

Optimal Reduced Order Modelling

Robert A. Milton, Solomon F. Brown

Department of Chemical and Biological Engineering, University of Sheffield, Sheffield, S1 3JD, United Kingdom

Abstract

Keywords: Gaussian Process, Global Sensitivity Analysis, Sobol' Index, Surrogate Model

1. Introduction

A broad range of engineering topics ultimately concern the response of some noisy scalar quantity $y(\mathbf{x})$ to its M -dimensional input \mathbf{x} . It is extremely desirable to reduce M as far as possible without materially impacting y for a number of reasons.

Firstly, $y(\mathbf{x})$ is usually onerous to obtain, from pilot plant, laboratory or simulation. Therefore mitigating the input variables to be controlled and monitored is disproportionately advantageous.

Secondly, visualisation and analysis is far easier and more fruitful in a low dimensional space. There is epistemological value in model-order reduction for its own sake, and to inspire and guide advances in theory or modelling.

Thirdly, refined examination of the response throughout the input space demands an ensemble of results $y(\mathbf{x})$ whose numerosity grows exponentially with M . Worse still, the basic geometry of high-dimensional space contains some nasty surprises, collectively known as the curse of dimensionality. In particular, nearly all the hypervolume of a high-dimensional input space resides proximate to its boundary hypersurface. Taken without prejudice, the input hypervolume consists almost entirely of anomalies, flung to the outer reaches and extremal in one input dimension or other. The curse thus

There is currently no mention of any relevant literature in this, please write this up

Email addresses: `r.a.milton@sheffield.ac.uk` (Robert A. Milton),
`s.f.brown@sheffield.ac.uk` (Solomon F. Brown)

manifests as a kind of engineering paranoia: failure (represented by extremal conditions) becomes almost inevitable as ever more inputs are measured or modelled.

PLACEHOLDER

Variance-based sensitivity analysis by calculation of Sobol' indices [?] is one of the most widely used approaches for global sensitivity analysis (GSA). And with the increased popularity of using GPs as surrogate models, the main disadvantage to the Sobol' method was solved [? ? ?]. The use of GPs provided an alternative method of estimating multidimensional integrals using Monte Carlo schemes. This alternative method was incredibly useful as the previous required a large number of model evaluations, nearly 10'000 are needed to reach 10% precision [?]. GPs are an incredibly useful surrogate modelling technique for sensitivity analysis as they have the great advantage of being analytically tractable and so can calculate the sensitivity indices given a training set of model evaluations. Hence, using GPs will bypass a large number of model evaluations required by Monte Carlo methods. The derivation enabling the use of GP regression for analytical evaluation of variance-based sensitivity indices were first introduced by Jin et al. [?] who applied the Sobol' index formula directly to the GP predictors. Before Oakley and O'Hagan [?] used the global stochastic model of a GP, providing the calculations to produce random variables as a new sensitivity measures. Oakley and O'Hagan's [?] model allows the sensitivity indices accuracy to be analysed due to the distribution of the variables. Marrel et al. [?] took the two methods further by comparing them and building on the work from Oakley and O'Hagan [?] leading to a novel algorithm which builds confidence intervals for the Sobol' indices. Marrel et al. [?] tested both methods on toy functions providing results that show very accurate sensitivity indices and satisfactory confidence intervals from the second method. However, when the approach was illustrated on real data to provide a sensitivity analysis on radionuclide groundwater transport, it was found that the confidence intervals were inaccurate for very low indices due to overestimation of the lowest Sobol' indices. Overall, the work from the three research groups [? ? ?] provided an excellent method of using GP surrogate models for GSA. Consequently the research gives new authors the decision whether the method should calculate the indices using just the GPs prediction means or whether to use the GPs full distribution to produce for confidence intervals on the calculated indices.

Since the derivation of the calculations which allow GPs to be used for

the computation of the Sobol' sensitivity indices, the technique has been used in many disciplines. For example, Rohmer and Foerster [?] used the technique to analyse large-scale numerical landslide models. Due to the limited knowledge of the slip surface in the La Frasse landslide, the simulator was approximated using a small size training data for the GP. Therefore, Rohmer and Foerster inclined to use confidence intervals on the sensitivity measures to outline regions where the GP was unsure. The work concluded by providing ideas of future work by the design of experiments, whereby in the identified unsure regions, further simulator runs should be carried out allowing the GP model to learn further and reduce the confidence intervals.

The method has also been proven to aid in engineering system safety by providing an early validation of health indicators for detection and identification in design sites [?]. The work used a GP to emulate a pumping unit of an aircraft fuel system. Using Latin hypercube sampling, the GP learnt from 400 model evaluations before the Sobol' indices for the 20 inputs were computed. On this occasion, due to a large number of model evaluations and a large number of model inputs, the authors opted to calculate the Sobol' indices alone without any confidence intervals. Interestingly, this work computed both the first-order indices and the total indices, consequently finding very close results. Therefore, this indicates that the inputs have few or no correlations and so the most influential parameter for each health indicator was found.

However, issues may occur when calculating the Sobol' indices of input parameters which are correlated because the construction of such measures rely on the assumption that inputs are independent. This is because if inputs are dependent on one another then the amount of variance due to a given parameter may be influenced by its dependence on another input [?]. Therefore, a lot of work has been conducted towards dealing with dependency in sensitivity analysis studies [? ? ? ?]. However, these methods concentrate solely on the GSA method, failing to incorporate using them with a GP surrogate model. Another issue with the previous studies is that the number of decomposition components exponentially grows with the model dimension [?]. Although, other interesting methods have been invented to deal with correlated inputs which can be used with GPs. For example, Chastaing and Le Gratiet [?] extended the work done by Durrande et al. [?] to deal with dependent inputs. In this work the GP is specified to a special class of ANOVA kernels [?], which is functionally decomposed as a sum of processes indexed by increasing dimension input variables. The work is similar to that

of Caniou and Sudret [?] except it considers the use of GPs instead of the polynomial chaos expansion. Then Li and Rabitz [?] developed another method to decompose the output variance by using a hierarchical orthogonality condition. The result of this reduces the prior formulas to them used for independent variables. Interestingly, Srivastava et al. [?] investigated the Li-Rabitz framework, comparing it to the original variance-based method using GPs. The research used both methods in an attempt to understand which calibration parameters can be fixed without losing output variability for an aircraft model. Consisting of 100 calibration parameters which had both correlated inputs and independent inputs, the problem had over 4000 two-way interactions, ensuring the complexity high enough to not be able to calculate all the two-way interaction indices. Therefore, the work showed the complexities involved in real-world applications, inferring that although separate levels of interactions between inputs can be calculated with Sobol' indices, it is not always possible to separate the sensitivities into structural and correlation elements. However, when the alternative method of approximations was used, the computed sensitivities had inaccuracies even though the relative ordering of variables were still correct. Therefore, it is important to understand that whichever method used, care must be taken when inputs are correlated.

Reduced order modelling concerns reducing the dimensions d of a d -dimensional input \mathbf{x} while preserving the behaviour of the output y . The result of this reduces the inputs to a set which is both mutually independent and highly relevant to the response. An exciting and novel method of dimensionality reduction created by Milton and Brown [ref] achieves the reduced order model (ROM) by an optimal rotation of the input basis. The ROM is achieved through the utilization of a GP surrogate model that facilitates GSA via Sobol' indices [?]. Simply, the method rotates the input basis so that the emulator only significantly depends on the most relevant rotated input dimensions.

However, the mathematics behind the analysis is difficult and so in this report, a very brief account will be given. For a full derivation please refer to the paper by Milton and Brown [ref].

The overall goal is to replace the input matrix \mathbf{X} of size $(d \times n)$ with a rotated input basis \mathbf{U} of size $(d \times n)$. When looking at a sample datum \mathbf{u} , it can be defined to provide another input to the emulator upon rotation by a row orthogonal matrix Θ

$$\mathbf{x} =: \mathbf{u}^T \Theta \tag{1}$$

Therefore, the Gaussian process can now be represented by the rotated input. Where the mean $\bar{f}(\mathbf{x})$, learnt from the original training data, can now be conditioned by calculating the variance of $\bar{f}(\mathbf{u}^T \Theta)$ due to the first d components of \mathbf{u} . Hence, GSA is proceeding in the same fashion now as it would have by calculating ordinary Sobol' indices. However, the analytic expressions for the variance can only be expressed due to the ARD kernel [?]. The algebra deriving these expressions would be vastly complicated by any other kernel.

The ROM is now completed by maximising the Sobol' index of each rotated input in turn by entirely optimising Θ row by row from top to bottom. The work is done using gradient descent, repeatedly rotating the input basis by Θ until the Sobol' indices are maximized so much that $\Theta \approx \mathbf{I}$.

This specific method of ROM is incredibly useful for the research conducted in this project due to it being directly derived using GPs and GSA. The method ultimately combines two of the main tools used in this research to create a new method that will benefit a broad range of engineering topics. The desire to reduce the dimensions of inputs without impacting the output is twofold: to achieve accurate descriptions of a system at a much lower computational cost and to provide a means of which a process can be visualized. For example, an industrial plant may have an output that wants to be analysed but is difficult to obtain and many input variables can be varied. So analysing the change in these variables to the change in the output can be difficult. A ROM can be used to capture most of, if not all of, the system's fundamental dynamics with a much smaller amount of dimensions to consider. This could make a significant difference as simple as being able to plot the output with respect to just one or two input dimensions instead of having to consider a large degree of variables. This method also introduces far fewer complications involving the use of GSA with dependent inputs due to it reducing the input to sets which are independent. The rotation of the input space represents combinations of dependent inputs and so the analysis deals with the issue of dependent inputs by simply combining them, ensuring each rotated basis is independent to one another.

However, the ROM does have a major limitation as it's goal is to reduce computational costs yet this is only achieved once the ROM has been created. Before that, the ROM stresses a huge computational burden due to the optimization of Θ creating many iterations of GP training. This repeated loop of training a GP, then calculating it's Sobol' indices, then rotating the input basis creates a huge amount of computational costs. Therefore, an

interesting concept to address would be whether the ROM could encapsulate the use of sparse GPs to be able to reduce the computational costs.

PLACEHOLDER

The antidote to this perceived curse is to infer that monitoring more inputs must diminish their mutual independence or their relevance. The cure is to reduce the inputs to a set which is both mutually independent and highly relevant to the response. Topologically this corresponds to sweeping naively sampled input points away from the faces of a sampling hypercube. Most of the sample points get swept towards the heart of the input space, where operating conditions are normal and safe. The few remaining input samples get swept towards selected corners of the input hypercube, representing unsafe conditions in which several inputs record anomalous values simultaneously. Heuristically this captures failure due to inscrutably arcane causes, which is first detected in several measured variables at once, as well as failure due to unsafe levels in a single highly relevant input. Model order reduction basically evacuates then excises those regions of input space which are irrelevant to the response, or represent impossible combinations of dependent inputs. This can only be achieved by a principled analysis of the response $y(\mathbf{x})$ which detects both irrelevance and dependence. Which is dauntingly hard for all the same reasons it is so sorely needed.

The purpose of this work is to present a Global Sensitivity Analysis based model order reduction approach, which uses

- One point;
- second point

This paper is organised as follows: Section

2. Review of Gaussian Processes and Global Sensitivity Analysis

2.1. Gaussian Process Surrogate

In order to avoid the difficulty and expense in obtaining and analyzing response data we adopt the a Gaussian Process (GP) as a surrogate or emulator. The response $y(\mathbf{x})$ to arbitrarily fixed input is modelled as the sum $f(\mathbf{x}) + e(\mathbf{x})$ of two Gaussian random variables encapsulating coherent signal

this
all
needs
to be
de-
scribed
with
re-
spect
to ex-
isting
litera-
ture

and incoherent noise. The latter is characterized by a zero-mean distribution that is independent of the input:

$$e(\mathbf{x}) \sim \mathcal{N}[0, \sigma_e^2]$$

The signal $f(\mathbf{x})$ is characterized by its covariance kernel $\sigma_f^2 k(\mathbf{x}_n, \mathbf{x})$ which measures the similarity between inputs \mathbf{x}_n and \mathbf{x} , and propagates any similarity to $y(\mathbf{x}_n)$ and $y(\mathbf{x})$. In the majority of applications, the kernel is naturally stationary, a function of $(\mathbf{x} - \mathbf{x}_n)$ alone. We shall further assume that the kernel is twice differentiable at its maximum ($\mathbf{x} = \mathbf{x}_n$). Hence, the Hessian at the maximum must be symmetric negative semi-definite and therefore diagonalizes to

$$\partial_{\mathbf{x}\mathbf{x}} \log k(\mathbf{x}, \mathbf{x}) =: -\Theta^\top \Lambda^{-2} \Theta$$

When $|\mathbf{x} - \mathbf{x}_n|$ is large the kernel value is miniscule in any any relevant direction. The kernel details are therefore largely irrelevant to the response *any* time $|\mathbf{x} - \mathbf{x}_n|$ is large, advocating (if not justifying) the Taylor approximation

$$k(\mathbf{x}_n, \mathbf{x}) = \exp \left(-\frac{(\mathbf{x} - \mathbf{x}_n)^\top \Theta^\top \Lambda^{-2} \Theta (\mathbf{x} - \mathbf{x}_n)}{2} (1 + O(|\mathbf{x} - \mathbf{x}_n|)) \right)$$

This paper is exclusively concerned with kernels of this form. The differentiability we have imposed forces the power spectrum of the signal f to decay rapidly. Modes of response oscillating rapidly with \mathbf{x} are interpreted as noise by the GP, as the kernel smoothes y into f . Such regularization is often, but not always, desirable, to avoid wildly unreliable interpolation of an overfit regression.

2.2. Kernel Optimization

In order to deal with the curse of dimensionality we propose to find orthogonal rotation matrix Θ and diagonal length-scale matrix Λ which best fit observed responses $y(\mathbf{X}^\top)$. The largest lengthscales in Λ mark the least relevant directions that can be ignored. However, the best fit must optimize $M(M+1)/2 + 2$ hyperparameters simultaneously to determine $\Theta, \Lambda, \sigma_f^2$ and σ_e^2 . As a direct optimization of such a large problem may sto get bogged down in local optima. Exploratory grid search is astronomically expensive $O(\exp(M(M+1)/2))$, likewise any random sampling which is not hopelessly sparse. Perhaps for these reasons, Θ has always been fixed as identity in the

literature. The lengthscales comprising Λ are also usually identical, furnishing a radial basis function (RBF) kernel. The few studies where Λ is not identical speak of an automatic relevance determination (ARD) kernel, with model order reduction in mind. However, admitting no rotation Θ severely restricts the reductions available. If relevant inputs are mutually dependent, they must all be retained. Rotation, on the contrary, combines them into a low dimensional subspace.

2.3. Global Sensitivity Analysis

This paper proposes to achieve kernel optimization indirectly, via global sensitivity analysis (GSA). The surrogate expectation

$$\mathbb{E}_\Omega[y(\mathbf{x})] = \mathbb{E}_\Omega[f(\mathbf{x})] =: \bar{f}(\mathbf{x})$$

has a variation (over $\mathbf{x} \in \mathbb{R}^M$) which can be apportioned by Sobol' index

$$S_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}) := \text{Var}_{\mathbf{x}}[\mathbb{E}_{\mathbf{x}}[\bar{f}(\mathbf{x}) | (\Theta\mathbf{x})_{\mathbf{m}}]] / \text{Var}_{\mathbf{x}}[\bar{f}(\mathbf{x})] \leq 1$$

to subspaces $(\Theta\mathbf{x})_{\mathbf{m}}$ of dimension $m \leq M$. These may be calculated analytically for the exponential quadratic kernel used here. To cure to the curse of dimensionality is to find $(\Theta)_{\mathbf{m} \times \mathbf{M}}$ such that $S_{\mathbf{m}} \approx 1$ for $m \ll M$. The rotation sub-matrix $(\Theta)_{\mathbf{m} \times \mathbf{M}}$ has a manageable number of elements if m is small. This paper takes the most economical approach, maximizing $S_{\mathbf{m}}$ for $m = 1, \dots, M$ in turn, to find the most relevant direction, then the second most relevant, and so on. Other approaches are considered in Section 5.

3. Methodology

Let \mathbf{X} be the $(N \times M)$ design matrix of observed inputs eliciting the N response $y(\mathbf{X}^\top)$. The observations are standardized such that

$$\begin{aligned} (\mathbf{0})_{\mathbf{M}} = \mathbb{E}[\mathbf{x}_n] &:= \sum_{n=1}^N (\mathbf{X})_{n \times \mathbf{M}}^\top \quad ; \quad 1 = \text{Var}[\mathbf{x}_n] = N^{-1} \sum_{n=1}^N (\mathbf{X})_{n \times \mathbf{M}} (\mathbf{X})_{n \times \mathbf{M}}^\top \\ 0 = \mathbb{E}[y(\mathbf{x}_n)] &:= \sum_{n=1}^N (y(\mathbf{X}^\top))_n \quad ; \quad 1 = \text{Var}[y(\mathbf{x}_n)] = N^{-1} y(\mathbf{X}^\top)^\top y(\mathbf{X}^\top) \end{aligned}$$

where boldface subscripts refer to the multi-indices

$$\emptyset =: \mathbf{0} \subseteq \mathbf{m} := (1, \dots, m) \subseteq \mathbf{M} \tag{2}$$

which always precede superscript operations (such as transposition or inversion). For brevity, we shall admit vector Gaussian probability densities $p((\mathbf{z})_{\mathbf{m}}; (\mathbf{Z})_{\mathbf{m} \times \mathbf{J}}, \Sigma_{\mathbf{z}})$ such that

$$\begin{aligned} & (p((\mathbf{z})_{\mathbf{m}}; (\mathbf{Z})_{\mathbf{m} \times \mathbf{J}}, \Sigma_{\mathbf{z}}))_j \\ & := (2\pi)^{-M/2} |\Sigma_{\mathbf{z}}|^{-1/2} \exp \left(-\frac{(\mathbf{z} - (\mathbf{Z})_{\mathbf{m} \times j})^\top \Sigma_{\mathbf{z}}^{-1} (\mathbf{z} - (\mathbf{Z})_{\mathbf{m} \times j})}{2} \right) \end{aligned} \quad (3)$$

naturally collapsing to the (scalar) normal multivariate density when $J = 1$.

3.1. Gaussian Process Surrogate

Non-parametric GP regression fits signal f and noise e Gaussian processes to

$$y(\mathbf{X}^\top) = f(\mathbf{X}^\top) + e(\mathbf{X}^\top) \quad (4)$$

This work exclusively employs objective Bayesian priors

$$\begin{aligned} f(\mathbf{X}^\top) & \sim \mathcal{N}[(\mathbf{0})_{\mathbf{N}}, \sigma_{\mathbf{f}}^2 k(\mathbf{X}^\top, \mathbf{X}^\top)] \\ e(\mathbf{X}^\top) & \sim \mathcal{N}[(\mathbf{0})_{\mathbf{N}}, \sigma_{\mathbf{e}}^2 (\mathbf{I})_{\mathbf{N} \times \mathbf{N}}] \end{aligned}$$

built on an ARD kernel [? ?]

$$k(\mathbf{x}_n, \mathbf{x}) := (2\pi)^{M/2} |\Lambda| p(\mathbf{x}; \mathbf{x}_n, \Lambda^2) \quad (5)$$

with diagonal positive definite lengthscale matrix Λ . Bayesian conditioning ultimately furnishes the predictive process

$$y(\mathbf{x}) \sim \mathcal{N}[\bar{f}(\mathbf{x}), \Sigma_{\mathbf{f}}(\mathbf{x}) + \sigma_{\mathbf{e}}^2]$$

with signal mean and variance

$$\begin{aligned} \bar{f}(\mathbf{x}) & := \sigma_{\mathbf{f}}^2 k(\mathbf{x}, \mathbf{X}^\top) \mathbf{K}^{-1} y(\mathbf{X}^\top) \\ \Sigma_{\mathbf{f}}(\mathbf{x}) & := \sigma_{\mathbf{f}}^2 k(\mathbf{x}, \mathbf{x}) - \sigma_{\mathbf{f}}^2 k(\mathbf{x}, \mathbf{X}^\top) \mathbf{K}^{-1} \sigma_{\mathbf{f}}^2 k(\mathbf{X}^\top, \mathbf{x}) \end{aligned} \quad (6)$$

where

$$\mathbf{K} := \sigma_{\mathbf{f}}^2 k(\mathbf{X}^\top, \mathbf{X}^\top) + \sigma_{\mathbf{e}}^2 (\mathbf{I})_{\mathbf{N} \times \mathbf{N}} \quad (7)$$

The $M + 2$ hyperparameters constituting Λ , $\sigma_{\mathbf{f}}$ and $\sigma_{\mathbf{e}}$ are simultaneously optimized for maximum marginal likelihood $\mathbf{p}[y|\mathbf{X}^\top]$, using the GPy software library (reference).

3.2. Global Sensitivity Analysis

Imagine a sample datum \mathbf{u} is drawn from a standardized normal test distribution

$$\mathbf{u} \sim \mathbf{N}[(\mathbf{0})_{\mathbf{M}}, (\mathbf{I})_{\mathbf{M} \times \mathbf{M}}] \quad (8)$$

The datum basis is rotated to

$$\mathbf{x} =: \Theta^\top \mathbf{u} \quad (9)$$

eliciting the conditional surrogate responses

$$f_{\mathbf{m}}((\mathbf{u})_{\mathbf{m}}) := \mathbb{E}[\bar{f}(\Theta^\top \mathbf{u}) | (\mathbf{u})_{\mathbf{m}}] \quad (10)$$

Knowledge of \mathbf{u} herein ranges from totally conditional $f_{\mathbf{M}}(\mathbf{u}) = \bar{f}(\mathbf{x})$ to unconditional ignorance $f_{\mathbf{0}} = \mathbb{E}[\bar{f}(\mathbf{x})]$. Equations (5) to (8) enable analytic integration yielding

$$f_{\mathbf{m}}((\mathbf{u})_{\mathbf{m}}) = \tilde{\mathbf{f}}^\top \frac{p((\mathbf{u})_{\mathbf{m}}; (\mathbf{T})_{\mathbf{N} \times \mathbf{m}}^\top, (\Sigma)_{\mathbf{m} \times \mathbf{m}})}{p((\mathbf{u})_{\mathbf{m}}; (\mathbf{0})_{\mathbf{m}}, (\mathbf{I})_{\mathbf{m} \times \mathbf{m}})} \quad (11)$$

where $\tilde{\mathbf{f}}$ is the Hadamard (element-wise) product \circ of two vectors

$$\tilde{\mathbf{f}} := (2\pi)^{M/2} |\Lambda| p(\mathbf{0}; \mathbf{X}^\top, \Lambda^2 + \mathbf{I}) \circ (\sigma_{\mathbf{f}}^2 \mathbf{K}^{-1} y(\mathbf{X}^\top)) \quad (12)$$

and

$$\mathbf{T} := \mathbf{X} (\Lambda^2 + \mathbf{I})^{-1} \Theta^\top \quad (13)$$

$$\Sigma := \Theta (\Lambda^{-2} + \mathbf{I})^{-1} \Theta^\top \quad (14)$$

According to these formulae, the unconditional surrogate response is

$$f_{\mathbf{0}} = \mathbb{E}[\bar{f}(\mathbf{x})] = \tilde{\mathbf{f}}^\top (\mathbf{1})_{\mathbf{N}} \quad (15)$$

which does not depend on Θ of course. Standardization of $y(\mathbf{X}^\top)$ instills an expectation of precisely zero here if $\mathbf{x}_n \sim \mathbf{N}[(\mathbf{0})_{\mathbf{M}}, (\mathbf{I})_{\mathbf{M} \times \mathbf{M}}]$ (which is often not exactly true).

Conditional variances may now be calculated as

$$D_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}) := \text{Var}[f_{\mathbf{m}}((\mathbf{u})_{\mathbf{m}})] = \frac{\tilde{\mathbf{f}}^\top \mathbf{W}_{\mathbf{m}} \tilde{\mathbf{f}}}{|2(\Sigma)_{\mathbf{m} \times \mathbf{m}} - (\Sigma)_{\mathbf{m} \times \mathbf{m}}^2|^{1/2}} - f_{\mathbf{0}}^2 \quad (16)$$

where

$$\begin{aligned}
(\mathbf{W}_{\mathbf{m}})_{n \times o} &:= \exp \left(\frac{-(\mathbf{T})_{n \times \mathbf{m}} (\Sigma)_{\mathbf{m} \times \mathbf{m}}^{-1} (\mathbf{T})_{n \times \mathbf{m}}^{\top} - (\mathbf{T})_{o \times \mathbf{m}} (\Sigma)_{\mathbf{m} \times \mathbf{m}}^{-1} (\mathbf{T})_{o \times \mathbf{m}}^{\top}}{2} \right) \\
&\times \exp \left(\frac{+((\mathbf{T})_{n \times \mathbf{m}} + (\mathbf{T})_{o \times \mathbf{m}}) (\Psi)_{\mathbf{m} \times \mathbf{m}}^{-1} (\Sigma)_{\mathbf{m} \times \mathbf{m}}^{-1} ((\mathbf{T})_{n \times \mathbf{m}}^{\top} + (\mathbf{T})_{o \times \mathbf{m}}^{\top})}{2} \right)
\end{aligned} \tag{17}$$

and

$$\Psi := \Theta (\Lambda^{-2} + \mathbf{I})^{-1} (2\Lambda^{-2} + \mathbf{I}) \Theta^{\top} \tag{18}$$

The proportion of response variance ascribable to the first m basis directions of \mathbf{u} is given by the Sobol' index

$$S_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}) := D_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}) / D_{\mathbf{M}}(\Theta) \leq S_{\mathbf{M}}(\Theta) = 1 \tag{19}$$

Analytic expressions for $\partial_{\Theta} D_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}})$ have been obtained from Eq. (16) using standard formulae for differentiating matrix inverses and determinants. As $D_{\mathbf{m}}$ projects M -dimensional $(\mathbf{x})_{\mathbf{M}}$ onto m -dimensional $(\mathbf{u})_{\mathbf{M}}$, the result is affected by just a few components of rotation:

$$(\partial_{\Theta} D_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}))_{i \times j} \neq 0 \implies i \leq m < M \tag{20}$$

In particular $D_{\mathbf{M}}(\Theta) = D_{\mathbf{M}}$ and $S_{\mathbf{M}}(\Theta) = 1$ are independent of Θ , as there is no projection, only rotation, in transforming $(\mathbf{x})_{\mathbf{M}}$ into $(\mathbf{u})_{\mathbf{M}}$.

3.3. Basis Optimization

At this point in the analysis, everything has been fixed save the rotation

$$\mathbf{u} := \Theta \mathbf{x} \tag{21}$$

relating sampling distribution $\mathbf{u} \sim \mathbf{N}[(\mathbf{0})_{\mathbf{M}}, (\mathbf{I})_{\mathbf{M} \times \mathbf{M}}]$ to the input of the surrogate response $\tilde{f}(\mathbf{x})$. This rotation will now be determined by maximizing the relevance – as measured by Sobol' index – of each \mathbf{u} -direction in turn. This means optimizing Θ in Eq. (21) row by row from top to bottom.

Row orthonormality leaves just $(M - m - 1)$ elements free in row m , which we encode as

$$(\Theta)_{\mathbf{m} \times \mathbf{M}} =: (\Xi)_{\mathbf{m} \times \mathbf{M}} \tilde{\Theta} \tag{22}$$

where Ξ is orthogonal, and identical on the $(m - 1)$ rows already optimized

$$\begin{aligned} (\Xi)_{\mathbf{m} \setminus \{m\} \times \mathbf{M}} &= \mathbf{I}_{\mathbf{m} \setminus \{m\} \times \mathbf{M}} \\ (\Xi)_{m \times \mathbf{m} \setminus \{m\}} &= (\mathbf{0})_{1 \times \mathbf{m} \setminus \{m\}} \\ (\Xi)_{m \times m} &= \left(1 - \sum_{k=m+1}^M (\Xi)_{m \times k}^2 \right)^{1/2} \end{aligned} \quad (23)$$

The last line induces a derivative adjustment

$$\frac{\partial}{\partial (\Xi)_{m \times k}} = \frac{\partial}{\partial (\Xi)_{m \times k}} - \frac{(\Xi)_{m \times k}}{(\Xi)_{m \times m}} \frac{\partial}{\partial (\Xi)_{m \times m}} \quad (24)$$

which should be exploited by the optimizer as a powerful repellant to orthonormality violations. This work uses a BFGS optimizer, fed an analytic Jacobian.

Given these constraints, row m is optimally determined by

$$(\Xi)_{m \times \mathbf{M} \setminus \mathbf{m}} = \underset{(\Xi)_{m \times \mathbf{M} \setminus \mathbf{m}}}{\operatorname{argmax}} S_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}) = \underset{(\Xi)_{m \times \mathbf{M} \setminus \mathbf{m}}}{\operatorname{argmax}} D_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}}) \quad (25)$$

The optimal row m is then incorporated in $\tilde{\Theta}$ and Ξ according to

$$\begin{aligned} \tilde{\Theta} &= \mathbf{Q}^\top \text{ where } \Theta^\top = \mathbf{Q}\mathbf{R} \text{ is the QR factorization of the update} \\ (\Xi)_{\mathbf{m} \times \mathbf{M}} &= \mathbf{I}_{\mathbf{m} \times \mathbf{M}} \end{aligned} \quad (26)$$

ready to optimize row $m + 1$. Optimization followed by incorporation is performed for $m = 1, \dots, M - 1$ in turn to entirely optimize Θ . The later rows could be left unoptimized, though they are successively cheaper to obtain.

3.4. Summary

The main loop of the algorithm is described i:

During testing the optimization in Step 5 is found to be prone to converge to local optima, especially in the first iteration or two of the outermost loop. An approach was therefore developed that the early iterations explore the behaviour of $(\Xi)_{m \times \mathbf{M} \setminus \mathbf{m}}$ by grid or randomized search, before attempting exploitation by gradient descent).

As the input basis is updated at each step, $\mathbf{X} = \mathbf{U}$ ultimately. The key output of the algorithm is the overall rotation Θ_Π of the original basis for \mathbf{x} to the optimal basis for \mathbf{u} .

Algorithm 1 Summary of the basis optimization algorithm.

- 1: **repeat**
- 2: Fit GP surrogate to $y(\mathbf{X}^\top)$, determining $\bar{f}(\mathbf{x})$ according to Section 3.1
- 3: Set $\tilde{\Theta} \leftarrow \Theta \leftarrow \Theta_\Pi \leftarrow \mathbf{I}$
- 4: **for** $m = 1$ **to** M **do**
- 5: According to Section 3.2, optimize

$$(\Xi)_{m \times \mathbf{M} \setminus \mathbf{m}} \leftarrow \operatorname{argmax}_{(\Xi)_{m \times \mathbf{M} \setminus \mathbf{m}}} D_{\mathbf{m}}((\Theta)_{\mathbf{m} \times \mathbf{M}})$$

- 6: where $(\Theta)_{\mathbf{m} \times \mathbf{M}} =: (\Xi)_{\mathbf{m} \times \mathbf{M}} \tilde{\Theta}$
 - 7: Update $\tilde{\Theta} \leftarrow \mathbf{Q}^\top$ where $\Theta^\top = \mathbf{Q}\mathbf{R}$
 - 8: **end for**
 - 9: Update the input basis to $\mathbf{X}^\top \leftarrow \Theta \mathbf{X}^\top$
 - 10: Update the overall rotation to $\Theta_\Pi \leftarrow \Theta \Theta_\Pi$
 - 11: **until** $\Theta \approx \mathbf{I}$
-

4. Results

In this section, the method described in Section 3.4 is applied to a series of test functions to evaluate its performance. In each case an $N \times M$ design matrix \mathbf{X} is sampled from a standard normal distribution. The input to the test function $f: [x_-, x_+]^M \rightarrow \mathbb{R}$ is generally constructed as

$$\hat{\mathbf{X}}^\top = (x_+ - x_-)c(\Phi \mathbf{X}^\top) + x_-(\mathbf{I})_{\mathbf{M} \times \mathbf{N}} \quad (27)$$

where $c: \mathbb{R}^M \rightarrow \mathbb{R}^M$ is the cumulative density function for M independent standard normal random variables, and Φ is a test rotation matrix. The corresponding optimal input rotation from Section 3.4 is

$$\Theta_\Pi = \begin{cases} \Theta_{\mathbf{I}} & \text{if } \Phi \text{ is identity matrix } \mathbf{I} \\ \Theta_{\mathbf{R}} & \text{if } \Phi \text{ is a random rotation matrix } \Phi_{\mathbf{R}} \end{cases} \quad (28)$$

which should recover the random rotation as

$$\Theta_{\mathbf{R}} = \Phi_{\mathbf{R}} \Theta_{\mathbf{I}} \quad (29)$$

This provides the test measure

$$|\epsilon_\Theta| = \left\| (\Theta_{\mathbf{R}}^\top \Phi_{\mathbf{R}} \Theta_{\mathbf{I}} - \mathbf{I})_{\underline{\mathbf{M}} \times \underline{\mathbf{M}}} \right\| \quad (30)$$

where $\|\cdot\|$ is the usual Frobenius matrix norm and $\underline{\mathbf{M}}$ includes no irrelevant input directions. This reduced dimensionality is determined for each test function by the cumulative Sobol' index $\underline{\mathbf{M}}$ exceeding 95%. All inputs and outputs are standardized to a mean of zero and variance of 1. All functions are tested with and without output noise $e \sim \mathcal{N}[0, 0.001]$ added to $f(\hat{\mathbf{X}}^\top)$. In the noiseless cases $e = 0$ set a lower bound on the Gaussian process noise variance as $\epsilon_e \geq 10^{-6}$ in order to prevent numerical instability in the calculation of Sobol' indices which is otherwise observed.

what
do we
mean
irrelevant?

How
are
the
inputs
ob-
tained?

4.1. Sine Function

$$f(\hat{\mathbf{x}}) := \sin(\hat{\mathbf{x}}_1) \quad (31)$$

$$S_1 := S_{(1)} = 1$$

Noise	N	0	1	2	3	4
0.0010	100	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	100	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	200	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	200	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	400	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	400	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	800	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	800	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	1600	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	1600	1.0000	1.0000	1.0000	1.0000	1.0000
Noise	N	0	1	2	3	4
0.0010	100	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	100	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	200	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	200	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	400	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	400	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	800	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	800	1.0000	1.0000	1.0000	1.0000	1.0000
0.0010	1600	1.0000	1.0000	1.0000	1.0000	1.0000
0.0000	1600	1.0000	1.0000	1.0000	1.0000	1.0000

Noise	N	0	1	2	3	4
0.0010	100	0.8997	0.9253	0.9576	0.9969	1.0000
0.0000	100	0.9315	0.9697	0.9927	0.9994	1.0000
0.0010	200	0.8972	0.9294	0.9544	0.9791	1.0000
0.0000	200	0.9145	0.9799	0.9894	0.9957	1.0000
0.0010	400	0.8996	0.9362	0.9649	0.9959	1.0000
0.0000	400	0.9341	0.9683	0.9910	0.9979	1.0000
0.0010	800	0.9009	0.9568	0.9795	1.0000	1.0000
0.0000	800	0.8920	0.9349	0.9536	0.9807	1.0000
0.0010	1600	0.9085	0.9504	0.9689	0.9790	1.0000
0.0000	1600	0.8844	0.9300	0.9758	0.9996	1.0000

4.2. Ishigami Function

$$f(\mathbf{x}) := (1 + b\mathbf{x}_3^4) \sin(\mathbf{x}_1) + a \sin^2(\mathbf{x}_2) \quad (32)$$

$$a = 7.0 \quad ; \quad b = 0.1$$

$$S_1 = 0.3139 \quad ; \quad S_2 = 0.7563 \quad ; \quad S_3 = 1$$

Noise	N	0	1	2	3	4
0.0010	100	0.2682	0.7942	0.9991	1.0000	1.0000
0.0000	100	0.2648	0.7874	0.9995	0.9999	1.0000
0.0010	200	0.3086	0.7454	1.0000	1.0000	1.0000
0.0000	200	0.2991	0.7629	1.0000	1.0000	1.0000
0.0010	400	0.3139	0.7628	1.0000	1.0000	1.0000
0.0000	400	0.3124	0.7652	1.0000	1.0000	1.0000
0.0010	800	0.3147	0.7605	1.0000	1.0000	1.0000
0.0000	800	0.3149	0.7574	1.0000	1.0000	1.0000
0.0010	1600	0.3151	0.7589	1.0000	1.0000	1.0000
0.0000	1600	0.3146	0.7583	1.0000	1.0000	1.0000

Noise	N	0	1	2	3	4
0.0010	100	0.4941	0.7919	0.9996	1.0000	1.0000
0.0000	100	0.4803	0.7872	0.9995	0.9999	1.0000
0.0010	200	0.4239	0.7456	1.0000	1.0000	1.0000
0.0000	200	0.4499	0.7631	1.0000	1.0000	1.0000
0.0010	400	0.4439	0.7629	1.0000	1.0000	1.0000
0.0000	400	0.4478	0.7652	1.0000	1.0000	1.0000
0.0010	800	0.4420	0.7595	1.0000	1.0000	1.0000
0.0000	800	0.4402	0.7574	1.0000	1.0000	1.0000
0.0010	1600	0.4430	0.7592	1.0000	1.0000	1.0000
0.0000	1600	0.4429	0.7583	1.0000	1.0000	1.0000
Noise	N	0	1	2	3	4
0.0010	100	0.3808	0.5331	0.8077	0.9945	1.0000
0.0000	100	0.3854	0.6824	0.9823	1.0000	1.0000
0.0010	200	0.2441	0.3533	0.7709	0.9245	1.0000
0.0000	200	0.3351	0.5265	0.7442	0.9608	1.0000
0.0010	400	0.2196	0.3092	0.4799	0.7473	1.0000
0.0000	400	0.2478	0.3664	0.6252	0.8237	1.0000
0.0010	800	0.2276	0.3492	0.5546	0.7574	1.0000
0.0000	800	0.3034	0.5199	0.5991	0.7104	1.0000
0.0010	1600	0.2284	0.3242	0.4808	0.7545	1.0000
0.0000	1600	0.1979	0.3638	0.5055	0.6648	1.0000

4.3. Sobol' G Function

$$f(\mathbf{x}) := \prod_{i=1}^D \frac{|4\mathbf{x}_i - 2| + \mathbf{a}_i}{1 + \mathbf{a}_i} \quad (33)$$

$$\mathbf{a}_i = (i - 1)/2$$

$$S_1 = 0.4107 \quad ; \quad S_2 = 0.6541 \quad ; \quad S_3 = 0.8113 \quad ; \quad S_4 = 0.9203 \quad ; \quad S_5 = 1$$

Noise	N	0	1	2	3	4
0.0010	100	0.4095	0.8833	0.9380	0.9732	1.0000
0.0000	100	0.4157	0.6609	0.7603	0.9857	1.0000
0.0010	200	0.4532	0.6910	0.8704	0.9257	1.0000
0.0000	200	0.4678	0.6743	0.8132	0.9560	1.0000
0.0010	400	0.4308	0.6797	0.8270	0.9069	1.0000
0.0000	400	0.4480	0.6688	0.7992	0.9210	1.0000
0.0010	800	0.4388	0.6631	0.7980	0.9035	1.0000
0.0000	800	0.4538	0.6882	0.8153	0.9205	1.0000
0.0010	1600	0.4348	0.6720	0.8100	0.9056	1.0000
0.0000	1600	0.4377	0.6663	0.8140	0.9198	1.0000
Noise	N	0	1	2	3	4
0.0010	100	0.4365	0.8781	0.9398	0.9773	1.0000
0.0000	100	0.4222	0.6606	0.8598	0.9868	1.0000
0.0010	200	0.4508	0.6860	0.8809	0.9700	1.0000
0.0000	200	0.4825	0.6730	0.8385	0.9828	1.0000
0.0010	400	0.4393	0.6751	0.8260	0.9210	1.0000
0.0000	400	0.4507	0.6705	0.8018	0.9207	1.0000
0.0010	800	0.4396	0.6639	0.7987	0.9039	1.0000
0.0000	800	0.4543	0.6890	0.8160	0.9206	1.0000
0.0010	1600	0.4350	0.6722	0.8101	0.9057	1.0000
0.0000	1600	0.4379	0.6665	0.8141	0.9198	1.0000
Noise	N	0	1	2	3	4
0.0010	100	0.8419	0.8882	0.9250	0.9703	1.0000
0.0000	100	0.8429	0.8935	0.9459	0.9847	1.0000
0.0010	200	0.5299	0.6439	0.7479	0.8641	1.0000
0.0000	200	0.5656	0.7679	0.8886	0.9622	1.0000
0.0010	400	0.8754	0.9148	0.9435	0.9771	1.0000
0.0000	400	0.6352	0.7292	0.8217	0.9295	1.0000
0.0010	800	0.4523	0.5722	0.7101	0.8671	1.0000
0.0000	800	0.6907	0.7239	0.7712	0.9566	1.0000
0.0010	1600	0.8432	0.8783	0.9230	0.9715	1.0000
0.0000	1600	0.8595	0.9406	0.9718	0.9859	1.0000

5. Discussion