Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

STOCHASTIC GAUSSIAN PROCESS MODEL AVERAGING FOR HIGH-DIMENSIONAL INPUTS

Maxime Xuereb
Szu Hui Ng

Department of Industrial Systems Engineering
and Management
National University of Singapore
1 Engineering Drive 2
Singapore, 117576, SINGAPORE

Giulia Pedrielli

School of Computing, Informatics, and Decision
Systems Engineering
Arizona State University
699 S Mill Avenue
Tempe, AZ 85281, USA

ABSTRACT

Many statistical learning methodologies exhibit loss of efficiency and accuracy when applied to large, high-dimensional data-sets. Such loss is exacerbated by noisy data. In this paper, we focus on Gaussian Processes (GPs), a family of non-parametric approaches used in machine learning and Bayesian Optimization. In fact, GPs show difficulty scaling with the input data size and dimensionality. This paper presents, for the first time, the Stochastic GP Model Averaging (SGPMA) algorithm, to tackle both challenges. SGPMA uses a Bayesian approach to weight several predictors, each trained with an independent subset of the initial data-set (solving the large data-sets issue), and defined in a low-dimensional embedding of the original space (solving the high dimensionality). We conduct several experiments with different input size and dimensionality. The results show that our methodology is superior to naive averaging and that the embedding choice is critical to manage the computational cost / prediction accuracy trade-off.

1 INTRODUCTION & BACKGROUND

Large data sets with increasingly high dimension have become vastly common due to increased sensing, storage capacity, as well as increased intelligence embedded within modern devices (e.g. robots, self driven cars, power grids). In such a scenario, traditional statistical models are difficult to apply due to the scalability challenges. In this paper, we focus on Gaussian Processes, a particularly common modeling approach within the machine learning community (Williams and Rasmussen 2006). Gaussian Processes (GPs) are known to have issues in scaling to data set size and dimensionality, mainly due to the prediction complexity (scaling cubically with the data set size) and the model hyper-parameter optimization, with an associated complexity that increases exponentially with the dimensionality of the data set (Santner et al. 2003). These challenges have recently received an important attention within the statistical learning theory community.

In fact, scaling GP models is a very important and challenging problem. As highlighted in the literature, while deriving a low-dimensional embedding allows to reduce the computational burden associated with the model estimation (Xuereb et al. 2019), the use of a unique low-embedding leads to important losses in prediction accuracy compared with the original model (i.e., the model defined in the original space).

Approaches with a single model Several approaches in the literature focus on a single model that is designed to solve the dimensionality and/or large data size challenges. Examples of single model approaches aimed at tackling the dimensionality issue can be found in (Bouhlel and Martins 2019; Reich et al. 2011). In (Bouhlel and Martins 2019), the authors use Partial Least Squares, thus projecting the original space into a lower dimensional "target" space where the model estimation is actually performed. The question of choosing a sufficient dimension reduction has been addressed in (Reich et al. 2011), where the authors minimize the number of components in a linear predictor.

A number of contributions have focused on data size reduction (also referred to as sub-sampling), rather than dimensionality reduction (Gardner et al. 2018; Lu et al. 2019; Hayashi et al. 2019). In (Gardner et al. 2018), the authors approximate a matrix vector multiple of the covariance matrix by decomposing each Kernel with a Lanczos decomposition (Lanczos 1950). Similarly, in (Lu et al. 2019) the authors use a low-rank approximation of the covariance matrix based on sub-sample selection by means Nyström centers (Rudi et al. 2015). A review on the use of sub-sampling for increasing the computational efficiency of Gaussian Processes is provided in (Hayashi et al. 2019). Therein, experimental results show that Nyström centers-based approaches are out-performed by random sub-sampling (where sub-samples are selected at random), both in terms of accuracy as well as required computational effort. Moreover, random sub-sampling can be easily generalized.

Some GP models are specifically developed to improve their scalability without losing too much prediction quality: they are categorized by Liu et al. (2020) as *scalable GPs*. A comprehensive review on GP models scalability was written by Liu et al. (2020). In a comparison between scalable and exact GP models on large data sets, Wang et al. (2019) claim that exact GPs have better performance, in terms of prediction accuracy, compared to approximate GPs. However, this comparison was done on a specific data set and can hardly be generalized.

Approaches with Multiple Models A way to overcome the performance issues caused by the use of a single model is to generate multiple predictors, each with an individual support (possibly with reduced dimensionality), and training set. The output predictor can then be built as the average of the predictors resulting from the individual models. A key question is how to weight the different predictors, since arguably each will have a different relative accuracy across the different sub-regions of the original space. Hence, one would like the ability to weight each predictor differently across regions. In the literature, Bayesian and frequentist model averaging have both witnessed an important development.

For the Bayesian Model Averaging (BMA), in (Raftery et al. 1997; Hoeting et al. 1999), the authors show how the posterior distribution of a quantity of interest, given a set of data, can be expressed as the weighted average of the posterior distributions under each considered model, where the weights are the posterior model probabilities. The resulting Bayesian Model Averaging (BMA), while appealing, presents several computational challenges due to the potentially extremely large number of elements in the summation and the consequent intractability of the likelihood. In light of this, Madigan and Raftery (1994) propose to scale graphical models by applying the Occam's window approach. The same approach was then applied by the authors to linear regression models in (Raftery et al. 1997). Wasserman (2000) noted that for the case of nested models, Bayesian Selection implies the Occam's principle. Occam's principle was also mentioned in the review by Fragoso et al. (2018) as a "popular criterion" when approximations can be used to compute the posterior probabilities in a computationally efficient manner (Kass and Wasserman 1995; Eicher et al. 2011; Hu et al. 2018).

When the data structure is completely unknown, Frequentist Model Averaging (FMA) can be used over BMA, as it does not require any prior to be defined by the user (Wang et al. 2009). Recently, Mitra et al. (2019) developed a general framework for FMA, where the model averaging weights estimator is determined without any knowledge of the data structure. Moreover, optimal weights for the model averaging are suggested. FMA approaches have also been developed for specific forms of predictors, such as the threshold models (Gao et al. 2019), the logit models (Wan et al. 2014), and the linear mixed-effects models (Chen et al. 2013).

Contribution In this work, we propose a new Bayesian Model Averaging approach. In particular, we propose a weighting structure and an efficient estimation mechanism. Non overlapping subsets form a partition of the original set of points. A model is estimated in each subset, that is defined in a space that has lower dimension with respect to the original points. Each of the models generates a predictor, and these predictors are weighted accounting for both the dimensionality of the associated embedding as well as the size of the training set.

A preliminary version of this manuscript was presented in (Xuereb et al. 2019), where Principal Component Analysis (PCA) was used to obtain *a single* lower-dimensional model and prediction. In this paper, we present, for the first time, the new model averaging approach that can consider multiple low-dimensional embeddings. The result is the Stochastic Gaussian Process Model Averaging (SGPMA) developed for the efficient estimation of a Gaussian Process for noisy high-dimensional data, when the training data set contains a large number of data points.

Section 2 describes the Bayesian Stochastic GP model. Section 3 develops the Stochastic GP Model Averaging algorithm: the model averaging is expressed along with an estimation of the models priors. Moreover, Section 3 presents low-embedding algorithms based on random and PCA embeddings. Section 4 conducts numerical experiments to show the effectiveness of our methodology. Final Section 5 shows some future work possibilities.

2 STOCHASTIC GAUSSIAN PROCESS MODEL

Gaussian Process models have been successfully used for prediction of noisy responses, and several implementations have been proposed in the literature (Ankenman et al. 2010; Chen et al. 2013). The model presented by Yin et al. (2011) allows to explicitly handle heteroscedastic noise, and a follow up paper proposed an efficient Bayesian scheme for the hyper-parameter estimation, which we use in this work (Ng and Yin 2012). Let $\mathbf{X} \in \mathbb{R}^{k \times d}$ a matrix of input locations such that its i^{th} row is a d-dimensional sample $\mathbf{x}_i$, $i = 1, ..., k$, and $\mathbb{X}$ the feasible set. Let $\mathbf{Y} \in \mathbb{R}^k$ the vector with elements $\bar{y}(\mathbf{x}_i)$, the noisy evaluation associated to location $\mathbf{x}_i, \forall i$. Under a Gaussian Process modeling setup, the simulation response of any $\mathbf{x} \in \mathbb{X}$ is interpreted as a random process $Y(\mathbf{x})$ defined as:

$$Y(\mathbf{x}) = S(\mathbf{x}) + \varepsilon(\mathbf{x}) = \mu(\mathbf{x}) + \delta(\mathbf{x}) + \varepsilon(\mathbf{x}), \tag{1}$$

where the Gaussian Process $S(\mathbf{x})$ is the sum of a deterministic mean function $\mu(\mathbf{x})$, and a Gaussian Process $\delta(\mathbf{x})$. The process $\varepsilon(\mathbf{x})$ models the noise in the function evaluations, assumed to be independent across locations. The processes $\delta(\mathbf{x})$ and $\varepsilon(\mathbf{x})$ describe the dependency among locations, which is fully represented by the covariance function

$$\mathrm{Cov}\left(Y(\mathbf{x}_j), Y(\mathbf{x}_l)\right) = \begin{cases} c_1^*(\mathbf{x}_j) + c_1 & \text{if } d_{jl} = 0 \\ c_1 \,\mathrm{corr}\left(d_{jl}, \theta\right) & \text{if } d_{jl} \neq 0 \end{cases} \qquad \forall j, l = 1, \dots, k, \tag{2}$$

where $c_1^*(\mathbf{x}_j)$ is the variance associated to the noise ε at location $\mathbf{x}_j$, $\mathbf{c}_1^*$ the vector of $c_1^*(\mathbf{x}_j)$'s, c_1 represents the process variance, and d_{jl} the distance between $\mathbf{x}_j$ and $\mathbf{x}_l$. corr$(\cdot)$ is a correlation function.
Under the assumptions that the hierarchical priors are defined as $\beta | c_1 \sim \mathrm{N_p}(w_0, c_1 Q_0)$, $c_1 \sim \mathrm{IG}(\alpha, \gamma)$, $\theta_j \sim \mathrm{G}(a, b), j = 1, 2, \dots, d$, and θ and $\mathbf{c}_1^*/c_1$ are known, the optimal predictor results (Ng and Yin 2012):

$$Y(\mathbf{x}_0) \sim \mathrm{T}_1\left(\alpha_{\mathbf{Y}}, \mu(Y(\mathbf{x}_0)|\mathbf{Y}), V(Y(\mathbf{x}_0)|\mathbf{Y})\right). \tag{3}$$

$$\mu(Y(\mathbf{x}_0)|\mathbf{Y}) = B(\mathbf{x}_0) M^{-1}\lambda + C_0^T C^{-1}\left(\mathbf{Y} - BM^{-1}\lambda\right), \alpha_{\mathbf{Y}} = k + 2\alpha \tag{4}$$

$$V(Y(\mathbf{x}_0)|\mathbf{Y}) = \left(2\gamma + \mathbf{Y}^T C^{-1}\mathbf{Y} + w_0^T Q_0^{-1} w_0 - \lambda^T M^{-1}\lambda\right) \frac{1 - C_0^T C^{-1} C_0 + \Lambda M^{-1}\Lambda^T}{2\alpha_{\mathbf{Y}}}.$$

where $\lambda = B^T C^{-1}\mathbf{Y} + Q_0^{-1} w_0$, $M = B^T CB + Q_0^{-1}$, $\Lambda = C_0^T C^{-1} B - B(\mathbf{x}_0)$ (with B and $B(\mathbf{x}_0)$ the known regression matrices of $\mathbf{X}$ and $\mathbf{x}_0$ respectively, C and C_0 the covariance matrices of $\mathbf{X}$, and between $\mathbf{x}_0$ and $\mathbf{X}$, respectively, as defined by equation (2)). The sample variance is the common estimator for $c_1^*(\mathbf{x}_j)$, while θ and c_1 can be estimated by the MLE method.

3 THE STOCHASTIC GAUSSIAN PROCESS WITH MODEL AVERAGING (SGPMA)

We develop a Bayesian Model Averaging approach that uses n predictors of the type in (2) generated by training n models over a partition of the matrix $\mathbf{X}$ of the sampled points. The basic idea is to obtain a predictor for the original noisy function defined in a high-dimensional space, by averaging the predictions generated by n lower dimensional models.

Algorithm 1 shows an overview of the proposed approach. Let $\mathbf{X} \in \mathbb{R}^{k\times d}$ be such that the i^{th} row is a d-dimensional sample $\mathbf{x}_i, i = 1,...,k$, and let $\mathbf{Y} \in \mathbb{R}^k$ be the vector with elements $\bar{y}(\mathbf{x}_i)$, the noisy evaluation associated to location $\mathbf{x}_i, \forall i$. We consider n predictors, each trained by an input formed by k_i elements from $\mathbf{X}$ such that $\sum_i k_i = k$ and all locations from $\mathbf{X}$ are used for training. Such inputs are referred to as $\tilde{\mathbf{X}}_i, i = 1,\dots,n$, and the associated evaluations are the k_i dimensional vectors $\mathbf{Y}_i$ with $i = 1,\dots,n$. The elements of each $\tilde{\mathbf{X}}_i$ are further projected onto a d_i-dimensional space with $d_i < d$, resulting into the projected inputs $\mathbf{X}_i, i = 1,\dots,n$. Projections are obtained using a specified mechanism (random projections and Principal Components Analysis in this paper). In SGPMA, both n and d_i are user defined.

The predictor for a point $\mathbf{x}_0 \in \mathbb{R}^d$ is obtained by performing model averaging of n predictors of the type in equation (4), where the n models are defined in a lower dimensional space $d_i, i = 1,\dots,n$. The resulting predictor is

$$\hat{Y}(\mathbf{x}_0) = \sum_{i=1}^{n} \mu_i(Y_i(\pi_i(\mathbf{x}_0))|\mathbf{Y}_i)w^i(\mathbf{X}_i), \tag{5}$$

where $\pi_i(\mathbf{x}_0)$ is the projection of $\mathbf{x}_0$ in d_i dimensions. We will introduce two different projection operators π^{random}, and π^{PCA}.

Algorithm 1: A Bayesian Approach for Modeling Responses with High Dimensional Input

```
Initialization: Initial sample matrix $\mathbf{X} \in \mathbb{R}^{k\times d}$ and the corresponding function values $\mathbf{Y} \in \mathbb{R}^k$; number of embeddings to construct $n$;
Subset Generation;
Generate the subsets sizes $k_i < k, i = 1,\dots,n$, and the relative embedding dimension $d_i < d, i = 1,\dots,n$;
for $i = 1,\dots,n$ do
    $[\tilde{\mathbf{X}}_i, \mathbf{Y}_i] \leftarrow$ SELECT$(\mathbf{X}, \mathbf{Y}, k_i)$ returns a matrix of $k_i$ samples (rows) from the sample matrix $\mathbf{X}$, and the related values from the vector $\mathbf{Y}$, to be assigned to the sub-matrix $\tilde{\mathbf{X}}_i$;
    Derive a $d_i$-dimensional embedding for $\tilde{\mathbf{X}}_i$, $\mathbf{X}_i \leftarrow$ DIM-EMBED$(\tilde{\mathbf{X}}_i, \mathbf{Y}_i, d_i)$;
    Estimate the parameters of the Gaussian Process $\mathscr{M}_i$ using the training set $\mathbf{X}_i, \mathbf{Y}_i$
end for
Weight the models using $w^i \leftarrow$ BAYESWEIGHT$(\{\mathscr{M}_i\}_{i=1}^n), i = 1,\dots,n$;
Return $\hat{Y}(\mathbf{x}) = \sum_i w^i \hat{Y}_i(\mathbf{x})$
```

3.1 Generating Subsets and Low Dimensional Embeddings

The subset generation procedure is designed to return n sets of input locations and the associated outputs starting from the original matrix $\mathbf{X}$, and vector $\mathbf{Y}$, respectively. While the number of subsets n, and the related dimensions d_i are user-defined, the algorithm decides the number of points to assign to each subset, i.e., $k_i, i = 1, 2, \dots n$. Once k_i, $\forall i$ are defined, the projections need to be executed for all locations onto the related d_i, $\forall i$. Once the subsets are defined, the algorithm assigns the locations in matrix $\mathbf{X}$ to the n subsets. Here we randomly assign the locations to the n subsets, with the procedure "SUBSAMPLES" in Algorithm 3.

Finally, we propose two embedding procedures: Random Embedding of the samples and dimensions (Algorithm 2 with Using-PCA = False); Random Embedding of the samples and Principal Component Analysis of the dimensions (Algorithm 2 with Using-PCA = True).

Based on a randomly generated index set of size d_i (amongst the d original dimensions), the "RANDOM-EMBED" procedure in Algorithm 4 selects the subdimensions of the original input that are still within the index set. The procedure "PCA-EMBED" in Algorithm 5 uses Principal Components Analysis (PCA) to low-embed the dimension. It can be noted that for an out-of-sample location $\mathbf{x}_0$, the associated predictor $Y(\pi_i(\mathbf{x}_0))$ from the low-dimensional model (k_i, d_i) is obtained by first projecting $\mathbf{x}_0$ onto the lower-dimensional space of the model (applying $\pi_i(\mathbf{x}_0)$), then predicting its answer with the model.

```
Algorithm 2: Low Size Low Dimension Subset Generation
Inputs: original data $\mathbf{X},\mathbf{Y}$, number $n$ of low embeddings to generate, number of points in each
    embedding $\{n_i\}_{i=1}^{k} : k_i < k \,\forall i$, and dimension of the low embedding $\{d_i\}_{i=1}^{n} : d_i < d \,\forall i$;
    boolean Using-PCA;
for $i = 1,\dots,n$ do
    Samples Selection: Randomly choose an index set $I_i^S \subseteq \{1,...,k\},\ |I_i^S| = k_i\ (k_i < k)$;
    $\tilde{\mathbf{X}}_i, \mathbf{Y}_i \leftarrow$ SUBSAMPLES $\left(\mathbf{X}, \mathbf{Y}, I_i^S\right)$;
    if Using-PCA then
        $\mathbf{X}_i \leftarrow$ PCA-EMBED $\left(\tilde{\mathbf{X}}_i, d_i\right)$;
    else
        $\mathbf{X}_i \leftarrow$ RANDOM-EMBED $\left(\tilde{\mathbf{X}}_i, d_i\right)$;
    end if
end for
```

```
Algorithm 3: Sub Procedure SUBSAMPLES
Inputs: original data $\mathbf{X},\mathbf{Y}$, sample reduction index set $I_i^S$;
Create the sample reduction matrix $M_{S,i} = \left[m_{lj}^S\right]_{l\in\{1,...,k_i\},j\in\{1,...,k\}}$, $m_{lj}^S = \begin{cases} 1 & \text{if } I_{li}^S = j \\ 0 & \text{otherwise} \end{cases}$;
Output: Return $\tilde{\mathbf{X}}_i = M_{S,i}\mathbf{X}$ and $\mathbf{Y}_i = M_{S,i}\mathbf{Y}$;
```

```
Algorithm 4: Sub Procedure RANDOM-EMBED
Inputs: sample-reduced data $\tilde{\mathbf{X}}_i$ dimension to reduce the data into $d_i$;
Dimensions Selection: Randomly choose an index set $I_i^D \subseteq \{1,...,d\},\ |I_i^D| = d_i\ (d_i < d)$;
Create the dimension reduction matrix $M_{D,i} = \left[m_{lj}^D\right]_{l\in\{1,...,d_i\},j\in\{1,...,d\}}$, $m_{lj}^D = \begin{cases} 1 & \text{if } I_{li}^D = j \\ 0 & \text{otherwise;} \end{cases}$;
Output: Return $\mathbf{X}_i = \tilde{\mathbf{X}}_i M_{D,i}^T$
```

```
Algorithm 5: Sub Procedure PCA-EMBED
Inputs: sample-reduced data $\tilde{\mathbf{X}}_i$, dimension to reduce the data into $d_i$;
Compute the covariance matrix $C_i$ of $\tilde{\mathbf{X}}_i$;
Compute the eigenvectors and eigenvalues of $C_i$;
Order the eigenvectors of $C_i$ in a descending way according to their eigenvalues;
Create the matrix $M_{D,i}$, whose columns are the $d_i$ first eigenvectors;
Output: Return $\mathbf{X}_i = \tilde{\mathbf{X}}_i M_{D,i}^T$
```

3.2 Bayesian Approach to Low Dimensional Predictors and Estimation of the Weighting Structure

Given n such that $\forall i = 1,...,n, \mathbf{X}_i \in \mathbb{R}^{k_i \times d_i}, \mathbf{Y}_i \in \mathbb{R}^{k_i}, k_i < k, d_i < d$, n models $\mathscr{M}_i, i = 1,\dots,n$ will be estimated.

$$\mathscr{M}_i = \left\{ p_{\phi_i} : \phi_i = (\theta_i, c_i)^T \in \mathbb{R}^{d_i+1} \right\}, \tag{6}$$

where

$$p_{\phi_i}(y) = \frac{1}{(2\pi)^{\frac{k_i}{2}}\sqrt{det\left(C_i\right)}} \exp\left(-\frac{1}{2}\left(y - B_i\hat{\beta}_i\right)^T C_i^{-1}\left(y - B_i\hat{\beta}_i\right)\right),$$

$$C_i = \left[c_{jl}^{(i)}\right]_{j,l\in\{1,\dots,k_i\}}, \forall j,l \in \{1,...,k_i\}, c_{jl}^{(i)} = \mathrm{Cov}\left(\mathbf{Y}_i[j], \mathbf{Y}_i[l]\right) = \begin{cases} \hat{c}_1\left(\mathbf{X}_i[l]\right)^* + c_i \text{ if } d_{jl}^{(i)} = 0 \\ c_i \operatorname{corr}\left(d_{jl}^{(i)}, \theta_i\right) \text{ if } d_{jl}^{(i)} \neq 0 \end{cases},$$

$$d_{jl}^{(i)} = \mathrm{dist}(\mathbf{X}_i[j], \mathbf{X}_i[l]), Y_i(\mathbf{x}_0) \sim \mathrm{T}_1\left(\alpha_{i_{\mathbf{Y}_i}}, \mu_i(Y(\mathbf{x}_0)|\mathbf{Y}_i), V_i(Y(\mathbf{x}_0)|\mathbf{Y}_i)\right), \hat{c}_i\left(\mathbf{X}_i[j]\right)^* = \frac{s^2\left(\mathbf{X}_i[j]\right)}{r}.$$

Each model $\mathscr{M}_i$ is trained over $(\mathbf{X}_i, \mathbf{Y}_i)$. $\mathbf{X}_i[j]$ denotes the j^{th} element (sample) of the low-embedded matrix $\mathbf{X}_i$, while $\mathbf{Y}_i[j]$ the j^{th} element of $\mathbf{Y}_i$. ϕ_i is the vector of unknown parameters of the model $\mathscr{M}_i$. Each parameter θ_i is a vector of d_i elements and $dist(\cdot)$ a distance function. $s^2(\mathbf{X}_i[j])$ is the sample variance and r the number of replications. The likelihood of a model $\mathscr{M}_i$ is expressed as $L_i(\phi_i\ ) = p_{\phi_i}(\mathbf{Y}_i)$. Note that the non-central t distribution is valid within the assumptions made in Section 2 ($\beta_i|c_i \sim \mathrm{N_p}(w_{0i}, c_iQ_{0i})$, $c_i \sim \mathrm{IG}(\alpha_i, \gamma_i)$, $\forall j \in [d_i], \theta_{ij} \sim \mathrm{G}(a_i, b_i)$, and θ_i and c_i^*/c_i are known).

Our SGPMA approach uses n models of the type (6) and returns the following predictor for a point $\mathbf{x}_0 \in \mathbb{R}^d$:

$$\begin{aligned}\hat{Y}\left(\pi\left(\mathbf{x}_0\right)\right) &= \sum_{i=1}^{n} \mu_i(Y(\pi_i(\mathbf{x}_0))|\mathbf{Y}_i) P\left(\mathscr{M}_i|\mathbf{X}_i, \mathbf{Y}_i\right) = \sum_{i=1}^{n} \mu_i(Y(\pi_i(\mathbf{x}_0))|\mathbf{Y}_i) \frac{P\left(\mathbf{Y}_i|\mathscr{M}_i\right) \cdot P\left(\mathscr{M}_i\right)}{\sum_{j=1}^{n} P\left(\mathbf{Y}_i|\mathscr{M}_i\right) \cdot P\left(\mathscr{M}_i\right)} \\ &= \sum_{i=1}^{n} \mu_i(Y(\pi_i(\mathbf{x}_0))|\mathbf{Y}_i) \frac{L_i(\hat{\phi}_i) \exp\left(-\frac{|\phi_i|}{2}\log k_i\right) P\left(\mathscr{M}_i\right)}{\sum_{j=1}^{n} L_j(\hat{\phi}_j) \exp\left(-\frac{|\phi_j|}{2}\log k_j\right) P\left(\mathscr{M}_j\right)}. \qquad (7)\end{aligned}$$

The assumptions on the prior density are embedded within the general assumption made by the approximation given in (3). Then, the posterior for each model (given by the Bayes' formula) and the conditional probability of obtaining the true answer with the model $\mathscr{M}_i$ are expressed below:

$$P\left(\mathscr{M}_i|\mathbf{Y}_i\right) = \frac{P\left(\mathbf{Y}_i|\mathscr{M}_i\right) P\left(\mathscr{M}_i\right)}{\sum_{j=1}^{n} P\left(\mathbf{Y}_j|\mathscr{M}_j\right) P\left(\mathscr{M}_j\right)} \quad \forall i = 1, \dots, n,$$

$$P\left(\mathbf{Y}_i|\mathscr{M}_i\right) = \int p_{\phi_i}(\mathbf{Y}_i) p_i(\phi_i) d\phi_i = \int L_i(\phi_i\ ) p_i(\phi_i) d\phi_i \approx L_i(\hat{\phi}_i) \exp\left(-\frac{|\phi_i|}{2}\log k_i\right) \quad \forall i = 1, \dots, n,$$

where the approximation is given by Wasserman (2000). $\hat{\phi}_i$ is the MLE of ϕ_i (Ng and Yin 2012).

Based on Section 2, it is possible to calculate the predictive distributions means $\mu_i(Y(x_0^i)|\mathbf{Y}_i)$ as well as the marginal likelihoods $L_i(\hat{\phi}_i)$. These means and likelihoods, each based on sample and dimension-reduced input $\mathbf{X}_i$, should be more computationally tractable than the prediction of a model based on the original input $\mathbf{X}$. The estimation of the model priors $P(\mathscr{M}_i)$ will be discussed in the next section.

Weights Estimation One possibility for the weights, $P(\mathscr{M}_i)$, is to use a noninformative uniform prior (Wasserman 2000). Priors can also be tailored to specific classes of models. For linear regression models, a binomial prior is often used to reflect the importance of the variables (Hoeting et al. 1999; Steel 2011). The intuition behind the binomial prior is that if a model $\mathscr{M}_i$ contains more covariates than $\mathscr{M}_j$, then $\mathscr{M}_i$ should have larger associated weight compared to $\mathscr{M}_j$, because the more the covariates, the higher the prediction power of the model. However, for these models, dimensionality reduction is not discussed.

Here, we associate the idea of increased prediction power as a function of the dimensionality d_i and sample size k_i. For each $\mathscr{M}_i$ we use a different training set characterized by a different number of points k_i, and different dimensionality d_i, and *both of them characterize the prediction power of the resulting model.* In other words, for each model $\mathscr{M}_i$, a proxy of this quantity is the pair (k_i, d_i). Our SGPMA considers a power function of the percentages of the sample size and dimension with respect to their original sizes, where the relative importance of the d_i can be controlled through η. Given the model $\mathscr{M}_i$ as in equation (6), we propose the following prior form:

$$P(\mathscr{M}_i) = \frac{p_i}{\sum_{j=1}^{n} p_j}, p_i = \left(\frac{k_i}{k}\right)^2 \times \left(\frac{d_i}{d}\right)^\eta \; \forall i = 1, ..., n. \tag{8}$$

From equation (8), it is possible to develop a procedure for the prediction of the Stochastic GP model averaging when the input is high-dimensional.

Motivating Example In order to build the intuition behind the weights in equation (8) we show, with an example, that there exists an exploitable relationship between the (k_i, d_i)-pair and the prediction error associated to a model estimated with k_i samples, and a d_i-dimensional embedding. We use the Griewank function (Locatelli 2003):

$$f(x) = \sum_{i=1}^{d} \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, x \in [-600, 600]^d. \tag{9}$$

We add a noise function $\delta = 0.8$, constant across the input space (given a location $\mathbf{x}$ a normal noise $\mathscr{N}(0, \delta)$ is added to the output of the function in (9)). We set $k = 500$, and $d = 50$ and test different combinations of sub-sample size and dimensionality (k_i, d_i). Since the evaluations are affected by noise, 15 replications are performed for each location. The idea is that, if $\text{error}[\mathscr{M}_i(k_i, d_i)] < \text{error}[\mathscr{M}_j(k_j, d_j)]$, then $P(\mathscr{M}_i) > P(\mathscr{M}_j)$.

For each combination (k_i, d_i), we estimate the model 30 times. Table 1 summarizes the absolute errors resulting from each model prediction, calculated with a $B = 100d = 5000$-samples validation dataset. The error is calculated as

$$\text{error} = \frac{1}{B} \sum_{i=1}^{B} \left| \hat{Y}(\mathbf{x}_{\text{test},i}) - f(\mathbf{x}_{\text{test},i}) \right|, \tag{10}$$

where $\hat{Y}$ is the prediction and f the true function value.

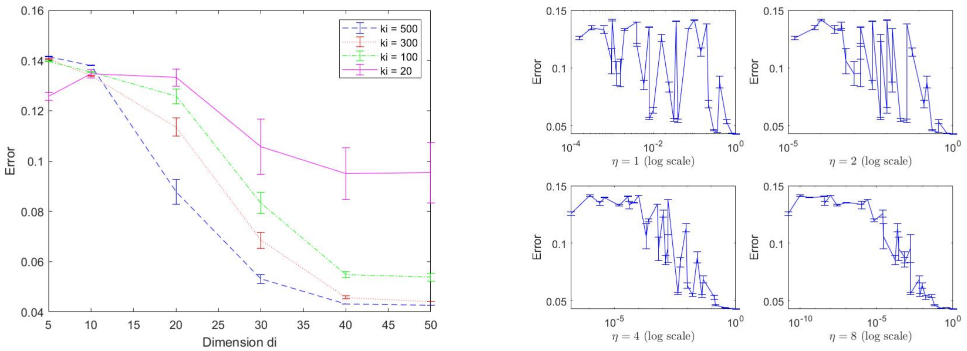
Figure 1(a) shows the error against the dimension d_i, for the different values of $k_i = \{500, 300, 100, 20\}$. It can be observed that, as hypothesized, the model error depends on the *pair* (k_i, d_i). Hence, we argue that a model prior should consider both k_i and d_i. In fact, if our weight was only to consider the sample size k_i, models would be mistakenly ranked: in our example, the error of the model $(k_i = 300, d_i = 40)$ is higher than the error of $(k_i = 500, d_i = 40)$, but lower than the error of $(k_i = 500, d_i = 30)$. Similar results were obtained for different values of η.

Figure 1(b) supports the intuition that larger values of the proposed weights correspond to "better" models, i.e., with lower associated errors. In the figure, the models are ordered according to the value of the associated $p_i = (k_i/k)^2 \times (d_i/d)^\eta$, with different values of η (equation(6)). More experiments were performed for different values of η, and we noticed that $\eta = 8$ was performing best (i.e. with the lowest error) in our experimental setup.

Interestingly, we observed that the model performance started deteriorating for values of $\eta > 8$. This observation highlights the importance of such parameter in determining the effectiveness of the weighting approach. Also, considering Figure 1(a), and as we previously highlighted, k_i and d_i interact in determining the performance of the model. In this preliminary version, SGPMA assumes that η is supplied by the user. Nevertheless, significant opportunities can be found in the understanding of the "optimal" relationship between the k_i and d_i exponent, leading to an automated derivation of the weight without requiring the user to set it. In the numerical section, we will further explore the impact of η.

Table 1: Effect of k_i and d_i over the prediction accuracy of a single model of the type (6)

Exp	k_i	500					
	d_i	50	40	30	20	10	5
Error	$\times 10^{-2}$	4.26 ± 0.01	4.32 ± 0.01	5.31 ± 0.02	8.77 ± 0.49	13.80 ± 0.01	14.15 ± 0.01
Exp	k_i	300					
	d_i	50	40	30	20	10	5
Error	$\times 10^{-2}$	4.41 ± 0.01	4.58 ± 0.06	6.86 ± 0.31	11.35 ± 0.36	13.39 ± 0.01	14.06 ± 0.03
Exp	k_i	100					
	d_i	50	40	30	20	10	5
Error	$\times 10^{-2}$	5.39 ± 0.15	5.48 ± 0.13	8.33 ± 0.42	12.58 ± 0.29	13.54 ± 0.03	13.97 ± 0.01
Exp	k_i	20					
	d_i	50	40	30	20	10	5
Error	$\times 10^{-2}$	9.54 ± 1.20	9.50 ± 1.40	10.58 ± 1.10	13.32 ± 0.34	13.47 ± 0.17	12.57 ± 0.16



(a) Performance for different dimensions (b) Error as a function of the associated weight

Figure 1: Impact of k_i, d_i over the test models in Table 1

4 PRELIMINARY RESULTS

To evaluate the performance of SGPMA, we ran several experiments considering different dimensionality and sample sizes, and corresponding dimensionality and number of points for the subsets. The d-dimensional Griewank function given in equation (9) was used across all the tests.

Hyperparameters Priors Following the guidelines of Qian and Wu (2008), a "vague prior" was adopted for c_i and a "location-flat prior" for $\beta_i|c_i$. Such setting translates into $\alpha_i = 2$, $\gamma_i = 1$, $w_{0i} = 0$, and Q_{0i} the identity matrix. As for the parameter η defined in equation (8), and previously discussed, we used the set of possible values $\{0.5, 3, 5, 7, 9\}$, with the idea of analyzing different scenarios in terms of the relative weight (when $\eta < 2$ the dimensionality is less important than the number of samples, and vice versa).

Experiments settings $k = 10 \times d$ input design points were generated using a Latin Hypercube over a support $[0, 10]^d$. Function value at each location is a normal distribution with mean equal to the true function value at that point, and constant (homogeneous) variance $\delta = 0.8$ across the entire solution space. A single experimental condition is defined by the tuple $\{d, n = n_{\text{random}} + n_{\text{PCA}}, S_k, S_d\}$, where n_{random} subsets were processed using Algorithm 2 with Using-PCA = False , whereas n_{PCA} subsets use it with Using-PCA = True. The set S_k contains the possible sample sizes that can be assigned to each model $\mathscr{M}_i, i = 1, \ldots, n$. As an example, if $d = 5$ and $S_k = \{10, 20\}$, each model $\mathscr{M}_i, i = 1, \ldots, n$ can be trained with 10 or 20 points (as long as $\sum_i k_i = 50$). Similarly, S_d is the set of possible, reduced, dimensions that can be assigned to each model $\mathscr{M}_i, i = 1,\ldots,n$.

In order to generate the experimental conditions, we solved a mixed integer-linear program having as decision variables the number of models n, and the sample sizes $k_i, i = 1,\ldots,n$. The objective function is to minimize the differences $|S_{ki}[j] - S_{ki}[l]| \forall i, j, l$ and $|n_{\text{random}} - n_{\text{PCA}}|$, satisfying the partition constraint on the possible sample size assignments defined by the set S_k.

The resulting experimental conditions are listed in Table 2.

Table 2: Description of the experiments

E	d	k	n		S_d	S_k
			n_{random}	n_{PCA}		
E_1	50	500	5	6	$\{2,4\}$	$\{5,35,70,100\}$
E_2	100	1000	8	8		
E_3	150	1500	14	14		
E_4	200	2000	15	15		
E_5	50	500	11	0		
E_6			0	11		
E_7			5	6	$\{10,15\}$	

Validation Each element of the vector $\mathbf{Y}$ represents the average of 15 independent replications. SGPMA was macro-replicated 30 times for each experiment. At each macro-replication, the SGPMA performance was estimated over $100 \times d$ out-of sample locations. The following benchmark models were considered for sake of comparison:

- OR: this refers to the original Stochastic GP model without embedding and without sample reduction;
- UMA: model averaging using the same models as SGPMA, but weighting them uniformly;
- LM: the model, amongst those estimated by SGPMA, with the lowest associated MSE;
- HM: the model, amongst those estimated by SGPMA, with the largest associated MSE;
- AM: the model, amongst those estimated by SGPMA, with the mode associated MSE.

Performance We considered the error metric defined in equation (10), and the computational time.

Results Tables 3-5 show the obtained errors and fitting times for the performed experiments. η^* in Table 3 is the value of η for which the model averaging error is the lowest, taken over the search space $\{0.5, 3, 5, 7, 9\}$. In Table 4, this search space is modified for each experiment.

Table 3: Model Averaging (MA) vs. MA with uniform prior (UMA) vs. original model (OR)

E	d	k	η^*	Error $\times 10^{-2}$			Time [s]	
				MA with η^*	UMA	OR	MAs	OR
E_1	50	500	5	6.70 ± 0.03	8.74 ± 0.11	4.29 ± 0.01	1	13
E_2	100	1000	5	9.33 ± 0.10	11.05 ± 0.72	5.98 ± 0.01	2	311
E_3	150	1500	9	11.97 ± 4.08	11.84 ± 2.97	7.31 ± 0.01	2	2731
E_4	200	2000	5	12.38 ± 0.19	12.67 ± 0.88	NA*	4	>24h
E_5	50	500	5	6.03 ± 0.02	8.90 ± 0.34	4.29 ± 0.01	2	13
E_6			5	7.16 ± 0.07	8.98 ± 0.34		2	

Discussion A first observation from Table 3 is that applying Algorithm 1 on high dimensional problems ($d \geq 50$ in these numerical experiments) decreases the computational fitting time. Experiments $E_1 - E_4$ show how the higher the original dimension, the faster the novel algorithm compared to the OR model. Comparing the results from experiments E_7 (in Table 4) and E_1 (in Table 3), focusing only on the fitting

Table 4: Effect of the dimension weight η for the case $E_7 := \{d = 50, k = 500\}$

E	Search space for η	η^*	Error $\times 10^{-2}$			Time [s]	
			MA with η^*	UMA	OR	MAs	OR
$E_{7,1}$	$\{1,...9\}$	9	9.87 ±0.48	17.12 ± 6.91	4.29 ± 0.01	6	13
$E_{7,2}$	$\{10,...,100\}$	12	9.41 ± 0.45				
$E_{7,3}$	$0.01 \times \{0,...,100\}$	1	16.23 ± 5.98				

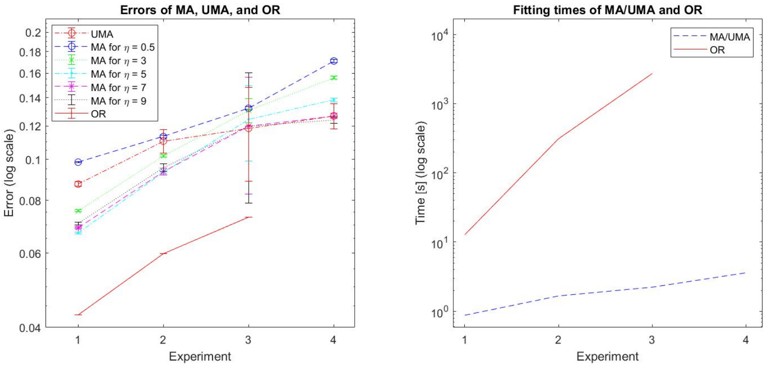


Figure 2: Model Averaging (MA) vs. Original model (OR)

Table 5: Comparison of SGPMA against the benchmarks

E	d	k	Error $\times 10^{-2}$			
			MA	LM	AM	HM
E_1	50	500	6.70 ± 0.03	12.19 ± 0.76	16.11 ± 1.68	22.34 ± 5.51
E_2	100	1000	9.33 ± 0.10	14.59 ± 0.08	20.36 ± 4.43	31.83 ± 4.16
E_3	150	1500	11.97 ± 4.08	15.25 ± 0.49	28.93 ± 13.60	57.29 ± 18.42
E_4	200	2000	12.38 ± 0.19	15.32 ± 0.35	24.99 ± 10.91	54.04 ± 18.05

times, we observe that, if the computational cost is of concern, it is better to use lower dimensional embeddings for SGPMA. This observation is particularly relevant in sequential optimization settings where the SGPMA model is estimated several times. Nonetheless, as the results show, the higher computational efficiency comes at the cost of lower prediction accuracy. This trade-off between efficiency gain and loss of accuracy can be beneficial in many time sensitive situations. However, it is problem specific and dependent on the risks associated with time delay and accuracy loss. In their book, Taylor and Vanmarcke (2002) illustrate this consideration with an earthquake estimation. When estimating the possible seriousness of an incoming earthquake, a 50 % accuracy loss is acceptable, however this same accuracy loss will lead to a wrong estimation of resource allocation.

Figure 2 plots the results from experiments $E_1 - E_4$ (x-axis) providing a visual comparison between SGPMA, UMA and the original model, and, for SGPMA, the prediction error is plotted for different values of the factor η. The results show how the dimension of a model, compared to its sample size, influences the prediction stronger ($\eta > 2$ leads to lower errors than $\eta \leq 2$).

Also, it appears that the value of η^*, i.e., the value of η with the associated best performance, varies substantially across conditions, implying that the starting values of (d,k) exert an influence on the value of the factor. Moreover, if we compare the results from experiments $E_{7,1}$ and E_1 (Table 3), we observe how the value of η^* appears to be influenced by the possible subdimensions S_{di}. While future work will be necessary on the search for better values of η, these insights are important, because choosing a good value of η appears to discriminate SGPMA from model averaging with uniform priors (Table 3).

E_5 and E_6 of Table 3 highlight the importance of choosing a right low-dimensional embedding. Here, PCA seems to be less efficient than random embedding for the case $\{d = 50, k = 500\}$. When compared to E_1, it seems better to use a unique random embedding than a mixture of PCA and random embeddings. However, it also seems better to use the mixture than PCA only. The type of embedding can be important.

Finally, Table 5 shows that model averaging always outperforms the best performing model within the set $\{\mathscr{M}_i\}_{i=1}^n$, independently from the embedding approach (random, PCA, or mixed). In our numerical experiments, SGPMA outperforms LM in all the cases. This tends to confirm that model averaging empirically outperforms single model strategies in terms of prediction accuracy.

5 CONCLUSION

This paper presents for the first time the Stochastic Gaussian Process with Model Averaging (SGPMA) algorithm for the prediction of noisy black box functions. SGPMA relies on a novel Bayesian model averaging approach that is used to mix several SGP's. Specifically, several stochastic GPs are estimated, each using a subset of the initial, large, input data, and each projecting the input location onto a lower dimensional space. SGPMA is scalable since any input data set can be separated into smaller, size-controlled, training sets, and each model has reduced dimensionality. The produced predictions are averaged using a weighting scheme based on their marginal likelihood and a "model prior", which we design to be dependent on the size of model training set and the dimensionality being used for the projection.

Our preliminary results show how SGPMA drastically decreases the computing time for model fitting while keeping the prediction accuracy reasonably close to the original model, and our proposed model weighting appears to outperform uniform weighting and shows more consistent performance. Nonetheless, more studies are necessary on the factor η, which appears to have a key role in the performance of SGPMA. The future direction of this work will be to attempt to derive η "optimally", and test alternative projection methods. Finally, SGPMA will be used within a Bayesian Optimization context for large scale optimization.

REFERENCES

Ankenman, B., B. L. Nelson, and J. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58(2):371–382.

Bouhlel, M. A., and J. R. R. A. Martins. 2019. "Gradient-Enhanced Kriging for High-Dimensional Problems". *Engineering with Computers* 35(1):357–373.

Chen, X., B. E. Ankenman, and B. L. Nelson. 2013. "Enhancing Stochastic Kriging Metamodels with Gradient Estimators". *Operations Research* 61(2):512–528.

Chen, X., G. Zou, and X. Zhang. 2013. "Frequentist Model Averaging for Linear Mixed-Effects Models". *Frontiers of Mathematics in China* 8(3):497–515.

Eicher, T. S., C. Papageorgiou, and A. E. Raftery. 2011. "Default Priors and Predictive Performance in Bayesian Model Averaging, with Application to Growth Determinants". *Journal of Applied Econometrics* 26(1):30–55.

Fragoso, T. M., W. Bertoli, and F. Louzada. 2018. "Bayesian Model Averaging: A Systematic Review and Conceptual Classification". *International Statistical Review* 86(1):1–28.

Gao, Y., X. Zhang, S. Wang, T. T. Chong, and G. Zou. 2019. "Frequentist Model Averaging for Threshold Models". *Annals of the Institute of Statisticalop Mathematics* 71(2):275–306.

Gardner, J. R., G. Pleiss, R. Wu, K. Q. Weinberger, and A. G. Wilson. 2018. "Product Kernel Interpolation for Scalable Gaussian Processes". In *Proceedings of the AISTATS 2018*, edited by A. Storkey and F. Perez-Cruz, 1407–1416. Piscataway, New Jersey.

Hayashi, K., M. Imaizumi, and Y. Yoshida. 2019. "On Random Subsampling of Gaussian Process Regression: A Graphon-Based Analysis". arXiv preprint arXiv:1901.09541.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky. 1999. "Bayesian Model Averaging: A Tutorial". *Statistical Science* 14(4):382–417.

Hu, S., A. O'Hagan, and T. B. Murphy. 2018. "Motor Insurance Claim Modeling with Factor Collapsing and Bayesian Model Averaging". *Stat* 7(1).

Kass, R. E., and L. Wasserman. 1995. "A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion". *Journal of the American Statistical Association* 90(431):928–934.

Lanczos, C. 1950. “An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators”. *Journal of Researchof the National Bureau of Standards* 45(4):255–282.

Liu, H., Y. S. Ong, X. Shen, and J. Cai. 2020. “When Gaussian Process Meets Big Data: A Review of Scalable GPs”. *IEEE Transactions on Neural Networks and Learning Systems* Early Access.

Locatelli, M. 2003. “A Note on the Griewank Test Function”. *Journal of Global Optimization* 25(2):169–174.

Lu, X., A. Rudi, E. Borgonovo, and L. Rosasco. 2019. “Faster Kriging: Facing High-Dimensional Simulators”. *Operations Research* Early Access.

Madigan, D., and A. E. Raftery. 1994. “Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam’s Window”. *Journal of the American Statistical Association* 89(428):1535–1546.

Mitra, P., H. Lian, R. Mitra, H. Liang, and M. Xie. 2019. “A General Framework for Frequentist Model Averaging”. *Science China Mathematics* 62(2):205–226.

Ng, S. H., and J. Yin. 2012. “Bayesian Kriging Analysis and Design for Stochastic Simulations”. *ACM Transactions on Modeling and Computer Simulation* 22(3).

Qian, P. Z. G., and C. F. J. Wu. 2008. “Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments”. *Technometrics* 50(2):192–204.

Raftery, A. E., D. Madigan, and J. A. Hoeting. 1997. “Bayesian Model Averaging for Linear Regression Models”. *Journal of the American Statistical Association* 92(437):179–191.

Reich, B. J., H. D. Bondell, and L. Li. 2011. “Sufficient Dimension Reduction via Bayesian Mixture Modeling”. *Biometrics* 67:886–895.

Rudi, A., R. Camoriano, and L. Rosasco. 2015. “Less is More: Nyström Computational Regularization”. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*:1657–1665.

Santner, T. J., B. J. Williams, W. Notz, and B. J. Williams. 2003. *The Design and Analysis of Computer Experiments*, Volume 1. New York, NY: Springer.

Steel, M. F. J. 2011. “Bayesian Model Averaging and Forecasting”. *Bulletin of EU and US Inflation and Macroeconomic Analysis* 200:30–41.

Taylor, C., and E. Vanmarcke. 2002. *Acceptable Risk Processes: Lifelines and Natural Hazards*, Volume 21. ASCE Publications.

Wan, A. T. K., X. Zhang, and S. Wang. 2014. “Frequentist Model Averaging for Multinomial and Ordered Logit Models”. *International Journal of Forecasting* 30(1):118–128.

Wang, H., X. Zhang, and G. Zou. 2009. “Frequentist Model Averaging Estimation: A Review”. *Journal of Systems Science and Complexity* 22:732–748.

Wang, K. A., G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. 2019. “Exact Gaussian Processes on a Million Data Points”. In *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, 14622–14632. Vancouver, Canada.

Wasserman, L. 2000. “Bayesian Model Selection and Model Averaging”. *Journal of Mathematical Psychology* 44(1):92–107.

Williams, C. K., and C. E. Rasmussen. 2006. *Gaussian Processes for Machine Learning*, Volume 2. Cambridge, MA: MIT Press.

Xuereb, M., T. M. Huo, and S. H. Ng. 2019. “Principal Component Analysis for High Dimension Stochastic Gaussian Process Model Fitting”. In *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 632–636. Macao, Macao.

Yin, J., S. H. Ng, and K. M. Ng. 2011. “Kriging Metamodel with Modified Nugget-Effect: The Heteroscedastic Variance Case”. *Computers and Industrial Engineering* 61(3):760–777.

AUTHOR BIOGRAPHIES

MAXIME XUEREB is a Ph.D. candidate in the Department of Industrial Systems Engineering and Management at the National University of Singapore. His email address is maxime.xuereb@u.nus.edu.

SZU HUI NG is Associate Professor and acting Department Head for the Department of Industrial Systems Engineering and Management at the National University of Singapore. She holds B.S., M.S. and Ph.D. degrees in Industrial and Operations Engineering from the University of Michigan. Her research interests include computer simulation and design of experiments. She is a member of IEEE and INFORMS, and a senior member of IIE. Her email address is isensh@nus.edu.sg.

GIULIA PEDRIELLI is Assistant Professor in the School of Computing, Informatics, and Decision Systems Engineering at the Arizona State University. She holds a Ph.D. degree in Mechanical Engineering from the Politecnico di Milano. She is a member of IEEE and INFORMS. Her email address is giulia.pedrielli@asu.edu.