L. (1994). *Mechanics of solid materials*. Cambridge university press.

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

[Maurya, 2016] Maurya, A. (2016). A well-conditioned and sparse estimation of covariance and inverse covariance matrices using a joint penalty. *Preprint*.

[Nickson et al., 2015] Nickson, T., Gunter, T., Lloyd, C., Osborne, M. A., and Roberts, S. (2015). Blitzkriging: Kronecker-structured stochastic Gaussian processes. *arXiv preprint arXiv:1510.07965*.

[Quinonero-Candela and Rasmussen, 2005] Quinonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959.

[Ranjan and Gneiting, 2010] Ranjan, R. and Gneiting, T. (2010). Combining probability forecasts. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):71–91.

[Roustant et al., 2012] 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. *Journal of Statistical Software*, 51(1).

[Rue and Held, 2005] Rue, H. and Held, L. (2005). *Gaussian Markov random fields, Theory and applications*. Chapman & Hall.

[Samo and Roberts, 2016] Samo, Y.-L. K. and Roberts, S. J. (2016). String and membrane gaussian processes. *Journal of Machine Learning Research*, 17(131):1–87.

[Sang and Huang, 2012] Sang, H. and Huang, J. Z. (2012). A full scale approximation of covariance functions for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):111–132.

[Santner et al., 2013] Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

[Satopää et al., 2015] Satopää, V. A., Pemantle, R., and Ungar, L. H. (2015). Modeling probability forecasts via information diversity. *Journal of the American Statistical Association*, just-accepted.

[Scott et al., 2016] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88.

[Stein, 2008] Stein, M. L. (2008). A modeling approach for large spatial datasets. *Journal of the Korean Statistical Society*, 37(1):3–10.

[Stein, 2012] Stein, M. L. (2012). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

[Stein, 2013] Stein, M. L. (2013). Statistical properties of covariance tapers. *Journal of Computational and Graphical Statistics*, 22(4):866–885.

[Stein, 2014] Stein, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19.

[Stein et al., 2013] Stein, M. L., Chen, J., and Anitescu, M. (2013). Stochastic approximation of score functions for Gaussian processes. *The Annals of Applied Statistics*, 7(2):1162–1191.

[Tresp, 2000] Tresp, V. (2000). A bayesian committee machine. *Neural Computation*, 12(11):2719–2741.

[Tzeng et al., 2005] Tzeng, S., Huang, H.-C., and Cressie, N. (2005). A fast, optimal spatial-prediction method for massive datasets. *Journal of the American Statistical Association*, 100(472):1343–1357.

[van Stein et al., 2015] van Stein, B., Wang, H., Kowalczyk, W., Bäck, T., and Emmerich, M. (2015). Optimally weighted cluster kriging for big data regression. In *International Symposium on Intelligent Data Analysis*, pages 310–321. Springer.

[Vazquez and Bect, 2010] Vazquez, E. and Bect, J. (2010). Pointwise consistency of the kriging predictor with known mean and covariance functions. In *mODa 9 (Model-Oriented Data Analysis and Optimum Design) Springer*.

[Vecchia, 1988] Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):297–312.

[Wahba, 1990] Wahba, G. (1990). *Spline models for observational data*, volume 59. SIAM.

[Wei et al., 2015] Wei, H., Du, Y., Liang, F., Zhou, C., Liu, Z., Yi, J., Xu, K., and Wu, D. (2015). A k-d tree-based algorithm to parallelize kriging interpolation of big spatial data. *GIScience & Remote Sensing*, 52(1):40–57.

[Williams and Rasmussen, 2006] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

[Winkler, 1968] Winkler, R. L. (1968). The consensus of subjective probability distributions. *Management Science*, 15(2):B–61.

[Winkler, 1981] Winkler, R. L. (1981). Combining probability distributions from dependent information sources. *Management Science*, 27(4):479–488.

[Xue et al., 2012] Xue, L., Ma, S., and Zou, H. (2012). Positive-definite l1-penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491.

# A Proof of Proposition 5

Because $D$ is compact we have $\lim_{n\to\infty} \sup_{x\in D} \min_{i=1,...,n} ||x_{ni} - x|| = 0$. Indeed, if this does not hold, there exists $\epsilon > 0$ and a subsequence $\phi(n)$ so that $\sup_{x\in D} \min_{i=1,...,\phi(n)} ||x_{\phi(n)i} - x|| \geq 2\epsilon$. Hence, there exists a sequence, $x_{\phi(n)} \in D$ so that $\min_{i=1,...,\phi(n)} ||x_{\phi(n)i} - x_{\phi(n)}|| \geq \epsilon$. Since $D$ is compact, up to extracting a further subsequence, we can also assume that $x_{\phi(n)} \to_{n\to\infty} x_{lim}$ with $x_{lim} \in D$. This implies that for all $n$ large enough, $\min_{i=1,...,\phi(n)} ||x_{\phi(n)i} - x_{lim}|| \geq \epsilon/2$, which is in contradiction with the assumptions of the proposition.

Hence there exists a sequence of positive numbers $\delta_n$ so that $\delta_n \to_{n\to\infty} 0$ and so that for all $x \in D$ there exists a sequence of indices $i_n(x)$ so that $i_n(x) \in \{1, ..., n\}$ and $||x - x_{ni_n(x)}|| \leq \delta_n$. There also exists a sequence of indices $j_n(x)$ so that $x_{ni_n(x)}$ is a component of $X_{j_n(x)}$. With these notations we have, since $M_1(x)$,..., $M_{p_n}(x)$, $M_{\mathcal{A}_n}(x)$ are linear combinations with minimal square prediction errors,

$$\begin{aligned} \sup_{x\in D} \mathrm{E}\left[(Y(x) - M_{\mathcal{A}_n}(x))^2\right] &\leq \sup_{x\in D} \mathrm{E}\left[\left(Y(x) - M_{j_n(x)}(x)\right)^2\right] \\ &\leq \sup_{x\in D} \mathrm{E}\left[\left(Y(x) - Y(x_{ni_n(x)})\right)^2\right]. \end{aligned} \tag{38}$$

In the rest of the proof we essentially show that, for a dense triangular array of observation points, the Kriging predictor which predicts $Y(x)$ based only on the nearest neighbour of $x$ among the observation points has a mean square prediction error which goes to zero uniformly in $x$ when $k$ is continuous. We believe that this fact is somehow known, but we have not been able to find a precise result in the literature.

We have from (38),

$$\begin{aligned} &\sup_{x\in D} \mathrm{E}\left[(Y(x) - M_{\mathcal{A}_n}(x))^2\right] \\ &\leq \sup_{x\in D} \left[\mathbf{1}\{k(x_{ni_n(x)}, x_{ni_n(x)}) = 0\}k(x,x) + \mathbf{1}\{k(x_{ni_n(x)}, x_{ni_n(x)}) > 0\}\left(k(x,x) - \frac{k(x, x_{ni_n(x)})^2}{k(x_{ni_n(x)}, x_{ni_n(x)})}\right)\right] \\ &\leq \sup_{\substack{x,t\in D;\\ ||x-t||\leq\delta_n}} \left[\mathbf{1}\{k(t,t) = 0\}k(x,x) + \mathbf{1}\{k(t,t) > 0\}\left(k(x,x) - \frac{k(x,t)^2}{k(t,t)}\right)\right] \\ &= \sup_{\substack{x,t\in D;\\ ||x-t||\leq\delta_n}} F(x,t), \end{aligned}$$

say. Assume now that the above supremum does not go to zero as $n \to \infty$. Then there exists $\epsilon > 0$ and two sub-sequences $x_{\phi(n)}$ and $t_{\phi(n)}$ with values in $D$ so that $x_{\phi(n)} \to_{n\to\infty} x_{lim}$ and $t_{\phi(n)} \to_{n\to\infty} x_{lim}$, with $x_{lim} \in D$ and so that $F(x_{\phi(n)}, t_{\phi(n)}) \geq \epsilon$. Then if $k(x_{lim}, x_{lim}) = 0$ then $F(x_{\phi(n)}, t_{\phi(n)}) \leq k(x_{\phi(n)}, x_{\phi(n)}) \to_{n\to\infty} 0$. If $k(x_{lim}, x_{lim}) > 0$ then for $n$ large enough

$$F(x_{\phi(n)}, t_{\phi(n)}) = k(x_{\phi(n)}, x_{\phi(n)}) - \frac{k(x_{\phi(n)}, t_{\phi(n)})^2}{k(t_{\phi(n)}, t_{\phi(n)})}$$

which goes to zero as $n \to \infty$ since $k$ is continuous. Hence we have a contradiction, which completes the proof.

# B Proof of Proposition 6

Because of the assumptions on $k$, $Y$ has the no-empty-ball property (Definition 1 and Proposition 1 in [Vazquez and Bect, 2010]). Hence for $\delta > 0$, letting

$$V(\delta) = \inf_{n\in\mathbb{N}} \inf_{\substack{x_1,\ldots,x_n\in D;\\ \forall i=1,\ldots,n, ||x_i-x_0||\geq\delta}} \mathrm{V}\left[Y(x_0)|Y(x_1),...,Y(x_n)\right],$$

we have that $V(\delta) > 0$.

Consider a sequence $\delta_n$ of non-negative numbers so that $\delta_n \to_{n\to\infty} 0$, and which will be specified below. There exists a sequence $(u_n)_{n\in\mathbb{N}} \in D^{\mathbb{N}}$, composed of two-by-two distinct elements, so that $\lim_{n\to\infty} \sup_{x\in D} \min_{i=1,\ldots,n} ||u_i - x|| = 0$, and so that for all $n$, $\inf_{1\leq i\leq n} ||u_i - x_0|| \geq \delta_n$.

Let $x_0$ and $\bar{x}$ be so that $k(x_0, \bar{x}) > 0$ and $D$ contains two open balls with strictly positive radii and centers $x_0$ and $\bar{x}$ (the existence is assumed in the proposition). We can find $0 < r_1 < ||x_0 - \bar{x}||/4$ so that $B(\bar{x}, r_1) \subset D$. Then, by continuity of $k$, we can find $\epsilon_2 > 0$, $0 < r \leq r_1$ and $0 < \delta_1 \leq r_1$ so that $B(\bar{x}, r) \subset D$ and for all $x \in B(\bar{x}, r)$, $||x - x_0|| \geq \delta_1$ and

$$k(x_0, x_0) - \frac{k(x, x_0)^2}{k(x, x)} \leq k(x_0, x_0) - \epsilon_2.$$

Consider then the sequence $(w_n)_{n\in\mathbb{N}} \in D^{\mathbb{N}}$ so that for all $n$, $w_n = \bar{x} - (r/(1+n))e_1$ with $e_1 = (1, 0, ..., 0)$. We can assume furthermore that $\{u_n\}_{n\in\mathbb{N}}$ and $\{w_n\}_{n\in\mathbb{N}}$ are disjoint.

Let us now consider two sequences of integers $p_n$ and $k_n$ with $k_n \to \infty$ and $p_n \to \infty$ to be specified later. Let $C_n$ be the largest natural number $m$ satisfying $m(p_n - 1) < n$. Let $X = (X_1, ..., X_{p_n})$ be defined by, for $i = 1, ..., k_n$, $X_i = (u_j)_{j=(i-1)C_n+1,\ldots,iC_n}$; for $i = k_n + 1, ..., p_n - 1$, $X_i = (w_j)_{j=(i-k_n-1)C_n+1,\ldots,(i-k_n)C_n}$; and $X_{p_n} = (w_j)_{j=(p_n-k_n-1)C_n+1,\ldots,n-k_nC_n}$. With this construction, note that $X_{p_n}$ is non-empty. Furthermore, the sequence of vectors $X = (X_1, ..., X_{p_n})$, indexed by $n \in \mathbb{N}$, defines a triangular array of observation points satisfying the conditions of the proposition.

Observing that $\inf_{i\in\mathbb{N}} ||w_i - x_0|| \geq \delta_1$ and letting $\epsilon_1 = V(\delta_1) > 0$, we have for all $n \in \mathbb{N}$ and for all $k = k_n + 1, ..., p_n$, since then $X_k$ is non-empty and only contains elements $w_i \in B(\bar{x}, r)$,

$$\epsilon_1 \leq v_k(x_0) \leq k(x_0, x_0) - \epsilon_2. \tag{39}$$

From (39), and since $\hat{k}$ is a positive function and $x_0$ is not a component of $X$, we have $v_k(x_0) > 0$ for all $k$, and $v_{p_n}(x_0) < k(x_0, x_0)$. Hence, $\bar{M}_{\mathcal{A}_n}$ is well-defined, at least for $n$ large enough.

For two random variables $A$ and $B$, we let $||A - B|| = (\mathrm{E}\left[(A-B)^2\right])^{1/2}$. Let

$$R = \left|\left|\sum_{k=1}^{k_n} \alpha_{k,n}(v_1(x_0), ..., v_{p_n}(x_0), v_{prior}(x_0))M_k(x_0)\right|\right|.$$

Then, from the triangular inequality, and since, from the law of total variance, $||M_k(x_0)|| \leq ||Y(x_0)|| = v_{prior}(x_0)$ we have

$$\begin{aligned} R &\leq \frac{\sum_{k=1}^{k_n} a(v_k(x_0), v_{prior}(x_0))\sqrt{v_{prior}(x_0)}}{\sum_{l=1}^{p_n} b(v_l(x_0), v_{prior}(x_0))} \\ &\leq \frac{k_n \sup_{s^2\geq V(\delta_n)} a(s^2, v_{prior}(x_0))\sqrt{v_{prior}(x_0)}}{(p_n - k_n)\inf_{\epsilon_1\leq s^2\leq v_{prior}(x_0)-\epsilon_2} b(s^2, v_{prior}(x_0))}, \end{aligned}$$

where the last inequality is obtained from (39) and the definition of $\delta_n$ and $V(\delta)$.

Let now for $\delta > 0$, $s(\delta) = \sup_{s^2\geq V(\delta)} a(s^2, v_{prior}(x_0))$. Since $a$ is continuous on $D$ and since $V(\delta) > 0$, we have that $s(\delta)$ is finite. Hence, we can choose a sequence $\delta_n$ of positive numbers so that $\delta_n \to_{n\to\infty} 0$ and $s(\delta_n) \leq \sqrt{n}$ (for instance, let $\delta_n = \inf\{\delta \geq n^{-1/2}; V(\delta) \leq n^{1/2}\}$). Then, we can choose $p_n = n^{4/5}$ and $k_n = n^{1/5}$. Then, for $n$ large enough

$$\frac{k_n}{p_n - k_n} s(\delta_n) \leq 2n^{-3/5}\sqrt{n} \to_{n\to\infty} 0.$$

Hence, since

$$\frac{\sqrt{v_{prior}(x_0)}}{\inf_{\epsilon_1\leq s^2\leq v_{prior}(x_0)-\epsilon_2} b(s^2, v_{prior}(x_0))}$$

is a finite constant, as $b$ is positive and continuous on $\mathring{D}$, we have that $R \to_{n\to\infty} 0$. As a consequence, we have from the triangular inequality

$$\left| ||Y(x_0) - \bar{M}_{\mathcal{A}_n}(x_0)|| - ||Y(x_0) - \sum_{k=k_n+1}^{p_n} \alpha_{k,n}(v_1(x_0), ..., v_{p_n}(x_0), v_{prior}(x_0))M_k(x_0)|| \right|$$
$$\leq || \sum_{k=k_n+1}^{p_n} \alpha_{k,n}(v_1(x_0), ..., v_{p_n}(x_0), v_{prior}(x_0))M_k(x_0) - \bar{M}_{\mathcal{A}_n}(x_0)||$$
$$= R$$
$$\to_{n\to\infty} 0.$$

Hence

$$\liminf_{n\to\infty} ||Y(x_0) - \bar{M}_{\mathcal{A}_n}(x_0)|| = \liminf_{n\to\infty} \left|\left| Y(x_0) - \sum_{k=k_n+1}^{p_n} \alpha_{k,n}(v_1(x_0), ..., v_{p_n}(x_0), v_{prior}(x_0))M_k(x_0) \right|\right|.$$

Hence, since $X_{k_n+1}, ..., X_{p_n}$ are composed only of elements of $\{w_i\}_{i\in\mathbb{N}}$,

$$\liminf_{n\to\infty} ||Y(x_0) - \bar{M}_{\mathcal{A}_n}(x_0)|| \geq V(\delta_1) > 0.$$

# C Proof of Proposition 11

*Complexities*: under chosen assumption on $\alpha$ and $\beta$ coefficients, for a regular tree and in the case of simple Kriging sub-models, $\mathcal{C}_\alpha = \sum_{\nu=1}^{\bar{\nu}} \sum_{i=1}^{n_\nu} \alpha c_\nu^3 = \alpha \sum_{\nu=1}^{\bar{\nu}} c_\nu^3 n_\nu$ and $\mathcal{C}_\beta = \sum_{\nu=1}^{\bar{\nu}} \sum_{i=2}^{n_\nu} \sum_{j=1}^{i-1} \beta c_\nu^2 = \frac{\beta}{2} \sum_{\nu=1}^{\bar{\nu}} n_\nu(n_\nu - 1)c_\nu^2$. Notice that the sum starts from $\nu = 1$ in order to include sub-models calculation. *Equilibrated trees complexities*: In a constant child number setting, when $c_\nu = c$ for all $\nu$, the tree structure ensures that $n_\nu = n/c^\nu$, thus as $c = n^{1/\bar{\nu}}$, we get when $n \to +\infty$, $\mathcal{C}_\alpha \sim \alpha n^{1+\frac{2}{\bar{\nu}}}$ and $\mathcal{C}_\beta \sim \frac{\beta}{2} n^2$. The result for equilibrated two-layer tree where $\bar{\nu} = 2$ directly derives from this one, and in this case $\mathcal{C}_\alpha \sim \alpha n^2$ and $\mathcal{C}_\beta \sim \frac{\beta}{2} n^2$ (it derives also from the expressions of $\mathcal{C}_\alpha$, $\mathcal{C}_\beta$, when $c_1 = c_2 = \sqrt{n}$, $n_1 = \sqrt{n}$, $n_2 = 1$). *Optimal tree complexities*: One easily shows that under the chosen assumptions $\mathcal{C}_\beta \sim \frac{\beta}{2} n^2$. Thus, it is indeed not possible to reduce the whole complexity to lower orders than $O(n^2)$. However, one can choose the tree structure in order to reduce the complexity $\mathcal{C}_\alpha$. For a regular tree, $n_\nu = n/(c_1 \cdots c_\nu)$ so that $\frac{\partial}{\partial c_k} n_\nu = -\mathbf{1}_{\{\nu \geq k\}} n_\nu / c_k$. Using a Lagrange multiplier $\ell$, one defines $\xi(k) = c_k \frac{\partial}{\partial c_k} \left(\mathcal{C}_\alpha - \ell(c_1 \cdots c_{\bar{\nu}} - n)\right) = 3\alpha c_k^3 n_k - \alpha \sum_{\nu=k}^{\bar{\nu}} c_\nu^3 n_\nu - \ell c_1 \cdots c_{\bar{\nu}}$. The tree structure that minimizes $\mathcal{C}_\alpha$ is such that for all $k < \bar{\nu}$, $\xi(k) = \xi(k+1) = 0$. Using $c_{k+1} n_{k+1} = n_k$, one gets $3c_{k+1}^2 = 2c_k^3$ for all $k < \bar{\nu}$, and setting $c_1 \cdots c_{\bar{\nu}} = n$, $c_\nu = \delta \left(\delta^{-\bar{\nu}} n\right)^{\frac{\delta^\nu - 1}{2(\delta^{\bar{\nu}} - 1)}}$, $\nu = 1, \ldots, \bar{\nu}$, with $\delta = \frac{3}{2}$. Setting $\gamma = \frac{27}{4} \delta^{-\frac{\bar{\nu}}{\delta^{\bar{\nu}} - 1}} \left(1 - \delta^{-\bar{\nu}}\right)$. After some direct calculations this tree structure corresponds to complexities, $\mathcal{C}_\alpha = \gamma \alpha n^{1+\frac{1}{\delta^{\bar{\nu}} - 1}}$ and $\mathcal{C}_\beta \sim \frac{\beta}{2} n^2$. In a two-layers setting one gets $c_1 = \left(\frac{3}{2}\right)^{1/5} n^{2/5}$ and $c_2 = \left(\frac{3}{2}\right)^{-1/5} n^{3/5}$, which leads to $\mathcal{C}_\alpha = \gamma \alpha n^{9/5}$ and $\mathcal{C}_\beta = \frac{\beta}{2} n^2 - \frac{\beta}{2} \left(\frac{3}{2}\right)^{\frac{1}{5}} n^{\frac{7}{5}}$, where $\gamma = (\frac{2}{3})^{-2/5} + (\frac{2}{3})^{3/5} \simeq 1.96$ (eventually notice that even for values of $n$ of order $10^5$, terms of order like $n^{9/5}$ are not necessarily negligible compared to those of order $n^2$, and that $\mathcal{C}_\beta$ is slightly affected by the choice of the tree structure, but the global complexity benefits from the optimization of $\mathcal{C}_\alpha$).
*Storage footprint*: First, covariances can be stored in triangular matrices. So temporary objects $M$, $k$ and $K$ in Algorithm 1 require the storage of $c_{\max}(c_{\max} + 5)/2$ real values. For a given step $\nu$, $\nu \geq 2$, building all vectors $\alpha_i$ requires the storage of $\sum_{i=1}^{n_\nu} c_i^\nu = n_{\nu-1}$ values. At last, for a given step $\nu$, we simultaneously need objects $M_{\nu-1}, K_{\nu-1}, M_\nu, K_\nu$, which require the storage of $n_{\nu-1}(n_{\nu-1} + 3)/2 + n_\nu(n_\nu + 3)/2$ real values. In a regular tree, as $n_\nu$ is decreasing in $\nu$, the storage footprint is $\mathcal{S} = (c_{\max}(c_{\max} + 5) + n_1(n_1 + 5) + n_2(n_2 + 3))/2$. Hence the equivalents for $\mathcal{S}$ for the different tree structures, $\mathcal{S} \sim n$ for the two-layer equilibrated tree, $\mathcal{S} \sim \frac{1}{2} n^{2-2/\bar{\nu}}$ for the $\bar{\nu}$-layer, $\bar{\nu} > 2$, and the indicated result for the optimal tree. Simple orders are given in the proposition, which avoids separating the case $\bar{\nu} = 2$ and a cumbersome constant for the optimal tree.