Variable Fidelity Regression Using Low Fidelity Function Blackbox and Sparsification

A. Zaytsev[1,2(✉)]

[1] IITP RAS, 127051 Moscow, Russia
likzet@gmail.com
[2] MIPT, Dolgoprudny, 141700 Moscow, Russia

Abstract. We consider construction of surrogate models based on variable fidelity samples generated by a high fidelity function (an exact representation of some physical phenomenon) and by a low fidelity function (a coarse approximation of the exact representation). A surrogate model is constructed to replace the computationally expensive high fidelity function. For such tasks Gaussian processes are generally used. However, if the sample size reaches a few thousands points, a direct application of Gaussian process regression becomes impractical due to high computational costs. We propose two approaches to circumvent this difficulty. The first approach uses approximation of sample covariance matrices based on the Nyström method. The second approach relies on the fact that engineers often can evaluate a low fidelity function on the fly at any point using some blackbox; thus each time calculating prediction of a high fidelity function at some point, we can update the surrogate model with the low fidelity function value at this point. So, we avoid issues related to the inversion of large covariance matrices — as we can construct model using only a moderate low fidelity sample size. We applied developed methods to a real problem, dealing with an optimization of the shape of a rotating disk.

Keywords: Multifidelity data · Gaussian process · Nonlinear regression · Nyström approximation · Cokriging

1 Introduction

Nowadays most advanced engineers encounter the problem of a surrogate model construction, when it is required to replace an expensive high fidelity function with an inexpensive but precise surrogate model [17]. Typically, to accomplish such a task one generates a sample of points and values of the corresponding high fidelity function at these points, and then using the generated sample and the machinery of regression analysis one constructs a surrogate model. Among various surrogate model construction techniques, the Gaussian process regression remains an attractive approach, as the machinery of this method provides a nonlinear regression model with prediction uncertainty estimate [17,37]). Moreover, Gaussian process framework provides straightforward solutions for classification

A. Gammerman et al. (Eds.): COPA 2016, LNAI 9653, pp. 147–164, 2016.
DOI: 10.1007/978-3-319-33395-3_11

[43], adaptive design of experiments [9] and surrogate based optimization [21] problems.

Another nice property of Gaussian process regression is the ability to treat variable fidelity data (see for example [12,16,22,27,28,35]): one can construct a surrogate model of a high fidelity function using data from high and low fidelity sources (e.g., a high fidelity function can be modeled by an experiment in a wind tunnel, and the low fidelity function can be realized by a computer simulation of the same physical process) and then use this model for surrogate-based optimization. Similar approaches are used for multiple output Gaussian processes modeling [2,8,11,25].

Straightforward maximum likelihood estimation of Gaussian process regression model parameters and application of model to new points require inversion of the covariance matrix of the sample [18]. The covariance matrix of the sample is a square matrix with number of both rows and columns equal to the sample size n. Consequently, as typically the covariance matrix has no specific structure, we need $O(n^2)$ to store the covariance matrix and $O(n^3)$ to invert it. Due to this computational complexity usually not more than a few thousands of points are used when training Gaussian Process regression. As a sample generated using the low fidelity function is often large, because the evaluation of a low fidelity function is significantly cheaper than that of a high fidelity function, the problem is even worse for variable fidelity data.

Currently there are several ways to avoid inversion of the full covariance matrix in Gaussian process regression. Using of Nyström approximation [13] of the covariance matrix has remained a popular approach to do large sample Gaussian process regression inference for more than 10 years [18,36,41]. The idea is to select a subsample of the full sample for which we can do Gaussian process regression inference, and then approximate the full sample covariance matrix and inverse of the full sample covariance matrix by combination of the covariance matrix for selected subsample and covariance between points in the selected subsample and in the full sample. Another approach consists of usage of Bayesian approximate inference to estimate the full sample likelihood with an easy-to-calculate expression [24,42]. Rather popular approach with proved theoretical properties is covariance tapering [19,38]: we suppose that covariance function equals zero for points with distance above the taper parameters, so we obtain sparse covariance matrices, and can proceed them with routines specific to sparse matrices. Hierarchical models move away the computational burden, as they split the sample to separate subsamples, which leads to covariance matrix with specific structure [5,33,39]. However, exact inference is possible if data have some specific structure: for example, [6] has developed an exact inference scheme to construct Gaussian process regression. Another example that works with variable fidelity data of big size with specific structure (we aggregate many low fidelity uncalibrated models using observations at the same points) was presented in [11]. However, as far as we know there are no approaches to large scale variable fidelity Gaussian process regression for data without any specific structure.

Another issue with Gaussian process regression lies in its bad extrapolation properties, since the model prediction at a new point is the weighted sum of values at given training points with weights defined by covariances between points [37]; i.e., the prediction can be determined only locally near the training points, and we need to be careful with points that are far away from the training sample.

We propose two approaches that mitigate the sample size limitation and improve the extrapolation properties of variable fidelity Gaussian process regression. The first approach uses the Nyström approximation to the covariance matrices and relies on the results obtained for a single fidelity data in the Sparse Gaussian process regression framework [18]. The main idea of the second approach is to use the low fidelity function blackbox during the model evaluation, so one can evaluate a low fidelity function on the fly only at the points where it is required to approximate a high fidelity function and use these evaluations to update the surrogate model predictions. While, for simple heuristic models it is a common practice to use a low fidelity function blackbox [1,29,40,44,45], Gaussian process regression doesn't support usage of such an approach in a direct way. As we are able to evaluate the low fidelity function at any point from the design space, we avoid usage of large sample to cover all the design space. Instead, it is sufficient only to get enough points to estimate parameters of Gaussian process regression model.

For proposed approaches we investigate their computational complexity and compare their accuracy using real and artificial data. The real problem at hand is optimization of a rotating disk in an aircraft engine. The problem of a disk shape optimization remains challenging and often involves usage of surrogate modeling [15,26], so it is required to construct accurate surrogate models for maximal stress and radial displacement of the disk used then for surrogate optimization. We compare four approaches to construct the rotating disk surrogate models: Gaussian process (kriging), Gaussian process for variable fidelity data (cokriging) and our approaches — Gaussian process for variable fidelity data with usage of a low fidelity blackbox and large scale variable fidelity Gaussian process regression.

The paper is organized as follows:

- Section 2 describes the Gaussian process regression framework;
- Section 3 outlines the Variable fidelity Gaussian process regression framework;
- Section 4 proposes an approach to construct Sparse Gaussian process regression for variable fidelity data;
- Section 5 describes our approach to Variable Fidelity Gaussian process regression with a low fidelity function blackbox;
- Section 6 provides the results of computational experiments for both real and artificial data,
- Conclusions are given in Sect. 7.

In Appendix we provide proofs of some technical statements and details on low and high fidelity models for rotating disk problem.

2 Gaussian Process Regression for a Single Fidelity Data

We consider a single training sample $D = (\mathbf{X}, \mathbf{y}) = \{\mathbf{x}_i, y_i = y(\mathbf{x}_i)\}_{i=1}^n$, where points $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^d$ and a function value $y(\mathbf{x}) \in \mathbb{R}$. We assume that $y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$, where $f(\mathbf{x})$ is a realization of a Gaussian process, and ε is a Gaussian white noise with variance σ^2. The goal is to construct a surrogate model for the target function $f(\mathbf{x})$.

The mean value and the covariance function

$$k(\mathbf{x}, \mathbf{x}') = \operatorname{cov}(f(\mathbf{x}), f(\mathbf{x}')) = \mathbb{E}\left(f(\mathbf{x}) - \mathbb{E}(f(\mathbf{x}))\right)\left(f(\mathbf{x}') - \mathbb{E}(f(\mathbf{x}'))\right)$$

completely define the Gaussian process $f(\mathbf{x})$. Without loss of generality we assume its mean value to be zero. We also assume that the covariance function belongs to some parametric family $\{k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}'), \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$; i.e., $k(\mathbf{x}, \mathbf{x}') = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$ for some $\boldsymbol{\theta} \in \Theta$. Thus $y(\mathbf{x})$ is also a Gaussian process [37] with zero mean and covariance function $\operatorname{cov}(y(\mathbf{x}), y(\mathbf{x}')) = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') + \sigma^2\delta(\mathbf{x} - \mathbf{x}')$, where $\delta(\mathbf{x} - \mathbf{x}')$ is the delta function. Example of a covariance function, widely used in applications, is the multivariate squared exponential covariance function [37] $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp\left(-\sum_{k=1}^d \theta_k^2 (x_k - x'_k)^2\right)$.

The covariance function parameters $\boldsymbol{\theta}$ and the variance σ^2 are known as fully specifying the data model. We use the Maximum Likelihood Estimation (MLE) of $\boldsymbol{\theta}$ and σ^2 [7,37] to fit the model; i.e., we maximize the logarithm of the training sample likelihood

$$\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \sigma^2) = -\frac{1}{2}\left(n \log 2\pi + \log|\mathbf{K}| + \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}\right) \to \max_{\boldsymbol{\theta}, \sigma^2}, \tag{1}$$

where $\mathbf{K} = \{k_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2\delta(\mathbf{x}_i - \mathbf{x}_j)\}_{i,j=1}^n$ is the matrix of covariances between values $\mathbf{y}(\mathbf{X})$ of the training sample and $|\mathbf{K}|$ is the determinant of $\mathbf{K}$. σ^2 plays the role of a regularization parameter for the kernel matrix $\{k_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n$, being a matrix of covariances between values $f(\mathbf{X})$. The recent theoretical work [10] and the experimental works [4,46] suggest that, under general assumptions, MLE parameters estimates $\hat{\boldsymbol{\theta}}$ are accurate even if the sample size is limited and the model is misspecified.

Using estimates of $\boldsymbol{\theta}$ and σ^2 we can calculate the posterior mean and the covariances of $y(\mathbf{x})$ at new points playing, respectively, the role of a prediction and its uncertainty. The posterior mean $\mathbb{E}(\mathbf{y}(\mathbf{X}^*)|\mathbf{y}(\mathbf{X}))$ at the new points $\mathbf{X}^* = \{\mathbf{x}_i^*\}_{i=1}^{n^*}$ has the form

$$\hat{\mathbf{y}}(\mathbf{X}^*) = \mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{K}^{-1}\mathbf{y}, \tag{2}$$

where $\mathbf{K}(\mathbf{X}^*, \mathbf{X}) = \{k(\mathbf{x}_i^*, \mathbf{x}_j)\}_{i=1,\ldots,n^*, j=1,\ldots,n}$ are the covariances between the values $\mathbf{y}(\mathbf{X}^*)$ and $\mathbf{y}(\mathbf{X})$. The posterior covariance matrix $\mathbb{V}(\mathbf{X}^*) = \mathbb{E}\big[(\mathbf{y}(\mathbf{X}^*) - \mathbb{E}\mathbf{y}(\mathbf{X}^*))^T(\mathbf{y}(\mathbf{X}^*) - \mathbb{E}\mathbf{y}(\mathbf{X}^*)) \,|\, \mathbf{y}(\mathbf{X})\big]$ has the form

$$\mathbb{V}(\mathbf{X}^*) = \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) - \mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{K}^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}^*), \tag{3}$$

where $\mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) = \{k(\mathbf{x}_i^*, \mathbf{x}_j^*) + \sigma^2\delta(\mathbf{x}_i^* - \mathbf{x}_j^*)\}_{i,j=1}^{n^*}$ is the matrix of covariances between values $\mathbf{y}(\mathbf{X}^*)$.

3 Variable Fidelity Gaussian Process Regression

Now we consider the case of variable fidelity data: there are a sample of the low fidelity function $D_l = (\mathbf{X}_l, \mathbf{y}_l) = \left\{\mathbf{x}_i^l, y_l(\mathbf{x}_i^l)\right\}_{i=1}^{n_l}$ and a sample of the high fidelity function $D_h = (\mathbf{X}_h, \mathbf{y}_h) = \left\{\mathbf{x}_i^h, y_h(\mathbf{x}_i^h)\right\}_{i=1}^{n_h}$ with $\mathbf{x}_i^l, \mathbf{x}_i^h \in \mathbb{R}^d$, $y_l(\mathbf{x}), y_h(\mathbf{x}) \in \mathbb{R}$. The low fidelity function $y_l(\mathbf{x})$ and the high fidelity function $y_h(\mathbf{x})$ model the same physical phenomenon, but with different fidelities.

With the use of samples of the low and the high fidelity functions our aim is to construct, as accurately as possible, a surrogate model $\hat{y}_h(\mathbf{x}) \approx y_h(\mathbf{x})$ of the high fidelity function; moreover, we also need an uncertainty estimate of the prediction.

If data come from two sources of different fidelities, then an appropriate model should be used. We assume that the following variable fidelity data model holds true [16]:

$$y_l(\mathbf{x}) = f_l(\mathbf{x}) + \varepsilon_l,\ y_h(\mathbf{x}) = \rho y_l(\mathbf{x}) + y_d(\mathbf{x}),$$

where $y_d(\mathbf{x}) = f_d(\mathbf{x}) + \varepsilon_d$. $f_l(\mathbf{x})$, $f_d(\mathbf{x})$ are realizations of independent Gaussian processes with zero means and covariance functions $k_l(\mathbf{x}, \mathbf{x}')$ and $k_d(\mathbf{x}, \mathbf{x}')$, respectively, and ε_l, ε_d are Gaussian white noise processes with variances σ_l^2 and σ_d^2, respectively. We also set $\mathbf{X} = \begin{pmatrix}\mathbf{X}_l \\ \mathbf{X}_h\end{pmatrix}$, $\mathbf{y} = \begin{pmatrix}\mathbf{y}_l \\ \mathbf{y}_h\end{pmatrix}$. Then the posterior mean of the high-fidelity values at new points has the form

$$\hat{\mathbf{y}}_h(\mathbf{X}^*) = \mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{K}^{-1}\mathbf{y}, \tag{4}$$

where

$$\mathbf{K}(\mathbf{X}^*, \mathbf{X}) = \left(\rho\mathbf{K}_l(\mathbf{X}^*, \mathbf{X}_l)\ \rho^2\mathbf{K}_l(\mathbf{X}^*, \mathbf{X}_h) + \mathbf{K}_d(\mathbf{X}^*, \mathbf{X}_h)\right),$$
$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{pmatrix}\mathbf{K}_l(\mathbf{X}_l, \mathbf{X}_l) & \rho\mathbf{K}_l(\mathbf{X}_l, \mathbf{X}_h) \\ \rho\mathbf{K}_l(\mathbf{X}_h, \mathbf{X}_l)\ \rho^2\mathbf{K}_l(\mathbf{X}_h, \mathbf{X}_h) + \mathbf{K}_d(\mathbf{X}_h, \mathbf{X}_h)\end{pmatrix},$$

$\mathbf{K}_l(\mathbf{X}_a, \mathbf{X}_b)$, $\mathbf{K}_d(\mathbf{X}_a, \mathbf{X}_b)$ are the matrices of pairwise covariances for the Gaussian processes $y_l(\mathbf{x})$ and $y_d(\mathbf{x})$ for points from some samples $\mathbf{X}_a$ and $\mathbf{X}_b$, respectively. The posterior covariance matrix is as follows:

$$\mathbb{V}(\mathbf{X}^*) = \rho^2\mathbf{K}_l(\mathbf{X}^*, \mathbf{X}^*) + \mathbf{K}_d(\mathbf{X}^*, \mathbf{X}^*) - \mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{K}^{-1}\left(\mathbf{K}(\mathbf{X}^*, \mathbf{X})\right)^T. \tag{5}$$

To estimate covariance function parameters and noise variances for Gaussian processes $f_l(\mathbf{x})$ and $f_d(\mathbf{x})$ we use the following common algorithm [16]:

1. Estimate the parameters of the covariance function $k_l(\mathbf{x}, \mathbf{x})$ using the algorithm from Sect. 2 with sample $D = D_l$,
2. Calculate the posterior mean estimates $\hat{y}_l(\mathbf{x})$ of the Gaussian process $y_l(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}_h$,
3. Estimate the parameters of the Gaussian process $y_d(\mathbf{x})$ with the covariance function $k_d(\mathbf{x}, \mathbf{x}')$ and parameter ρ by maximizing likelihood (1) with $D = D_{\text{diff}} = (\mathbf{X}_h, \mathbf{y}_d = \mathbf{y}_h - \rho\hat{\mathbf{y}}_l(\mathbf{X}_h))$ and $k(\mathbf{x}, \mathbf{x}') = k_d(\mathbf{x}, \mathbf{x}')$.

As we have big enough sample of low fidelity data, we assume that we can get precise estimates of parameters of covariance function $k_l(\mathbf{x}, \mathbf{x})$, so we don't need to refine these estimates using high fidelity data.

4 Sparse Gaussian Process Regression

To perform inference for Variable Fidelity Gaussian process regression we have to invert the sample covariance matrix of size $n \times n$, where $n = n_h + n_l$. This operation is of complexity $O(n^3)$, so for samples of sizes larger than few thousands points we cannot construct a Gaussian process regression in a reasonable time.

In order to construct a Gaussian process regression for large sample sizes we propose to use an approximation to the exact inference. The Nyström approximation [18] of all involved matrices $\mathbf{K}(\mathbf{X}^*, \mathbf{X})$, $\mathbf{K}$ and $\mathbf{K}(\mathbf{X}^*, \mathbf{X}^*)$ allows one to obtain such an approximation.

Let us select from the initial sample a subsample $\mathbf{X}^1 = \begin{pmatrix} \mathbf{X}_l^1 \\ \mathbf{X}_h^1 \end{pmatrix}, \mathbf{y}^1 = \begin{pmatrix} \mathbf{y}_l(\mathbf{X}_l^1) \\ \mathbf{y}_h(\mathbf{X}_h^1) \end{pmatrix}$ of *base* points with the size $n_1 = n_h^1 + n_l^1$ to be small enough so we can perform an exact inference for it. The simplest, rather robust and efficient way for this is to perform uniform random selection without repetitions among points from the initial samples.

Hence, by definition,

$$\mathbf{K}_{11} = \begin{pmatrix} \mathbf{K}_l(\mathbf{X}_l^1, \mathbf{X}_l^1) & \rho\mathbf{K}_l(\mathbf{X}_l^1, \mathbf{X}_h^1) \\ \rho\mathbf{K}_l(\mathbf{X}_h^1, \mathbf{X}_l^1) & \rho^2\mathbf{K}_l(\mathbf{X}_h^1, \mathbf{X}_h^1) + \mathbf{K}_d(\mathbf{X}_h^1, \mathbf{X}_h^1) \end{pmatrix},$$
$$\mathbf{K}_1 = \begin{pmatrix} \mathbf{K}_l(\mathbf{X}_l^1, \mathbf{X}_l) & \rho\mathbf{K}_l(\mathbf{X}_l^1, \mathbf{X}_h) \\ \rho\mathbf{K}_l(\mathbf{X}_h^1, \mathbf{X}_l) & \rho^2\mathbf{K}_l(\mathbf{X}_h^1, \mathbf{X}_h) + \mathbf{K}_d(\mathbf{X}_h^1, \mathbf{X}_h) \end{pmatrix},$$
$$\mathbf{K}_1^* = \left(\rho\mathbf{K}_l(\mathbf{X}^*, \mathbf{X}_l^1)\ \rho^2\mathbf{K}_l(\mathbf{X}^*, \mathbf{X}_h^1) + \mathbf{K}_d(\mathbf{X}^*, \mathbf{X}_h^1)\right)$$

for some new points $\mathbf{X}^* = \{\mathbf{x}_i^*\}_{i=1}^{n^*}$ and so using the Nyström approximation we get approximations of the matrices $\mathbf{K}(\mathbf{X}^*, \mathbf{X})$, $\mathbf{K}$ and $\mathbf{K}(\mathbf{X}^*, \mathbf{X}^*)$, respectively:

$$\hat{\mathbf{K}}(\mathbf{X}^*, \mathbf{X}) = \mathbf{K}_1^*\mathbf{K}_{11}^{-1}\mathbf{K}_1,\ \hat{\mathbf{K}} = (\mathbf{K}_1)^T\mathbf{K}_{11}^{-1}\mathbf{K}_1,\ \hat{\mathbf{K}}(\mathbf{X}^*, \mathbf{X}^*) = \mathbf{K}_1^*\mathbf{K}_{11}^{-1}(\mathbf{K}_1^*)^T.$$

We set

$$\mathbf{R} = \begin{pmatrix} \frac{1}{\sigma_l}\mathbf{I}_{n_l} & 0 \\ 0 & \frac{1}{\sqrt{\rho^2\sigma_l^2 + \sigma_d^2}}\mathbf{I}_{n_h} \end{pmatrix},$$

where $\mathbf{I}_k$ is the identity matrix of size k, $\mathbf{C}_1 = \mathbf{R}\mathbf{K}_1$ and $\mathbf{V} = \mathbf{C}_1\mathbf{V}_{11}^{-T}$, $\mathbf{V}_{11}$ is the Cholesky decomposition of $\mathbf{K}_{11}$.

Theorem 1. *For the posterior mean and the posterior covariance matrix the following Nystrom approximations hold*

$$\hat{\mathbf{y}}_h^{Ny}(\mathbf{X}^*) = \mathbf{K}_1^*\mathbf{V}_{11}^{-1}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{R}\mathbf{y}, \tag{6}$$
$$\mathbb{V}^{Ny}(\mathbf{X}^*) = \mathbf{K}_1^*\mathbf{V}_{11}^{-1}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}_{11}^{-T}\mathbf{K}_1^{*T} + (\rho^2\sigma_l^2 + \sigma_d^2)\mathbf{I}_{n^*}. \tag{7}$$

Theorem 2. *The computational complexities of the posterior mean and the posterior covariance matrix calculation using (6) and (7) at one point are $O(nn_1^2)$.*

Proof of these theorems are in Appendix A.

5 Gaussian Process Regression for Multifidelity Data with Blackbox for Low Fidelity Function

Suppose that we have a blackbox for the low fidelity function $y_l(\mathbf{x})$; i.e., the blackbox estimates the low fidelity function value at any point from the design space $\mathbb{X} \subseteq \mathbb{R}^d$ on the fly. Let us assume that we have already constructed a Variable fidelity Gaussian processes surrogate model and can calculate predictions using (4) and (5). We can't use huge sample of low fidelity function values at corresponding points due to typical computational limitations for Gaussian process regression. Instead, in order to improve an accuracy of these predictions we can update the posterior mean and the posterior variance of $y_h(\mathbf{x})$ at a new point $\mathbf{x}$ with the low fidelity function value $y_l(\mathbf{x})$ at this point, as calculated by the blackbox. Let us describe a computationally efficient procedure to calculate the update.

We set

$$\mathbf{k}_l(\mathbf{x}, \mathbf{X}) = \begin{pmatrix} \mathbf{K}_l(\mathbf{x}, \mathbf{X}_l) \\ \rho \mathbf{K}_l(\mathbf{x}, \mathbf{X}_h) \end{pmatrix},$$

where $\mathbf{x}$ is a some new point. For a sample with an additional point $\mathbf{x}$ included we get an expanded covariance matrix:

$$\mathbf{K}_{\text{exp}} = \begin{pmatrix} \mathbf{K} & \mathbf{k}_l \\ \mathbf{k}_l^T & k_l(\mathbf{x}, \mathbf{x}) \end{pmatrix}.$$

Suppose we know the Cholesky decompositions $\mathbf{L}$ and $\mathbf{L}^{-1}$ of the initial training sample covariance matrix $\mathbf{K}$ and its inverse $\mathbf{K}^{-1}$, respectively. To calculate the posterior mean and the posterior variance for the expanded model we will update these Cholesky decompositions and then update the posterior mean and the posterior variance values.

If we have an $n \times n$ matrix $\mathbf{K}_n$ and the Cholesky decomposition of it, we can get the updated Cholesky decomposition of the matrix $\mathbf{K}_{n+1}$ of size $(n+1) \times (n+1)$ if the initial matrix is in the upper left corner of the new matrix $\mathbf{K}_{n+1}$ with computational complexity $O(n^2)$ using a common routine [20]. To update inverse of the Cholesky decomposition we also need $O(n^2)$ operations, as it differs from the initial Cholesky decomposition only in the last row and is lower triangular. Therefore, we can calculate the matrix $\mathbf{K}_{\text{exp}}^{-1}$ in $O(n^2)$ operations.

The expanded vector of covariances between the new point $\mathbf{x}$ and the initial training sample has the form

$$\mathbf{k}_{\text{exp}} = \begin{pmatrix} \rho \mathbf{K}_l(\mathbf{x}, \mathbf{X}_l) \\ \rho^2 \mathbf{K}_l(\mathbf{x}, \mathbf{X}_h) + \mathbf{K}_d(\mathbf{x}, \mathbf{X}_h) \\ \rho k_l(\mathbf{x}, \mathbf{x}) \end{pmatrix}.$$

Using the value $y_l(\mathbf{x})$ calculated by the blackbox, we set $\mathbf{y}_{\text{exp}} = \left(\mathbf{y}^T, y_l(\mathbf{x})\right)^T$. Then the updated expressions for the posterior mean and the posterior variance are as follows:

$$\hat{y}_h^{\text{exp}}(\mathbf{x}) = \mathbf{k}_{\text{exp}} \mathbf{K}_{\text{exp}}^{-1} \mathbf{y}_{\text{exp}}, \tag{8}$$

$$\mathbb{V}_{\text{exp}}(\mathbf{x}) = \rho^2 \mathbf{K}_l(\mathbf{x}, \mathbf{x}) + \mathbf{K}_d(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\text{exp}}^T \mathbf{K}_{\text{exp}}^{-1} \mathbf{k}_{\text{exp}}. \tag{9}$$

As the Cholesky decomposition for the updated model differs only in the last row we can calculate (8) and (9) in $O(n^2)$ operations.

The total computational complexity is the sum of the computational complexities of the Cholesky decomposition update and the posterior mean and the posterior variance recalculation, so for a Variable fidelity Gaussian process regression with a blackbox, representing the low fidelity function, the following assertions holds.

Theorem 3. *Suppose we know the Cholesky decompositions* $\mathbf{L}$ *and* $\mathbf{L}^{-1}$ *of the initial training sample covariance matrix* $\mathbf{K}$ *and its inverse* $\mathbf{K}^{-1}$*, respectively. Then we can calculate the posterior mean* $\hat{y}_h^{\exp}(\mathbf{x})$ *via* (8) *and the variance* $\mathbb{V}_{\exp}(\mathbf{x})$ *via* (9) *in* $O(n^2)$ *operations, where* $n = n_l + n_h$*.*

As we add only one point to the initial training sample, we expect that estimate of parameters of Gaussian processes model remains accurate enough. While it can be reasonable to add many points in some cases, this issue raises the complex question on how and when we should re-estimate Gaussian processes parameters as we add more points. Using blackbox for the low fidelity function we can get significantly more accurate approximation with small additional computational cost.

6 Numerical Examples

In this section we consider several problems: two artificial problems and a real applied problem of surrogate model construction for a rotating disk from an aircraft engine. We compare the four approaches below for a surrogate model construction; two latter approaches are introduced above:

- GP — Gaussian Process Regression using only high fidelity data,
- VFGP — Variable Fidelity Gaussian Process Regression using high and low fidelity data,
- SVFGP — Sparse VFGP, which is a version of VFGP for the case of large training samples introduced in Sect. 4,
- BB VFGP — VFGP with the low fidelity function realized by a black box introduced in Sect. 5. In experiments we use the same design of experiments as in case of VFGP, while for model update for each new point we use low fidelity function value at this point.

As a covariance function for a Gaussian process regression we use the multivariate squared exponential covariance function, see [37]. To regularize the problem and avoid inversion of large ill-conditioned matrices, we impose a prior distribution of nugget term in Bayesian way [7], so we are sure that for all four approaches we avoid problems with poor estimation of parameters for Gaussian Processes for large samples due to computational issues (linked with small values of regularization parameter σ^2 (nugget effect) [31,34]). To estimate parameters in SVFGP we use only a selected subsample of points, while we use the full sample to predict values at new points.

To measure the accuracy of the obtained surrogate models we use an RRMS error estimated by k-fold cross-validation procedure [23] if not specified otherwise. Note that we use low fidelity point for training only if the same point doesn't belong to the selected high fidelity test design if not specified otherwise. For a single target variable and a test sample $D_{\text{test}} = \{\mathbf{x}_i^{\text{test}}, y_i^{\text{test}} = f_h(\mathbf{x}_i^{\text{test}})\}_{i=1}^{n_t}$ the RRMS error for a surrogate model $\hat{y}(\mathbf{x})$ equals to

$$RRMS(D_{\text{test}}, \hat{y}) = \sqrt{\frac{\sum_{i=1}^{n_t}(\hat{y}_h(\mathbf{x}_i^{\text{test}}) - y_i^{\text{test}})^2}{\sum_{i=1}^{n_t}(\overline{y} - y_i^{\text{test}})^2}},$$

here $\overline{y} = \frac{1}{n_t}\sum_{i=1}^{n_t} y_i^{\text{test}}$. The value of the RRMS error typically lies between 0 and 1. Accurate models have RRMS values close to 0, while inaccurate models have RRMS values close to or greater than 1.

6.1 Artificial Problem with Big Sample Size

To benchmark proposed approaches we use an artificial function with multiple local peculiarities and input dimension $d = 6$, so we really need rather big sample to get an accurate surrogate model. As a high fidelity function $y_h(\mathbf{x})$ and a low fidelity function $y_l(\mathbf{x})$ we use

$$y_h(\mathbf{x}) = 20 + \sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i)) + \varepsilon_h,\ \mathbf{x} \in [0,1]^d,$$

$$y_l(\mathbf{x}) = y_h(\mathbf{x}) + 0.2\sum_{i=1}^{d}(x_i + 1)^2 + \varepsilon_l,\ \mathbf{x} \in [0,1]^d.$$

The high fidelity function was corrupted by a Gaussian white noise ε_h with variance 0.001, and the low fidelity function was corrupted by a Gaussian white noise ε_l with variance 0.002. When preparing samples for experiments we generate points in $[0,1]^d$ using Latin Hypercube Sampling [32]. To test extrapolation properties we limit training sample points to the region with range $[0, 0.5]$ instead of $[0, 1]$ for one of 6 input variables. The high fidelity sample size was $n_h = 100$ and the size of the subsample for SVFGP was $n_l^1 = 1000$ in all experiments.

The results were averaged over 5 runs for each considered value of n_l. We thus have

- Table 1 contains RRMS errors for VFGP, SVFGP, and BB VFGP,
- Table 2 contains RRMS errors for VFGP, SVFGP, and BB VFGP in case we use the surrogate model in extrapolation regime,
- Table 3 provides training times for VFGP, SVFGP and BB VFGP approach.

One can see that RRMS errors of SVFGP are comparable with RRMS errors of VFGP for the same sample size, while the training time of SVFGP is tremendously smaller when the sample size is equal to 5000, and for SVFGP the training time increases only slightly when the sample size increases. For BB VFGP training time in this experiment coincides with that of VFGP, while for 1000 training

Table 1. Comparison of RRMS errors

n_l	1000	3000	5000
VFGP	0.0502	0.0170	0.0058
SVFGP	0.0502	0.0305	0.0260
BB VFGP	0.0010	0.00029	0.00017

Table 2. Comparison of extrapolation RRMS errors

n_l	1000	3000	5000
VFGP	0.3636	0.1351	0.1028
SVFGP	0.3636	0.3281	0.3586
BB VFGP	0.000998	0.00113	0.00034

Table 3. Comparison of training times in seconds for Ubuntu PC, Intel-Core $i7$ with 4 physical cores, 3.4 GHz, 16 Gb RAM.

n_l	1000	3000	5000
VFGP	30.46	852.70	7283.27
SVFGP	30.46	33.42	37.50
BB VFGP	30.38	842.97	7672.60

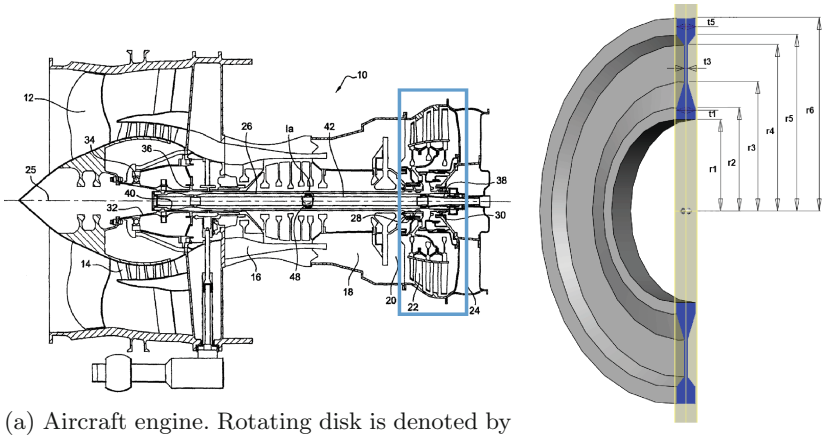
points we get better results with BB VFGP than for 5000 training points and VFGP. If we calculate prediction in extrapolation regime, we get significantly better results with BB VFGP.

6.2 Rotating Disk Problem

Now let us compare the approaches to the construction of surrogate models on a real applied problem of rotating disk surrogate modeling.

Rotating Disk Model Description. A high speed rotating risk is an important part of an aircraft engine (see Fig. 1a), three parameters define quality of the disk: the mass of the disk, the maximal radial displacement $u_{\max}$, the maximal stress $s_{\max}$ [3,6,30]. It is easy to calculate mass of the disk, as we know all geometrical parameters of the disk, while surrogate modeling of the maximal radial displacement and the maximal stress is challenging [26,30]. So the focus here is on modeling of the maximal radial displacement and the maximal stress.

Used parametrization of the rotating disk geometry consists of 8 parameters: the radii r_i, $i = 1, \ldots, 6$, which control where the thickness of the rotating disk changes, and the values t_1, t_3, t_5, which control the corresponding changes in thickness. In the considered surrogate modeling problem we fix the radii r_4, r_5 and the thickness t_3 of a rotating disk, so the input dimension for the surrogate



(a) Aircraft engine. Rotating disk is denoted by the bold rectangle at the right side of the figure.

(b) Rotating disk geometry

Fig. 1. Rotating disk problem

model is 6. The geometry and the parametrization of the rotating disk are shown in Fig. 1b.

There are two available solvers for $u_{\max}$ and $s_{\max}$ calculation. The low fidelity function is calculated using Ordinary Differential Equations (ODE) solver based on a simple Runge–Kutta's method. The high fidelity function is calculated using Finite Element Model (FEM) solver. A single evaluation of the low fidelity function takes $\sim$0.01 s, and a single evaluation of the high fidelity function takes $\sim$300 s. More detailed comparison of solvers is given in Appendix B.

Table 4. RRMS errors for introduced approaches with standard deviations

Output $u_{\max}$				
n_h	20	40	60	80
GP	0.287 ± 0.039	0.143 ± 0.031	0.082 ± 0.020	0.095 ± 0.023
VFGP	0.212 ± 0.075	0.088 ± 0.009	0.064 ± 0.007	0.068 ± 0.006
SVFGP	0.125 ± 0.029	0.074 ± 0.016	0.041 ± 0.007	0.047 ± 0.011
BB VFGP	0.123 ± 0.019	0.053 ± 0.008	0.030 ± 0.007	0.034 ± 0.006
Output $s_{\max}$				
n_h	20	40	60	80
GP	0.505 ± 0.10	0.367 ± 0.15	0.251 ± 0.049	0.196 ± 0.014
VFGP	0.363 ± 0.07	0.261 ± 0.06	0.193 ± 0.011	0.123 ± 0.043
SFGP	0.190 ± 0.06	0.122 ± 0.06	0.119 ± 0.015	0.088 ± 0.027
BB VFGP	0.158 ± 0.03	0.162 ± 0.03	0.137 ± 0.024	0.078 ± 0.020

Surrogate Model Accuracy. In this section we compare our approaches via SVFGP (Sparse variable fidelity Gaussian processes) and BB VFGP (Blackbox variable fidelity Gaussian processes) with GP (based only on high fidelity data) and VFGP baseline methods.

We used Latin Hypercube approach to sample points. Low fidelity training sample size was 1000, High fidelity training sample size n_h was 20, 40, 60, and 80 in different experiments. In order to estimate the accuracy of a high fidelity function prediction we used the cross-validation procedure, applied to 140 high fidelity data points (these points contain n_h points used for training of surrogate models). For each fixed sample size n_h we used 5 splits of the data to training and test samples to estimate means and standard deviations. For SVFGP, we use $n_l = 5000$ low fidelity points in total, and randomly select $n_l^1 = 1000$ points from them as base points.

The results are given in Table 4 for $u_{\max}$ and $s_{\max}$ outputs: VFGP outperforms GP, and both SVFGP and BB VFGP outperform VFGP in terms of RRMS error. Therefore, we decide which one to use, SVFGP or BB VFGP, by taking into account whether the blackbox for low fidelity function during a surrogate model usage is available, or whether one uses the surrogate model in extrapolation regime, etc.

6.3 Optimization of Rotating Disk Shape

We optimize the shape of the rotating disk described:

$$\begin{aligned} m, u_{\max} \rightarrow & \min_{r_1,\ldots,r_6,t_1,t_3,t_5}, \\ u_{\max} \leq 0.3, & s_{max} \leq 600, \\ 10 \leq r_1 \leq 110, & 120 \leq r_2 \leq 140, \\ 150 \leq r_3 \leq 168, & 170 \leq r_4 \leq 200, \\ 4 \leq t_1 \leq 50, & 4 \leq t_3 \leq 50, \\ r_5 = 210, & r_6 = 230, t_5 = 32. \end{aligned} \tag{10}$$

The presented problem has multiple objectives, and we are looking for a Pareto frontier, not a single point.

Single optimization run is the following:

- Generate initial high fidelity sample D_h of 30 points using the Latin Hypercube sampling.
- Construct surrogate models using GP, VFGP, SVFGP and BB VFGP approaches using the generated high fidelity sample D_h and low fidelity sample D_l of size 1000 for GP, VFGP and BB VFGP and of size 5000 for SVFGP.
- Solve multiobjective optimization problem at hand using these surrogate models as the target functions and constraints.
- Calculate true values at Pareto frontiers obtained during optimization using high fidelity solver to estimate quality of models.

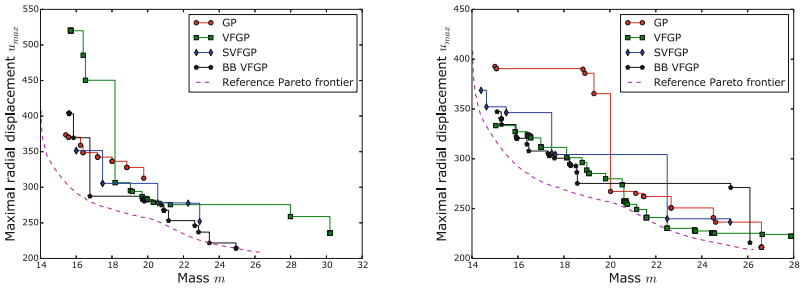


Fig. 2. Pareto frontiers obtained using optimization of surrogate models constructed with GP, VFGP, SVFGP and BB VFGP approaches along with the reference Pareto frontier

Table 5. Optimization results for different surrogate models along with minimal values for different optimization objectives. Also we present proportion of feasible points in the final Pareto frontier. The best values are in bold font.

Objective	GP	VFGP	SVFGP	BB VFGP
m	16.62	15.69	**15.09**	15.63
$0.8m + 0.2u_{\max}$	73.65	70.74	70.71	**68.10**
$0.6m + 0.4u_{\max}$	125.10	117.37	116.21	**112.55**
$0.4m + 0.6u_{\max}$	176.55	163.89	161.18	**156.99**
$0.2m + 0.8u_{\max}$	228.00	210.33	206.12	**201.44**
$u_{\max}$	279.44	256.77	251.05	**245.89**
Feasible points share	0.54	0.57	0.55	**0.75**

Due to properties of applied optimization algorithm sizes of Pareto frontiers can slightly differ for different runs of optimization algorithm, with mean size of Pareto frontier about 30 points [14]. So we need about 50 runs of high fidelity function to solve this optimization problem. In order to recover a reference Pareto frontier we constructed an accurate surrogate model using 5000 high fidelity points from uniform design over all the design space and additional sampling in a region where Pareto frontier points are located. So, instead of using a solver to evaluate an original function during optimization runs we used this surrogate model.

The examples of obtained Pareto frontiers for a single optimization run is in Fig. 2. For these runs SVFGP and BB VFGP work better than GP and VFGP.

Results of optimization are in Table 5. We compare minimum values of different weighted sums of two target variables m and $u_{\max}$ averaged over 10 runs of optimization for different initial samples. We obtain the best value of mass m output using SVFGP algorithm and the best value of $u_{\max}$ using BB VFGP algorithm while optimizations based on GP and VFGP work worse. Also, with BB VFGP we produce significantly larger amount of feasible points compared to

GP, VFGP and SVFGP, which typically leads to better Pareto frontier coverage with similar number of high fidelity blackbox runs.

7 Conclusions

We presented two new approaches to variable fidelity surrogate modeling, which allow one to perform large sample inference for Variable Fidelity Gaussian process regression: the first approach approximates the full covariance matrix of the sample and its inverse, the second approach uses the available low fidelity black box to update the surrogate model with the low fidelity function value at the point where one wants to estimate the high fidelity function thus avoiding requirement to use large low fidelity sample. Using developed approaches we can perform large sample inference for variable fidelity Gaussian process regression and construct more accurate surrogate models.

Acknowledgments. We thank Dmitry Khominich from DATADVANCE llc for making the solvers for rotating disk problem available, and Tatyana Alenkaya from MIPT for proofreading of the article. The research was conducted in IITP RAS and supported solely by the Russian Science Foundation grant (project 14-50-00150).

Appendix

A Proof of Technical Statements

In this section we provide the proofs of the statements of Sect. 4.

Proof (Proof of Statement 1). For the posterior mean we get:

$$\begin{aligned}\hat{\mathbf{y}}_h(\mathbf{x}^*) &\approx \mathbf{K}_1^*\mathbf{K}_{11}^{-1}\mathbf{K}_1^T(\mathbf{K}_1\mathbf{K}_{11}^{-1}\mathbf{K}_1^T + \mathbf{R}^{-2})^{-1}\mathbf{y} = \mathbf{K}_1^*\mathbf{K}_{11}^{-1}\mathbf{K}_1^T\mathbf{R}(\mathbf{R}\mathbf{K}_1\mathbf{K}_{11}^{-1}\mathbf{K}_1^T\mathbf{R} + \mathbf{I}_n)^{-1}\mathbf{R}\mathbf{y} = \\ &= \mathbf{K}_1^*\mathbf{K}_{11}^{-1}\mathbf{C}_1^T(\mathbf{C}_1\mathbf{K}_{11}^{-1}\mathbf{C}_1^T + \mathbf{I}_n)^{-1}\mathbf{R}\mathbf{y} = \mathbf{K}_1^*\mathbf{K}_{11}^{-1}(\mathbf{C}_1^T\mathbf{C}_1\mathbf{K}_{11}^{-1} + \mathbf{I}_{n_1})^{-1}\mathbf{C}_1^T\mathbf{R}\mathbf{y} = \\ &= \mathbf{K}_1^*(\mathbf{C}_1^T\mathbf{C}_1 + \mathbf{K}_{11})^{-1}\mathbf{C}_1^T\mathbf{R}\mathbf{y} = \mathbf{K}_1^*(\mathbf{C}_1^T\mathbf{C}_1 + \mathbf{V}_{11}^T\mathbf{V}_{11})^{-1}\mathbf{C}_1^T\mathbf{R}\mathbf{y} = \\ &= \mathbf{K}_1^*\mathbf{V}_{11}^{-1}(\mathbf{V}_{11}^{-T}\mathbf{C}_1^T\mathbf{C}_1\mathbf{V}_{11}^{-1} + \mathbf{I}_{n_1})^{-1}\mathbf{V}_{11}^{-T}\mathbf{C}_1^T\mathbf{R}\mathbf{y} = \mathbf{K}_1^*\mathbf{V}_{11}^{-1}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{R}\mathbf{y}.\end{aligned}$$

We use the same approach to derive an equation for the posterior variance:

$$\begin{aligned}\mathbb{V}\left(X^*\right) - (\rho^2\sigma_l^2 + \sigma_d^2)\mathbf{I}_{n^*} &\approx \mathbf{K}_1^*\mathbf{K}_{11}^{-1}\mathbf{K}_1^{*T} - \mathbf{K}_1^*\mathbf{K}_{11}^{-1}\mathbf{K}_1^T(\mathbf{R}^{-2} + \mathbf{K}_1\mathbf{K}_{11}^{-1}\mathbf{K}_1^T)^{-1}\mathbf{K}_1\mathbf{K}_{11}^{-1}\mathbf{K}_1^{*T} = \\ &= \mathbf{K}_1^*(\mathbf{K}_{11}^{-1} - \mathbf{K}_{11}^{-1}\mathbf{K}_1^T(\mathbf{R}^{-2} + \mathbf{K}_1\mathbf{K}_{11}^{-1}\mathbf{K}_1^T)^{-1}\mathbf{K}_1\mathbf{K}_{11}^{-1})\mathbf{K}_1^{*T} = \\ &= \mathbf{K}_1^*(\mathbf{K}_{11} + \mathbf{K}_1^T\mathbf{R}^2\mathbf{K}_1)^{-1}\mathbf{K}_1^{*T} = \mathbf{K}_1^*(\mathbf{V}_{11}^T\mathbf{V}_{11} + \mathbf{C}_1^T\mathbf{C}_1)^{-1}\mathbf{K}_1^{*T} = \\ &= \mathbf{K}_1^*\mathbf{V}_{11}^{-1}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}_{11}^{-T}\mathbf{K}_1^{*T}.\end{aligned}$$

Proof (Proof of Statement 2). First of all we have to calculate the matrices $\mathbf{V}_{11}$ and $\mathbf{V} = \mathbf{R}\mathbf{K}_1\mathbf{V}_{11}^{-T}$. The matrix $\mathbf{V}_{11}$ is of size $n_1 \times n_1$, so we need $O(n_1^3)$ to get its inverse. To calculate $\mathbf{K}_1\mathbf{V}_{11}^{-T}$ we need $O(n_1^2 n)$ operations. Finally, as $\mathbf{R}$ is a diagonal matrix, we use $O(n_1 n)$ operations to get $\mathbf{V}$.

In case $n^* = 1$ to get the posterior mean we have to calculate $\mathbf{V}_{11}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{y}$. We use $O(n_1^2 n)$ operations to calculate $\mathbf{V}^T\mathbf{V}$, to inverse $\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V}$ we need $O(n_1^3)$ operations, to calculate $\mathbf{V}_{11}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ one uses extra $O(n_1^2 n)$ operations, and finally to calculate the posterior mean we need additional $O(n_1 n)$ operations. Consequently, to calculate the posterior mean we use $O(n_1^2 n)$ operations.

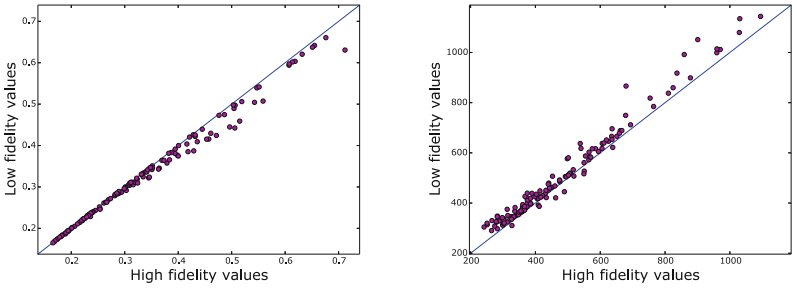
In the same way in order to calculate $\mathbf{V}_{11}(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}_{11}^{-1}$ we need $O(n_1^2 n)$ operations to calculate $(\mathbf{I}_{n_1} + \mathbf{V}^T\mathbf{V})^{-1}$ and additional $O(n_1^3)$ operations to get the final matrix. Consequently, in order to calculate the posterior variance we use $O(n_1^2 n)$ operations.

Finally, we need $O(n_1^2 n)$ operations to compute the required matrices, and $O(n_1^2 n)$, to obtain the posterior mean and the posterior variance from these precomputed matrices. So, the total computational complexity is $O(n_1^2 n)$.

B Comparison of Low and High Fidelity Model for Rotating Disk

There are two available solvers for $u_{\max}$ and $s_{\max}$ calculation. The low fidelity function is calculated using Ordinary Differential Equations (ODE) solver based on a simple Runge–Kutta's method. The high fidelity function is calculated using Finite Element Model (FEM) solver from ANSYS.

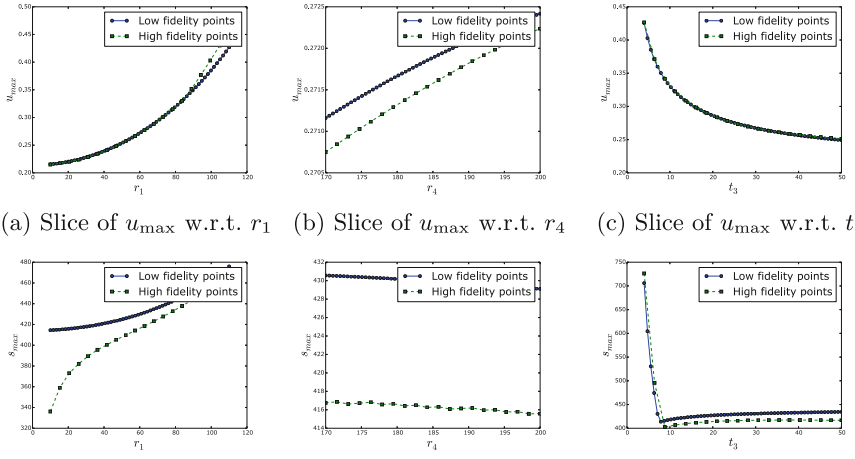
To compare the solvers we draw the scatter plots of low and high fidelity values and also plot slices of the corresponding functions. We generate a random sample of points in a specified design space box, calculate the low and high fidelity function values and draw the low fidelity function values versus the high fidelity function values at the same points. The scatter plots are in Fig. 3: the difference between values increases significantly when the values are increasing.



(a) Scatter plot for $u_{\max}$ (b) Scatter plot for $s_{\max}$

Fig. 3. Comparison of the high and the low fidelity solvers via scatter plots

For the central point of the design space box with $r_1 = 0.06, r_2 = 0.13, r_3 = 0.16, r_4 = 0.185, t_1 = 0.027, t_3 = 0.027$ we construct one-dimensional slices by



(a) Slice of $u_{\max}$ w.r.t. r_1 (b) Slice of $u_{\max}$ w.r.t. r_4 (c) Slice of $u_{\max}$ w.r.t. t_3

(d) Slice of $s_{\max}$ w.r.t. r_1 (e) Slice of $s_{\max}$ w.r.t. r_4 (f) Slice of $s_{\max}$ w.r.t. t_3

Fig. 4. Comparison of the high and the low fidelity solvers via outputs' slices

varying single input variable in specified bounds. Slices for different input variables for $u_{\max}$ and for $s_{\max}$ are given in Fig. 4. In case of $u_{\max}$ the high and the low fidelity functions have the same behaviour, and the low fidelity function models the high fidelity function accurately. For $s_{\max}$ the high and the low fidelity functions are sometimes different: their behaviours differ for a slice along r_1 input, and local maxima differ for slice along t_3 input.

References

1. Alexandrov, N.M., Nielsen, E.J., Lewis, R.M., Anderson, W.K.: First-order model management with variable-fidelity physics applied to multi-element airfoil optimization. Technical report, NASA (2000)
2. Álvarez, M.A., Lawrence, N.D.: Computationally efficient convolved multiple output Gaussian processes. J. Mach. Learn. Res. **12**, 1425–1466 (2011)
3. Armand, S.C.: Structural optimization methodology for rotating disks of aircraft engines. Technical report, National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Program (1995)
4. Bachoc, F.: Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. Comput. Stat. Data Anal. **66**, 55–69 (2013)
5. Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H.: Gaussian predictive process models for large spatial data sets. J. Royal Stat. Soc. Ser. B (Statist. Method.) **70**(4), 825–848 (2008)
6. Belyaev, M., Burnaev, E., Kapushev, Y.: Gaussian process regression for structured data sets. In: Gammerman, A., Vovk, V., Papadopoulos, H. (eds.) SLDS 2015. LNCS, vol. 9047, pp. 106–115. Springer, Heidelberg (2015)
7. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)

8. Boyle, P., Frean, M.: Dependent Gaussian processes. Adv. Neural Inf. Process. Syst. **17**, 217–224 (2005)
9. Burnaev, E., Panov, M.: Adaptive design of experiments based on Gaussian processes. In: Gammerman, A., Vovk, V., Papadopoulos, H. (eds.) SLDS 2015. LNCS, vol. 9047, pp. 116–125. Springer, Heidelberg (2015)
10. Burnaev, E.V., Zaytsev, A.A., Spokoiny, V.G.: The Bernstein-von Mises theorem for regression based on Gaussian processes. Russ. Math. Surv. **68**(5), 954–956 (2013)
11. Chang, W., Haran, M., Olson, R., Keller, K., et al.: Fast dimension-reduced climate model calibration and the effect of data aggregation. Ann. Appl. Stat. **8**(2), 649–673 (2014)
12. Doyen, P.: Porosity from seismic data: a geostatistical approach. Geophysics **53**(10), 1263–1275 (1988)
13. Drineas, P., Mahoney, M.W.: On the Nyström method for approximating a Gram matrix for improved kernel-based learning. J. Mach. Learn. Res. **6**, 2153–2175 (2005)
14. Druot, T., Alestra, S., Brand, C., Morozov, S.: Multi-objective optimization of aircrafts family at conceptual design stage. In: Design and Optimization Symposium. Albi, France, In Inverse Problems (2013)
15. Farshi, B., Jahed, H., Mehrabian, A.: Optimum design of inhomogeneous non-uniform rotating discs. Comput. Struct. **82**(9), 773–779 (2004)
16. Forrester, A.I.J., Sóbester, A., Keane, A.J.: Multi-fidelity optimization via surrogate modelling. Proc. Roy. Soc. A Math. Phys. Eng. Sci. **463**(2088), 3251–3269 (2007)
17. Forrester, A.I.J., Sóbester, A., Keane, A.J.: Engineering Design Via Surrogate Modelling: a Practical Guide. J. Wiley, Chichester (2008)
18. Foster, L., Waagen, A., Aijaz, N., Hurley, M., Luis, A., Rinsky, J., Satyavolu, C., Way, M.J., Gazis, P., Srivastava, A.: Stable and efficient Gaussian process calculations. J. Mach. Learn. Res. **10**, 857–882 (2009)
19. Furrer, R., Genton, M.G., Nychka, D.: Covariance tapering for interpolation of large spatial datasets. J. Comput. Graphical Stat. **15**(3), 502–523 (2006)
20. Golub, G.H., Van Loan, C.F.: Matrix Computations, vol. 3. JHU Press, Baltimore (2012)
21. Grihon, S., Burnaev, E., Belyaev, M., Prikhodko, P.: Surrogate modeling of stability constraints for optimization of composite structures. In: Koziel, S., Leifsson, L. (eds.) Surrogate-Based Modeling and Optimization, pp. 359–391. Springer, New York (2013)
22. Han, Z., Görtz, S., Zimmermann, R.: Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. Aerosp. Sci. Technol. **25**(1), 177–189 (2013)
23. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. Math. Intell. **27**(2), 83–85 (2005)
24. Hensman, J., Fusi, N., Lawrence, N.D.,Gaussian processes for big data. arXiv preprint arXiv: 1309.6835 (2013)
25. Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. J. Am. Stat. Assoc. **103**(482), 570–583 (2008)
26. Huang, Z., Wang, C., Chen, J., Tian, H.: Optimal design of aeroengine turbine disc based on kriging surrogate models. Comput. Struct. **89**(1), 27–37 (2011)
27. Kennedy, M.C., O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available. Biometrika **87**(1), 1–13 (2000)

28. Koziel, S., Bekasiewicz, A., Couckuyt, I., Dhaene, T.: Efficient multi-objective simulation-driven antenna design using co-kriging. IEEE Trans. Antennas Propag. **62**(11), 5900–5905 (2014)
29. Madsen, J.I., Langthjem, M.: Multifidelity response surface approximations for the optimum design of diffuser flows. Optim. Eng. **2**(4), 453–468 (2001)
30. Mohan, S.C., Maiti, D.K.: Structural optimization of rotating disk using response surface equation and genetic algorithm. Int. J. Comput. Methods Eng. Sci. Mech. **14**(2), 124–132 (2013)
31. Neal, R.M.: Monte carlo implementation of Gaussian process models for Bayesian regression and classification. arXiv preprint physics/9701026 (1997)
32. Park, J.-S.: Optimal Latin-hypercube designs for computer experiments. J. Stat. Plann. Infer. **39**(1), 95–111 (1994)
33. Park, S., Choi, S.: Hierarchical Gaussian process regression. In: ACML, pp. 95–110 (2010)
34. Pepelyshev, A.: The role of the nugget term in the Gaussian process method. In: Giovagnoli, A., Atkinson, A.C., Torsney, B., May, C. (eds.) mODa 9-Advances in Model-Oriented Design and Analysis, pp. 149–156. Springer, Heidelberg (2010)
35. Qian, Z., Seepersad, C.C., Joseph, V.R., Allen, J.K., Wu, C.F.: Building surrogate models based on detailed and approximate simulations. J. Mech. Des. **128**(4), 668–677 (2006)
36. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate Gaussian process regression. J. Mach. Learn. Res. **6**, 1939–1959 (2005)
37. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. The MIT Press, Cambridge (2006)
38. Shaby, B., Ruppert, D.: Tapered covariance: Bayesian estimation and asymptotics. J. Comput. Graphical Stat. **21**(2), 433–452 (2012)
39. Shi, J.Q., Murray-Smith, R., Titterington, D.M.: Hierarchical Gaussian process mixtures for regression. Stat. Comput. **15**(1), 31–41 (2005)
40. Sun, G., Li, G., Stone, M., Li, Q.: A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials. Comput. Mater. Sci. **49**(3), 500–511 (2010)
41. Sun, S., Zhao, J., Zhu, J.: A review of Nyström methods for large-scale machine learning. Inf. Fusion **26**, 36–48 (2015)
42. Titsias, M.K.: Variational learning of inducing variables in sparse Gaussian processes. In: International Conference on Artificial Intelligence and Statistics, pp. 567–574 (2009)
43. Williams, C.K.I., Barber, D.: Bayesian classification with Gaussian processes. IEEE Trans. Pattern Anal. Mach. Intell. **20**(12), 1342–1351 (1998)
44. Xu, W., Tran, T., Srivastava, R., Journel, A.: Integrating seismic data in reservoir modeling: the collocated cokriging alternative. Society of Petroleum Engineers, In: SPE Annual Technical Conference and Exhibition (1992)
45. Zahir, M.K., Gao, Z.: Variable fidelity surrogate assisted optimization using a suite of low fidelity solvers. Open J. Optim. **1**(1), 0–8 (2012)
46. Zaitsev, A., Burnaev, E., Spokoiny, V.: Properties of the posterior distribution of a regression model based on Gaussian random fields. Autom. Remote Control **74**(10), 1645–1655 (2013)