Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling

Maliki Moustapha, Bruno Sudret

ETH zürich

Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling

M. Moustapha and B. Sudret

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:

Report Ref.: RSUQ-2022-012

Arxiv Ref.: https://arxiv.org/abs/2211.16909 [stat.ML] [stat.CO] [stat.AP]

DOI:

Date submitted: November 30, 2022

Date accepted:

Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling

M. Moustapha[1] and B. Sudret[1]

[1] *Chair of Risk, Safety and Uncertainty Quantification,*

ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

Abstract

Surrogate models have shown to be an extremely efficient aid in solving engineering problems that require repeated evaluations of an expensive computational model. They are built by sparsely evaluating the costly original model and have provided a way to solve otherwise intractable problems. A crucial aspect in surrogate modelling is the assumption of smoothness and regularity of the model to approximate. This assumption is however not always met in reality. For instance in civil or mechanical engineering, some models may present discontinuities or non-smoothness *e.g.*, in case of instability patterns such as buckling or snap-through. Building a single surrogate model capable of accounting for these fundamentally different behaviours or discontinuities is not an easy task. In this paper, we propose a three-stage approach for the approximation of non-smooth functions which combines clustering, classification and regression. The idea is to split the space following the localized behaviors or regimes of the system and build local surrogates that are eventually assembled. A sequence of well-known machine learning techniques are used: Dirichlet process mixtures models (DPMM), support vector machines and Gaussian process modelling. The approach is tested and validated on two analytical functions and a finite element model of a tensile membrane structure.

Keywords: Surrogate modelling - non-smooth functions - discontinuities - Dirichlet process mixture models – uncertainty quantification

1 Introduction

Computational models, which allow scientists and engineers to accurately simulate complex systems and predict their behaviour in various contexts, are nowadays a key tool present in

virtually all fields of applied sciences and engineering. Cast as computer experiments, they are able to predict with high fidelity the behaviour of the studied system in replacement of, or as a complement to laboratory experiments. The downside of such high-fidelity models is however that they are computationally demanding. This is even more relevant in the context of uncertainty quantification or design optimization, where the models need to be evaluated multiple times.

Surrogate models have become paramount in such fields as they allow for an efficient solution of otherwise computationally intractable problems. They are inexpensive proxies that can be used *in lieu* of expensive computational models. Examples of such surrogates include Gaussian process models also known as Kriging (Santner et al., 2003; Rasmussen and Williams, 2006), polynomial chaos expansions (Xiu and Karniadakis, 2002; Blatman and Sudret, 2011), support vector machines (Vapnik, 1995), polynomial response surfaces (Myers and Montgomery, 2002), etc. These methods have been applied in various problems pertaining to uncertainty quantification or design optimization. The use of surrogate models in such fields are now mature as shown by the recent reviews in reliability analysis (Teixeira et al., 2021; Moustapha et al., 2022), Bayesian inversion (Yan and Zhang, 2017) or design optimization (Chatterjee et al., 2019; Moustapha and Sudret, 2019a).

In most of these applications, it is assumed that the computational models to approximate feature some accommodating properties such as smoothness, differentiability or stationarity. Yet there exists cases when these assumptions do not hold. In mechanical engineering, this may happen for instance when solving non-linear problems involving instability such as snap-through or bifurcations in the solution path, *e.g.*, crash simulation. In computational fluid dynamics, simulations of compressive flows that involve shocks also belong to this category. In other cases, the underlying phenomenon may present different localized features or extreme regime variations which are strongly dependent on the inputs.

Various methods have been developed in the field of uncertainty quantification to tackle such problems. The first class of methods borrows from digital signal processing and image detection to identify discontinuities or strong gradients of the function to approximate using techniques such as polynomial annihilation (Le Maître et al., 2004; Gorodetsky, 2012). Such approaches however rely on uniformly sampled grids and are often limited to two-dimensional problems. Sargsyan et al. (2012) proposed a technique combining Bayesian inference and polynomial chaos expansions that does not require using a regular grid and hence allowing for a reduced number of samples. However, their approach was also developed for two-dimensional problems and the authors did not investigate how well it scales with dimensionality.

Another class of methods relies on Gaussian process (GP) regression where the irregularities on the model to approximate are tackled by introducing non-stationary covariance functions or

kernels. Indeed, such kernels allow one to capture heterogeneous variations or heteroscedastic noise while keeping the computational budget low. The direct approach to build such kernels is to consider the noise variance, signal variance and/or characteristic length scale to be input-dependent, such as in Paciorek and Schervish (2003). Heinonen et al. (2016) proposed an approach where all three parameters are considered latent variables and inferred as hyper-parameters of the GP. Such an approach has shown increased efficiency compared to vanilla GP but it also comes with an increased inference cost due to the fact that there are no more closed-form solution and the hyperparameters need to be calibrated using sampling based techniques (See Rasmussen and Ghahramani (2001)). Furthermore, they do not allow to tackle problems with discontinuities.

A more sensible approach based on non-stationary GP consists in splitting the input space using for instance treed Gaussian processes or a mixture of experts (Tresp, 2000; Rasmussen and Ghahramani, 2001; Meeds and Osindero, 2005). Similarly, it is also possible to define non-stationary Gaussian process models by partitioning the training data into smaller subsets using clustering techniques, such as in Zhang et al. (2019) and Konomi et al. (2019), where K-means and nearest-neighbors clustering are used. Such approaches also have the advantage of offering faster training and testing of the model as the experimental design is divided into smaller and more computationally manageable subsets. Finally, another popular way to define non-stationary kernels is by warping the input, and sometimes the output, space. By doing so, one may find a latent space where the function to approximate is smoother. Examples of such techniques include warped GP (Marmin, 2018) or manifold GP regression (Calandra et al., 2016; Kuleshov et al., 2018).

In this work, we will focus on multi-stage techniques where the problem is solved by using a sequence of well-known machine learning techniques. More specifically, we consider the class of methods based on the following three-stage approach: clustering, classification and regression (Boroson and Missoum, 2017; Dupuis et al., 2018). Basudhar and Missoum (2008); Serna and Bucher (2009) were the first to propose decomposing the problem of identifying multiple failure domains of mechanical systems using support vector machines. However, they do not include the regression step as they are only concerned with an optimization problem where only the state of a sample is of interest (*i.e.*, whether it belongs to the failure domain or not). Moustapha (2016); Moustapha and Sudret (2019b) extended the approach to the prediction of the model responses by building local Kriging surrogates in each identified domain. However in all these approaches, it was assumed that the clusters were identified either using expert knowledge or by only considering the model responses which span different ranges. Niutta et al. (2018) proposed identifying the clusters by detecting jumps in the model responses for relatively close samples. However, this technique works only in low-dimensional problems and when the response of different clusters are

disjoint. This is a strong limitation and was to some extent overcome by using joint clustering of both the inputs and outputs in Bernholdt et al. (2019). In that work, they use K-means clustering to identify the clusters and multi-layer perceptrons for classification and regression tasks. The number of clusters is defined here using the elbow approach, which is a visual technique requiring user interaction. Furthermore it is not robust w.r.t. the initialization of the K-means algorithm and noise in the data. More generally, an important limitation in the contributions presented above is that the three steps are disconnected and the prediction uncertainty in one step is not accounted for in the subsequent ones.

In this paper, we propose an approach that aims at solving these two limitations. First, to automatically identify the number of clusters in a robust way, we consider a non-parametric Bayesian technique, namely *Dirichlet process mixture models* (DPMM). The interest in using DPMM are three-fold: i. they automatically estimate the optimal number of clusters according to patterns identified in the data, ii. they offer a probabilistic framework that allows one to propagate the epistemic uncertainty related to this clustering task to both the subsequent classification and regression steps, and iii. they are flexible enough and their complexity can grow as new data is observed (for instance in an active learning scheme, where new regimes of the model could be identified).

In the remainder of this paper, we first present the three-stage methodology and how the steps are connected in Section 2. In Section 3, we present in details the three methods used in each step, namely, Dirichlet process mixture models, support vector machines for classification and Gaussian process modelling. We finally illustrate the proposed approach in Section 4 using two analytical examples and an engineering application related to the design of a tensile membrane structure (Valdés-Vázquez et al., 2020, 2021).

2 Problem set-up and three-stage approach

Let us consider a set of N data points $(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X} = \left\{\boldsymbol{x}^{(i)} \in \mathbb{X} \subset \mathbb{R}^M, i = 1, \dots N\right\}$ is a set of M-dimensional inputs and $\mathcal{Y}$ are corresponding scalar outputs such that $\mathcal{Y} = \left\{y^{(i)} = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right) \in \mathbb{R}, i = 1 \dots N\right\}$. The model $\mathcal{M}$ is assumed black-box, meaning that it is only accessible through an evaluation over a finite set of input points. We further assume in this setting that the model is non-smooth, *i.e.*, it exhibits sharp localized features and, most noticeably, discontinuities. As the model can only be evaluated on a finite set of samples, discontinuities in the current work is assumed when the model presents extreme variations in the outputs for seemingly close input points.

The goal of the analysis is to learn the input-output relationship of the model $\mathcal{M}$ through the limited set of training data $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, also known as *experimental design*. This ultimately leads

to a cheaper-to-evaluate surrogate model that can be used to predict the response of the model for any new point. Generally, this type of problems is tackled using regression techniques where a class of parameterized models are assumed and then their hyper-parameters are calibrated so as to minimize a generalization error. Such models would however fail when there are discontinuities or heterogeneous variations associated to limited observations.

In this work, we consider tackling this problem by splitting the space along the discontinuities and building local regression models in each of the obtained subdomains. To achieve this, we consider a three-stage framework which is illustrated in Figure 1 and summarized as follows:

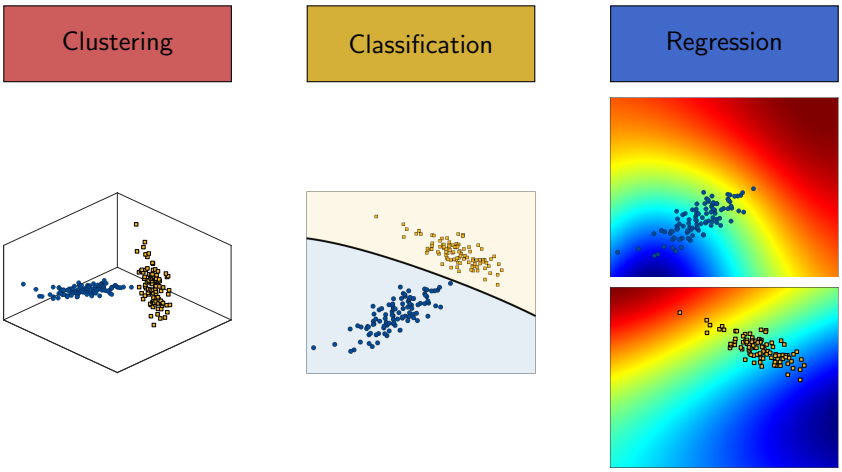


Figure 1: Illustration of the three-stage approach.

1. **Clustering:** The first learning step aims at identifying patterns in the data that hint to subdomains separated by discontinuities. To achieve this, we cluster the *joint input-output* data points. This is an unsupervised learning problem for which numerous techniques have been developed (Pham and Afify, 2017). K-means clustering (Lloyd, 1982) is probably the most common approach thanks to its simplicity. However, it assumes that the number of clusters is known and further fails when the clusters are of disproportionate sizes. Another approach that partially overcomes difficulties related to K-means clustering are Gaussian mixture models which offer a probabilistic framework for clustering (Rokach and Maimon, 2005). They hence allow for a more nuanced clustering of the data by providing soft cluster memberships, *i.e.*, each data point is assigned with a probability of belonging to a given cluster. This feature allows one to solve more complex problems, *e.g.*, when the clusters are partially overlapping. However, similarly to K-means, they assume that the number of clusters is known in advance. In general, trial-and-errors approaches are used to define the optimal number of clusters for such problems, which is not optimal.

   We therefore consider in this work a more holistic approach where the number of clusters

is also inferred from the data using a non-parametric Bayesian model, more specifically Dirichlet process mixture models (Li et al., 2019) as described in Section 3.1.

At the end of this step, the experimental design is split into K subsets $\mathcal{C}_k$, $k = 1, \ldots, K$.

2. **Classification:** Assuming that the data have been clustered, we can now place labels on them and turn to supervised learning. More specifically, let us assume K clusters are identified in the previous step. We thus define the labels $\{\ell_1, \ldots, \ell_K\}$ and the labelled training data $\mathcal{X} \times \mathcal{L}$ where each couple $\left(\boldsymbol{x}^{(i)}, \ell^{(i)}\right)$ is defined such that $\ell^{(i)} = \ell_k$ if $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right) \in \mathcal{C}_k$. The goal of this step is then to partition the input space such that any new sample can be mapped to at least one of the clusters $\mathcal{C}_k$. This will ultimately allow us to select the appropriate local regression model(s) to evaluate the new point.

   This task is carried out in this work by using support vector machines (SVM) for binary and multi-class classification (Vapnik, 1995). The probabilistic framework is introduced by considering Platt's approach to computing posterior probabilities given a binary SVM prediction (Platt, 2000). For multi-class problems, binary classifiers are appropriately combined to provide both class membership and posterior probabilities.

3. **Regression:** In this final step, Gaussian process (GP) models (Rasmussen and Williams, 2006) are employed to make the final prediction. We further investigate the use of three different approaches for combining the various GP models built in this stage. In the first two approaches, local surrogate models $\widehat{\mathcal{M}}_k$ are built for each of the K identified clusters. When it comes to prediction, the recombination is made as follows:

   - *Hard recombination*: In this approach, the surrogate model which corresponds to the cluster predicted by the classifier is solely used to make the final prediction, *i.e.*,

   $$\widehat{\mathcal{M}}(\boldsymbol{x}) = \sum_{k=1}^{K} \mathbb{1}_{\mathcal{C}_k}(\boldsymbol{x})\, \widehat{\mathcal{M}}_k(\boldsymbol{x}), \tag{1}$$

   where $\mathbb{1}_{\mathcal{C}_k}(\boldsymbol{x})$ is equal to 1 if $\boldsymbol{x}$ is predicted to belong to the cluster $\mathcal{C}_k$, *i.e.*, $\mathcal{M}^{\mathrm{SVC}}(\boldsymbol{x}) = \ell_k$ and 0 otherwise;

   - *Soft recombination*: In this approach, the prediction for each point is obtained as a weighted combination of all the local surrogate models, *i.e.*,

   $$\widehat{\mathcal{M}}(\boldsymbol{x}) = \sum_{k=1}^{K} w_k(\boldsymbol{x})\, \widehat{\mathcal{M}}_k(\boldsymbol{x}), \tag{2}$$

   where the weight $w_k(\boldsymbol{x}) \in [0, 1]$ with $\sum_{k=1}^{K} w_k(\boldsymbol{x}) = 1$ may be related to the actual probability that the point $\boldsymbol{x}$ belongs to the cluster $\mathcal{C}_k$ as defined by the classifier.

   - *Categorical recombination*: Contrary to the previous two approaches, a single Gaussian process model is built here. This is achieved by using an additional variable which is a categorical parameter indicating which cluster a given point belongs to, *i.e.*, the

training set is the couple $\{\mathcal{X}, \mathcal{L}\} \times \mathcal{Y}$ where $\mathcal{L} = \left\{\ell^{(i)}, i = 1, \ldots, N\right\}$ are the labels of the training set identified in the clustering stage. The surrogate model is therefore built on a space of dimension $M + 1$: $\widehat{\mathcal{M}}(\boldsymbol{x}) = \widehat{\mathcal{M}}^{\text{cat}}\left(\widetilde{\boldsymbol{x}} = \left(\boldsymbol{x}, \widehat{\ell}(\boldsymbol{x})\right)\right)$, where the categorical variable is given by the SVC prediction, *i.e.*, $\widehat{\ell}(\boldsymbol{x}) = \mathcal{M}^{\text{SVC}}(\boldsymbol{x})$.

The following section describes in details each of the ingredients introduced in the proposed framework.

3 Description of the components of the proposed method

3.1 Clustering using Dirichlet process mixture models

Gaussian mixture models

Let us now consider the set of available data $\mathcal{W} = \left\{\boldsymbol{w}^{(i)}, i = 1, \ldots, N\right\}$, where $\boldsymbol{w}^{(i)} = \left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$ is a vector gathering both inputs and outputs, and let us assume that they are associated to some latent variables $\boldsymbol{z}$. In a clustering set-up, say using a Gaussian mixture, the latent variables would be $\boldsymbol{z} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ where $\boldsymbol{\pi}$ are mixing coefficients and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of multivariate normal random variables. The goal is then to find the posterior distribution $p(\boldsymbol{z}|\boldsymbol{w})$ of the latent variables given the data and using Bayes rules, *i.e.*,

$$p(\boldsymbol{z}|\boldsymbol{w}) = \frac{p(\boldsymbol{w}, \boldsymbol{z})}{p(\boldsymbol{w})} = \frac{p(\boldsymbol{w}|\boldsymbol{z})\, p(\boldsymbol{z})}{p(\boldsymbol{w})} \propto p(\boldsymbol{w}|\boldsymbol{z})\, p(\boldsymbol{z}), \tag{3}$$

where $p(\boldsymbol{w}|\boldsymbol{z})$ is the data likelihood, $p(\boldsymbol{z}) = p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the prior over the latent variables and $p(\boldsymbol{w})$ is the evidence.

The prior can be fully factorized into $p(\boldsymbol{\pi})\, p(\boldsymbol{\mu})\, p(\boldsymbol{\Sigma})$ since the three parameters are considered mutually independent. The prior on the mixing coefficients $p(\boldsymbol{\pi})$ is usually chosen as a Dirichlet distribution with parameters α/K where α is a positive scaling parameter and K is the predefined number of clusters:

$$p(\pi_1, \ldots, \pi_K|\alpha) = \text{Dirichlet}(\alpha/K, \ldots, \alpha/K) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{k=1}^{K} \pi_k^{\alpha/K-1}, \tag{4}$$

where Γ is the Gamma function.

The Dirichlet distribution is chosen precisely because it is the *conjugate distribution* to the multinomial distribution, which is used for clusters membership assignment, later denoted by c. The generative model for data derived from a Gaussian mixture model can therefore be cast as

$$\begin{aligned} \pi_k &\sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K), \quad && k = \{1, \ldots, K\}, \\ c^{(i)} &\sim \text{Multinomial}(\pi_1, \ldots, \pi_K), \quad && i = \{1, \ldots, N\}, \\ \boldsymbol{w}^{(i)}|\left\{c^{(i)} = k\right\} &\sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), && i = \{1, \ldots, N\}, \end{aligned} \tag{5}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are respectively the mean and covariance parameters of each local Gaussian distribution in the mixture.

It is generally assumed in such a model that $K << N$, which in other words means that samples from all clusters have been observed. However, there may exist cases when K is in the same order or even larger than N. An alternative view to such cases is that at any moment all clusters have not yet been observed and drawing more data from the generative model will reveal new clusters. This naturally leads to extending this finite mixture model into an infinite one using non-parametric Bayesian models whose complexity can grow as more data are observed.

This is precisely what a Dirichlet process mixture model does. It generalizes the generative model described in Eq. (5) by assuming an infinite number of clusters, *i.e.*, that $K \to \infty$. This corresponds to choosing a Dirichlet process (Ferguson, 1973) as prior for the mixing coefficients, as explained in the sequel.

Dirichlet process

A Dirichlet process (DP) is a distribution over distributions defined by a base distribution G_0 and a positive scaling parameter α. The output from a Dirichlet process is therefore a discrete distribution. It is however not possible to directly draw from G considering the formal definition of a Dirichlet process. Other alternative views such as the Chinese restaurant process (Aldous, 1985), the Pólya urn scheme (Blackwell and MacQueen, 1973) or the stick-breaking representation (Sethuraman, 1994) have been proposed instead.

In this work, we consider the latter approach. More specifically, let us consider an infinite collection of two random variables $V_k \sim \text{Beta}(1, \alpha)$ and $\eta_k^* \sim G_0$ with $k = \{1, 2, \ldots\}$. The stick-breaking representation of G is then defined as follows:

$$\begin{aligned} \pi_k &= v_k \prod_{j=1}^{k-1} (1 - v_j) \,, \\ G &= \sum_{k=1}^{\infty} \pi_k (\boldsymbol{v}) \, \delta_{\eta_k^*} (\eta_k) \,, \end{aligned} \tag{6}$$

where δ is the Kronecker symbol. This representation is illustrated in Figure 2 where the η_k^* are location parameters also known as atoms and π_k are corresponding weights.

In a DP, there is a countably infinite number of atoms and the weights sum up to 1, making G a discrete distribution. This infinite set of atoms lends itself to modelling priors in infinite mixture models. More specifically, the DP is used in Dirichlet process mixture models as a non-parametric prior in a hierarchical Bayesian model specified as follows (Antoniak, 1974; Blei

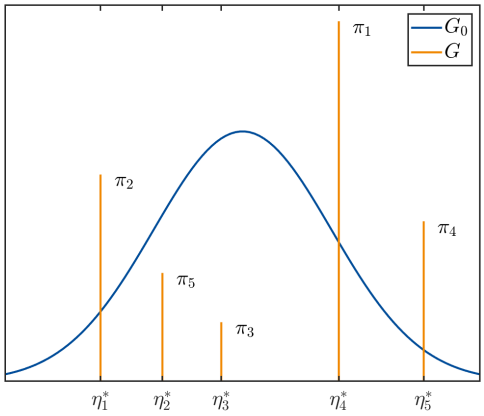


Figure 2: Illustration of a Dirichlet process: G_0 is the base distribution from which the atoms η_k^* are sampled, π_k are the corresponding weights and G a realization of the DP.

and Jordan, 2006):

$$\begin{aligned} G|\{\alpha, G_0\} &\sim DP(\alpha, G_0), \\ \eta^{(i)}|G &\sim G, \\ \boldsymbol{W}^{(i)}|\eta^{(i)} &\sim p(\boldsymbol{w}^{(i)}|\eta^{(i)}). \end{aligned} \tag{7}$$

Given a dataset $\mathcal{W}$, each data point $\boldsymbol{w}^{(i)}$ is assumed to be generated by first drawing a component label $c^{(i)} = \{1, 2, \ldots\}$ with probability distribution $p(c^{(i)} = k|\boldsymbol{V}) = \pi_k(\boldsymbol{v})$ and then drawing $\boldsymbol{w}^{(i)}$ from $p(\boldsymbol{w}^{(i)}|\eta_k)$. In this work, p is chosen as a distribution from the exponential family for which G_0 is a conjugate prior, which turns out to also belong to the exponential family and hence making inference easier.

Posterior estimation

The latent variables in this setting are therefore $\boldsymbol{z} = \{\boldsymbol{v}, \boldsymbol{\eta}, \boldsymbol{c}\}$. The goal of the analysis is then to find the posterior distribution of these latent variables given the observed data $\mathcal{W}$, which is denoted by $p(\boldsymbol{z}|\mathcal{W}, \boldsymbol{\theta})$. There is no closed-form solution to this problem and typical solution schemes rely on Markov Chain Monte Carlo (MCMC). MCMC algorithms allow one to obtain an approximation of the posterior using Markov chains whose stationary distribution is the sought posterior. The usual approach in Dirichlet process mixture models is Gibbs sampling which is particularly suited to this task as one can have access to the conditional distributions of the latent variables analytically (Neal, 2000; Ishwaran and James, 2001). However, the difficulty with MCMC algorithms is that they are expensive, as they require a large number of samples, often generated sequentially, and their convergence is difficult to monitor.

An alternative approach to circumvent these issues is *variational inference*, where the esti-

mation of the posterior is replaced by an optimization problem (Wainwright and Jordan, 2003). More specifically, the intractable posterior is replaced by a parametric family of variation distributions denoted here by $q_\nu(\boldsymbol{z}|\nu)$. In this paper, we consider the approach proposed by Blei and Jordan (2006) which relies on the mean-field approximation, *i.e.*, the variational distribution is fully factorized (all the latent variables are mutually independent). The optimization problem then consists in finding within the selected family of variational distributions the values of the hyperparameters ν that will minimize the Kullback-Liebler (KL) divergence between the true posterior and its approximation. This quantity reads

$$\begin{aligned} KL\left(q_\nu(\boldsymbol{z}|\nu)||p\left(\boldsymbol{z}|\mathcal{W},\boldsymbol{\theta}\right)\right) &= \int_{-\infty}^{\infty} q_\nu(\boldsymbol{z}|\nu)\log\left(\frac{q_\nu(\boldsymbol{z}|\nu)}{p\left(\boldsymbol{z}|\mathcal{W},\boldsymbol{\theta}\right)}\right)\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{q_\nu}\left[\log q_\nu(\boldsymbol{z}|\nu)\right] - \mathbb{E}_{q_\nu}\left[p\left(\boldsymbol{z},\mathcal{W}|\boldsymbol{\theta}\right)\right] + \log p\left(\boldsymbol{w}|\boldsymbol{\theta}\right). \end{aligned} \tag{8}$$

By noting that the divergence is always positive (or using Jensen's inequality), it can be shown that minimizing Eq. (8) is equivalent to maximizing a lower bound of the marginal log likelihood, also referred to as ELBO and denoted by

$$\log p\left(\boldsymbol{w}|\boldsymbol{\theta}\right) \geq \mathbb{E}_{q_\nu}\left[p\left(\boldsymbol{z},\mathcal{W}|\boldsymbol{\theta}\right)\right] - \mathbb{E}_{q_\nu}\left[\log q_\nu(\boldsymbol{z}|\nu)\right]. \tag{9}$$

By appropriately choosing the family of variational distributions for each latent variable, it is possible to make the computation of the ELBO tractable. In the approach proposed by Blei and Jordan (2006) considered here, the factorized variational distribution is cast as

$$q_\nu(\boldsymbol{v},\boldsymbol{\eta},\boldsymbol{z}|\nu) = \prod_{t=1}^{T-1} q_{\gamma_t}\left(v_t\right)\prod_{t=1}^{T} q_{\tau_t}\left(\eta_t\right)\prod_{k=1}^{N} q_{\Phi_k}\left(c_k\right), \tag{10}$$

where $q_{\gamma_t}\left(v_t\right)$ are Beta distributions, $q_{\tau_t}\left(\eta_t\right)$ are exponential family distributions and $q_{\Phi_k}\left(c_k\right)$ are multinomial distributions. In this equation, the infinite samples is truncated to T terms by setting $q(v_T = 1) = 1$, which implies that $\boldsymbol{\pi}_t\left(\boldsymbol{v}\right) = 0$ for $t \geq T$. The solution to this problem is eventually obtained using a coordinate ascent algorithm for which the incremental updates can be computed analytically (Ghahramani and Beal, 2000). The reader is referred to Blei and Jordan (2006) for further details.

3.2 Classification using support vector machines

3.2.1 Binary classification

Support vector machines are a popular supervised learning algorithm developed by Vapnik (1995). They were developed for binary classification and were later extended to account for multiple classes. Let us first consider the binary case (*i.e.*, assuming only two clusters were identified) and denote the dataset by $\left\{\left(\boldsymbol{x}^{(i)},\ell^{(i)}\right), i = 1,\ldots,N\right\}$ where $\ell^{(i)} = \{-1, 1\}$ are the labels of the training points.

Given this training set, the support vector classifier (SVC) prediction for any yet-to-be observed sample reads (Smola and Schölkopf, 2004)

$$\mathcal{M}^{\mathrm{SVC}}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \, \ell^{(i)} \, k\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}; \boldsymbol{\theta}\right) + b, \tag{11}$$

where $\{\boldsymbol{\alpha}, b\}$ are parameters to calibrate. The coefficients α_i, some of which are the so-called *support vectors*, and the offset parameter b are obtained by solving a quadratic optimization problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T \left(\widetilde{\boldsymbol{K}}\boldsymbol{Y}\boldsymbol{Y}^T\right)\boldsymbol{\alpha} + \boldsymbol{h}^T\boldsymbol{\alpha} \\ \text{subject to:} \quad & \boldsymbol{\alpha}^T\boldsymbol{Y} = 0, \qquad \alpha_i \geq 0, \qquad i = \{1, \ldots, N\}, \end{aligned} \tag{12}$$

where $\boldsymbol{h} = \{-1, \ldots, -1\}$ is a column vector of size N and $\widetilde{\boldsymbol{K}} = \boldsymbol{K} + 1/C\boldsymbol{I}_N$ with $C > 0$ being a penalty term. The matrix $\boldsymbol{K}$ is the so-called Gram matrix built by evaluating the parameterized kernel function on pairs of points of the training data set, such that $K_{ij} = k\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right), i, j \in \{1, \ldots, N\}$. Multiple kernels have been used in SVM. In this work, we consider the Gaussian kernel defined by

$$k\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right) = \prod_{l=1}^{M} \exp\left[-\frac{1}{2}\left(\frac{x_l^{(i)} - x_l^{(j)}}{\theta_l^2}\right)^2\right]. \tag{13}$$

The hyperparameters of this model are the penalty term C which controls the penalty incurred for misclassifying a training point and the kernel parameter $\boldsymbol{\theta}$ which controls, among others, the smoothness of the separating hyperplane. They are both estimated in this work by minimizing the span estimate of the leave-one-out error (Vapnik and Chapelle, 2000; Chapelle et al., 2002) using the covariance-matrix adaptation evolution scheme (CMA-ES) (See Arnold and Hansen (2012); Moustapha et al. (2018, 2021) for details).

3.2.2 Extension to multi-class classification

Let us now consider the case when the classification task aims at categorizing data with a set of $K > 2$ labels, where each label is defined as $\ell^{(i)} = \ell_k$ if the original training pair $\left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}$ belongs to the cluster $\mathcal{C}_k$.

The most popular approach to tackle this multi-class problem is to reduce it to a series of binary classification problems that can be solved using a standard SVM algorithm. The two most popular approaches are the *one-against-all* and the *one-vs-one* decomposition schemes (Hastie and Tibshirami, 1997; Moreira and Mayoraz, 1998). In the former, one binary problem is derived for each class k by assigning one label, say the positive one, to all samples such that $\ell^{(i)} = \ell_k$ and the negative label to all the other samples. In the one-vs-one approach, binary classifiers considering all pairs of labels and ignoring all other samples are built. This leads to a total of $K(K-1)/2$ classifiers, which is larger than the K classifiers required by the one-against-all

approach. However, such classifiers are trained on a noticeably smaller subset of the training samples making the overall procedure computationally efficient despite the larger number of classifiers to build.

Both approaches can be generalized, or somehow combined, using concepts of the *error correcting output codes* (ECOC) (Dieterich and Bakiri, 1995). The recombination of the binary classifiers into a final one can be achieved either by a simple voting system or by considering the posterior probabilities derived from each classifier. In this work, we consider the one-vs-one approach with a final voting system thanks to its simplicity and efficiency. We note that in case of equal voting between two classes, we heuristically choose the class that was predicted with the classifier that considered the two classes of interest.

3.2.3 Posterior probabilities

As mentioned in Section 2, the soft recombination of the final predictor requires some weights which are proportional to the probability that the sample belongs to a given class. In case of SVM, such weights can be derived by computing posterior probabilities derived from the classifier. In practice, this can be achieved by post-processing the output of the classifier using a sigmoid function as proposed by Platt (2000):

$$\mathbb{P}\left(\ell\left(\boldsymbol{x}\right)=1|\mathcal{M}^{\mathrm{SVC}}\left(\boldsymbol{x}\right)\right)=\frac{1}{1+\exp\left(A\,\mathcal{M}^{\mathrm{SVC}}\left(\boldsymbol{x}\right)+B\right)}, \tag{14}$$

where the coefficients A and B are calibrated by solving a regularized maximum likelihood problem. In this work, we use an efficient numerical implementation proposed by Lin et al. (2007).

There have been many attempts to extend these probabilities to multi-class problems (Hastie and Tibshirami, 1997; Moreira and Mayoraz, 1998; Wu et al., 2004; Wang, 2008). Let us denote by

$$p_{ij}=\mathbb{P}\left(\boldsymbol{x}\in\mathcal{C}_i|\,\boldsymbol{x}\in\mathcal{C}_i\cup\mathcal{C}_j\right) \tag{15}$$

the posterior probability provided by the classifier that discriminates between the classes $\mathcal{C}_i$ (positive) and $\mathcal{C}_j$ (negative). Note however that we are interested in the overall probability of belonging to a class given all possible classes, *i.e.* $p_i=\mathbb{P}\left(\boldsymbol{x}\in\mathcal{C}_i\right)$. Moreira and Mayoraz (1998) proposed estimating this probability by combining the partial ones, *i.e.*,

$$\widehat{p_i}=\frac{2}{k(k-1)}\sum_{j\neq i,j=1}^{K}p_{ij} \tag{16}$$

This value is however flawed, as it accounts for spurious probabilities defined by classifiers discriminating two classes, none of which being the true one.

Using Bayes theorem, it can however be noted that

$$p_i=\mathbb{P}\left(\boldsymbol{x}\in\mathcal{C}_i\right)=\mathbb{P}\left(\boldsymbol{x}\in\mathcal{C}_i|\,\boldsymbol{x}\in\mathcal{C}_i\cup\mathcal{C}_j\right)\mathbb{P}\left(\boldsymbol{x}\in\mathcal{C}_i\cup\mathcal{C}_j\right), \tag{17}$$

which, by averaging over all combinations of i and j, leads to the following system of equations:

$$p_i = \frac{1}{k-1} \sum_{j \neq i, j=1}^{K} p_{ij} \left(p_i + p_j \right), \tag{18}$$

since $\mathbb{P}\left(\boldsymbol{x} \in \mathcal{C}_i \cup \mathcal{C}_j\right) = (p_i + p_j)$. Wu et al. (2004) noted that this system of equations can be written in a matrix form

$$\boldsymbol{p} = \boldsymbol{T}\boldsymbol{p}, \tag{19}$$

where $\boldsymbol{p} = \{p_1, \ldots, p_K\}^T$ and $\boldsymbol{T}$ is a $K \times K$ matrix whose elements read

$$T_{ij} = \begin{cases} \frac{1}{k-1} p_{ij} & \text{if } i \neq j, \\ \frac{1}{k-1} \sum_{j \neq i, j=1}^{K} p_{ij} & \text{if } i = j. \end{cases} \tag{20}$$

Wu et al. (2004) then noted that there exists a finite Markov chain whose transition matrix is $\boldsymbol{T}$, since $\sum_{j=1}^{K} T_{ij} = 1$ and $0 \leq T_{ij} \leq 1$. Further assuming that $p_{ij} > 0$ for any i, $j \in \{1, \ldots, K\}$ implies that $T_{ij} > 0$, which ensures that the Markov chain is irreducible and aperiodic. In fine, these conditions guarantee that Eq. (19) defines a Markov chain whose stationary distribution exists and is unique.

Taking advantage of the fact that $\boldsymbol{T}$ is a transition kernel and $\boldsymbol{p}$ is the stationary distribution of the corresponding Markov chain, we cast Eq. (18) in an iterative scheme

$$p_i^{(t+1)} = \frac{1}{k-1} \sum_{j \neq i, j=1}^{K} p_{ij} \left(p_i^{(t)} + p_j^{(t)} \right), \tag{21}$$

where the initial values $p_i^{(0)}$, $i = \{1, \ldots K\}$ using the estimate in Eq. (16) and p_{ij} are the partial probabilities obtained by the binary one-vs-one classifiers using Eq. (14) . This chain eventually converges after a few iterations, generally with $t < 100$ in our examples, to the posterior probabilities estimates.

3.3 Regression using Kriging

3.3.1 Basics of Kriging

The final ingredient considered in the proposed framework is Kriging a.k.a. Gaussian process model. It is used here to build local surrogates in the different regions identified by the clustering step.

A Kriging model assumes that the model to approximate is of the form (Santner et al., 2003; Rasmussen and Williams, 2006)

$$\mathcal{M}\left(\boldsymbol{x}\right) = \sum_{j=1}^{p} \beta_j f_j\left(\boldsymbol{x}\right) + Z\left(\boldsymbol{x}\right), \tag{22}$$

where the first summand represents the *trend* written here in a polynomial form using p regressors f_j with corresponding coefficients β_j. The second summand is a zero-mean stationary covariance

process defined by an auto-covariance function $\mathrm{Cov}\left[Z\left(\boldsymbol{x}\right), Z\left(\boldsymbol{x}'\right)\right] = \sigma^2 R\left(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}\right)$ where σ^2 is the process variance and R is an auto-correlation function parameterized by the vector $\boldsymbol{\theta}$. In this work, we consider an anisotropic Matérn 5/2 auto-correlation function defined by

$$R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right) = \prod_{l=1}^{M} \left[\left(1 + \sqrt{5} \frac{\left| x_l^{(i)} - x_l^{(j)} \right|}{\theta_l} + \frac{5}{3} \left(\frac{\left| x_l^{(i)} - x_l^{(j)} \right|}{\theta_l} \right)^2 \right) \exp\left(-\sqrt{5} \frac{\left| x_l^{(i)} - x_l^{(j)} \right|}{\theta_l} \right) \right]. \tag{23}$$

The calibration of the model is performed by estimating the regression coefficients of the trend and the hyperparameters of the selected kernel that minimize a generalization error, herein using a maximum likelihood approach (Santner et al., 2003; Bachoc, 2013; Lataniotis et al., 2018).

Following this step, Kriging assumes that any unknown sample actually follows a normal distribution $\mathcal{N}\left(\mu_{\widehat{\mathcal{M}}}, \sigma^2_{\widehat{\mathcal{M}}}\right)$ where the mean is the actual prediction, while the standard deviation informs about the local accuracy of the prediction. The two quantities respectively read

$$\begin{aligned} \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right) &= \boldsymbol{f}^T\left(\boldsymbol{x}\right)\widehat{\beta} + r\left(\boldsymbol{x}\right)\boldsymbol{R}^{-1}\left(\mathcal{Y} - \boldsymbol{F}\widehat{\beta}\right), \\ \sigma^2_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right) &= \widehat{\sigma}^2\left(1 - \boldsymbol{r}\left(\boldsymbol{x}\right)^T \boldsymbol{R}^{-1}\boldsymbol{r}\left(\boldsymbol{x}\right) + \boldsymbol{u}\left(\boldsymbol{x}\right)^T\left(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{u}\left(\boldsymbol{x}\right)\right), \end{aligned} \tag{24}$$

where

- $\boldsymbol{u}\left(\boldsymbol{x}\right) = \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}\left(\boldsymbol{x}\right) - \boldsymbol{f}\left(\boldsymbol{x}\right)$ has been introduced for convenience,
- $\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\mathcal{Y}$ is the generalized least-square estimate of the regression coefficients $\boldsymbol{\beta}$,
- $\widehat{\sigma}^2 = \frac{1}{N}\left(\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)^T \boldsymbol{R}^{-1}\left(\mathcal{Y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)$ is the estimate of the process variance,
- $\boldsymbol{F} = \left\{f_j\left(\boldsymbol{x}^{(i)}\right),\, j = 1, \ldots, p,\, i = 1, \ldots, n_0\right\}$ is the Vandermonde matrix,
- $\boldsymbol{R}$ is the correlation matrix with $R_{ij} = R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right)$,
- $\boldsymbol{r}\left(\boldsymbol{x}\right)$ is a vector gathering the correlation between the unknown sample $\boldsymbol{x}$ and the experimental design points and
- $\mathcal{Y} = \left\{\mathcal{Y}^{(i)} = \mathcal{M}\left(\boldsymbol{x}^{(i)}\right), i = 1, \ldots, n_0\right\}$ is the vector of available model responses.

To account for the categorical variable, the *compound symmetry* kernel defined by Pelematti et al. (2020)

$$R\left(\ell^{(i)}, \ell^{(j)}\right) = \begin{cases} 1 & \text{if} \quad \ell^{(i)} = \ell^{(j)}, \\ r & \text{if} \quad \ell^{(i)} \neq \ell^{(j)}, \end{cases} \tag{25}$$

is considered. The parameter r is computed here by embedding this kernel within a usual auto-correlation function for continuous variables with a tunable parameter θ_{cat} that can be calibrated in the same setting than the continuous parameters. More precisely, we consider a Gaussian kernel which then reads:

$$R\left(\ell^{(i)}, \ell^{(j)}; \theta_{\mathrm{cat}}\right) = \exp\left(-\frac{1}{2}\left(\frac{S_{\ell^{(i)},\ell^{(j)}}}{\theta_{\mathrm{cat}}}\right)^2\right), \tag{26}$$

where $S_{\ell^{(i)},\ell^{(j)}} = 0$ if $\ell^{(i)} = \ell^{(j)}$ and 1 otherwise. The final auto-correlation function is obtained by multiplying the $M+1$ one-dimensional auto-correlation functions *i.e.*,

$$R\left(\widetilde{\boldsymbol{x}}^{(i)}, \widetilde{\boldsymbol{x}}^{j}, \widetilde{\boldsymbol{\theta}}\right) = \exp\left(-\frac{1}{2}\sum_{k=1}^{M}\left(\frac{\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}}{\theta_k}\right)^2 - \frac{1}{2}\left(\frac{S_{\ell^{(i)},\ell^{(j)}}}{\theta_{\text{cat}}}\right)^2\right), \tag{27}$$

where $\widetilde{\boldsymbol{\theta}} = \{\boldsymbol{\theta}, \theta_{\text{cat}}\}$ and $\widetilde{\boldsymbol{x}}^{(i)} = \left\{\boldsymbol{x}^{(i)}, \ell^{(i)}\right\}$.

4 Examples

The proposed algorithm is illustrated in this section with two analytical toy functions and an engineering problem related to a tensile membrane structure design. To assess its accuracy, we estimate the following two generalization errors using a validation set of size $N_{\text{val}} = 10^4$:

- Normalized mean-square error:

$$NMSE = \frac{\sum_{i=1}^{N_{\text{val}}}\left(\mathcal{Y}_i - \widehat{\mathcal{Y}}_i\right)^2}{\sum_{i=1}^{N_{\text{val}}}\left(\mathcal{Y}_i - \bar{\mathcal{Y}}\right)^2}, \tag{28}$$

- Mean absolute error:

$$MAE = \frac{1}{N_{\text{val}}}\sum_{i=1}^{N_{\text{val}}}\left|\mathcal{Y}_i - \widehat{\mathcal{Y}}_i\right|. \tag{29}$$

Furthermore, each analysis is repeated 20 times in order to assess the robustness of the proposed algorithm with respect to the statistical uncertainty associated with the experimental designs.

4.1 Manhattan function

For this first validation example, we consider a two-dimensional function proposed by Rai (2015). The function consists of three global regions, one of which is a checkerboard, and reads

$$\mathcal{M}(\boldsymbol{x}) = \begin{cases} \text{Checker board} & \text{if } x_1 \geq 0, \\ \sin(7x_1)\cdot\sin(4x_2)\,; & \text{if } x_1 \leq 0 \text{ and } x_2 \leq 0, \\ 1 + \frac{2}{7}(2x_1+1)^2 + (2x_2+1)^2; & \text{if } x_1 \leq 0 \text{ and } x_2 \geq 0 \end{cases}$$

The checkerboard is made of smaller rectangular regions alternating the values of 0 and 1 as illustrated in Figure 3.

In this section, we will illustrate each of the three steps of the proposed algorithm. We first start by showing how the clustering algorithm splits the data. Figure 4 shows the clusters identified using three experimental designs of different sizes. The original model is built assuming 10 regions where each of the squares in the checkerboard is considered as one region on its own. However, regardless of the experimental design, the clustering algorithm reduces the checkerboard into two regions, one with $y = 1$ and the other with $y = 0$. This results in disconnected

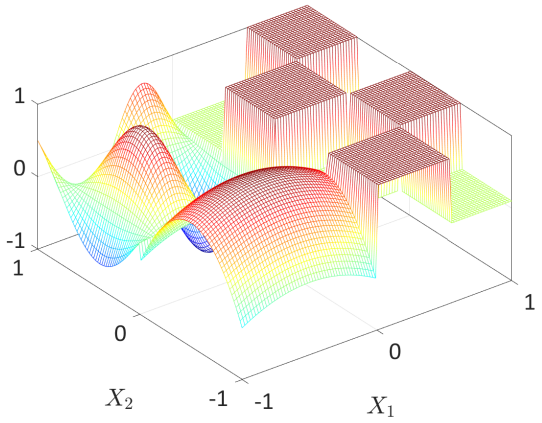
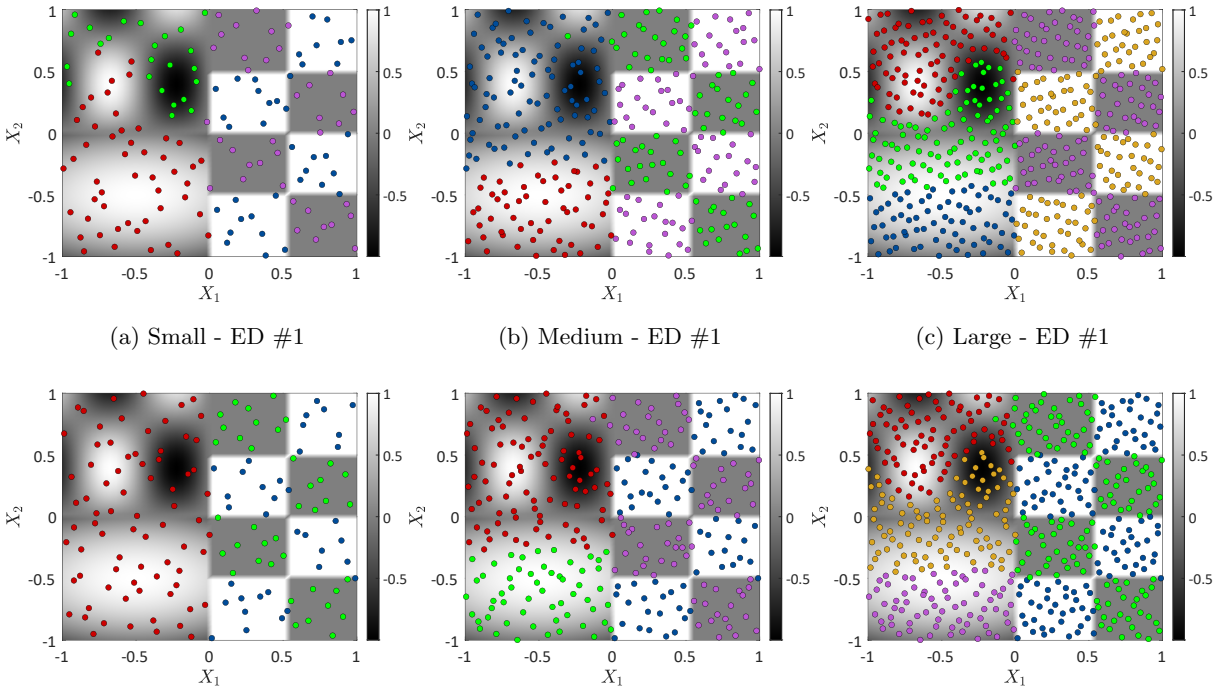


Figure 3: Example 1 - Three-dimensional representation of the Manhattan function.

subdomains but as we will see in the next paragraph, this does not affect the overall prediction capability of the algorithm. Another important observation from the partitions in Figure 4 is that the more data points, the more clusters are identified. For small datasets, the partition is quite sensitive to the data. However, the partition becomes more stable and robust as the data size is increased.

Once the clusters are identified (4 different ones in the case of medium-size experimental design, and in the sequel), the inputs are labelled accordingly and binary classification is performed on each pair of classes. Figure 5 shows the resulting classifiers for one realization of the experimental design. The blue and orange points correspond to the positive and negative labels respectively, while the support vectors are highlighted in green. The thick line is the classifier, whereas the dashed ones delimit the margin. Finally, the gray triangles represent the data points that were ignored by the illustrated classifier. As expected, support vector machines are appropriately calibrated for the problem at hand. However, the choice of the Gaussian kernel may not be appropriate for the classification of $\mathcal{C}_3$ against $\mathcal{C}_4$ (Figure 5f) as it produces smooth boundaries whereas the original boundary results from a checkerboard with sharp edges. This does not substantially affect the results. However, better prediction could have been obtained by including the choice of the kernel in the model selection.

The next step is then to recombine those predictions into a final one. In the hard reconstruction, a vote is carried out and the class that wins is the final prediction. The resulting partition of the input space is shown in Figure 6. Figure 7 shows the soft reconstruction approach where each tile represents the probabilities of a given point to belong to a given class. The resulting classification is in accordance with the regions defined by the original model except for the boundaries of the checkerboard which present some slight deviations. Also, the boundary between the two regions where $\mathcal{M}$ is smooth (*i.e.* , polynomial or sines) is not exactly the line



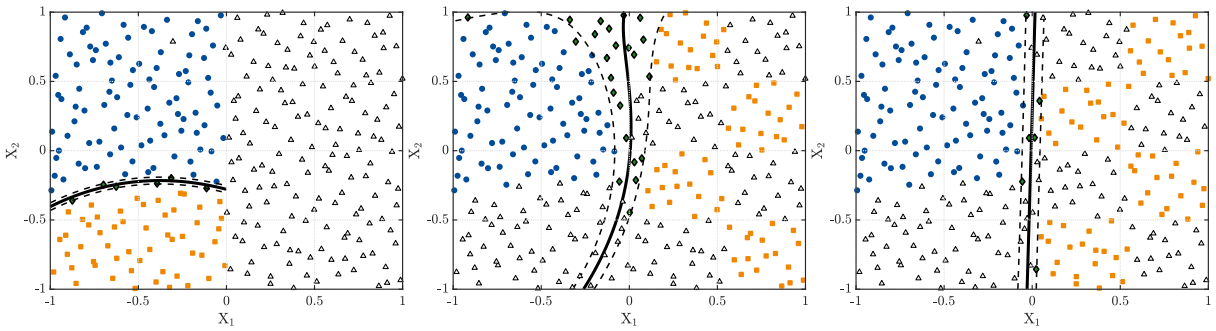
(a) Small - ED #1 (b) Medium - ED #1 (c) Large - ED #1

(d) Small - ED #2 (e) Medium - ED #2 (f) Large - ED #2

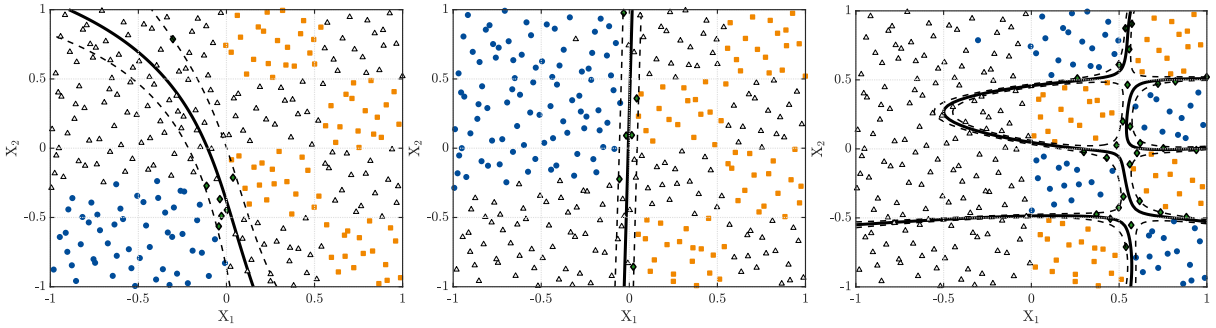
Figure 4: Example 1 - Clustering of the data by DPMM considering two repetitions of three experimental designs of increasing sizes.

$\{x_1 \leq 0, x_2 = 0\}$.

This partition of the input space is eventually used to build local Kriging surrogates to provide the final prediction. For this example, we repeat the analysis 20 times where each repetition starts with a randomly sampled Sobol' sequence. Figure 8 shows boxplots of the resulting errors for increasing sizes of the experimental design. For any ED size, both recombination techniques yield improved $NMSE$ and MAE. In general, the soft reconstruction also yields better prediction than the hard one. This is even more clear when considering the MAE error. For this example, the prediction with categorical Kriging is not included, since it does not lead to good results. This is due to the fact that each region is fundamentally different from the other, hence using a single Kriging model, even with categorical variables, is not appropriate.



(a) $\mathcal{C}_1$ vs. $\mathcal{C}_2$ (b) $\mathcal{C}_1$ vs. $\mathcal{C}_3$ (c) $\mathcal{C}_1$ vs. $\mathcal{C}_4$



(d) $\mathcal{C}_2$ vs. $\mathcal{C}_3$ (e) $\mathcal{C}_2$ vs. $\mathcal{C}_4$ (f) $\mathcal{C}_3$ vs. $\mathcal{C}_4$

Figure 5: Example 1 - Pairwise classification of the data (with 4 clusters identified in Step 1 for the medium-size experimental design).

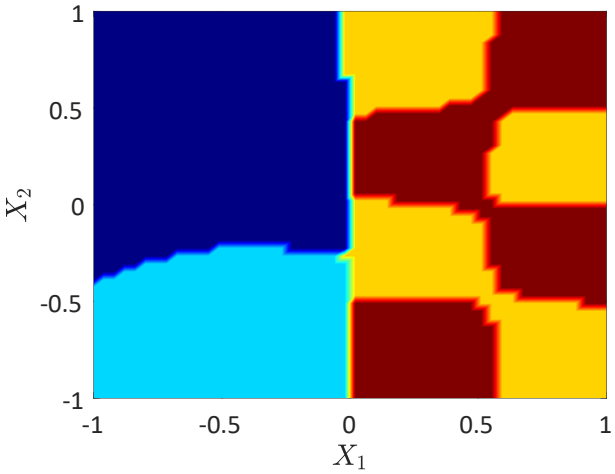
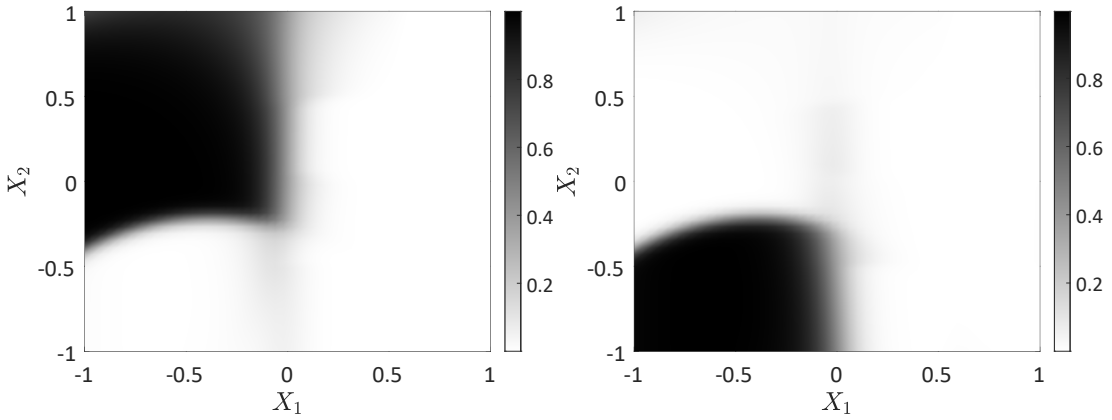


Figure 6: Example 1 - Partition of the space in the 4 regions using hard reconstruction.



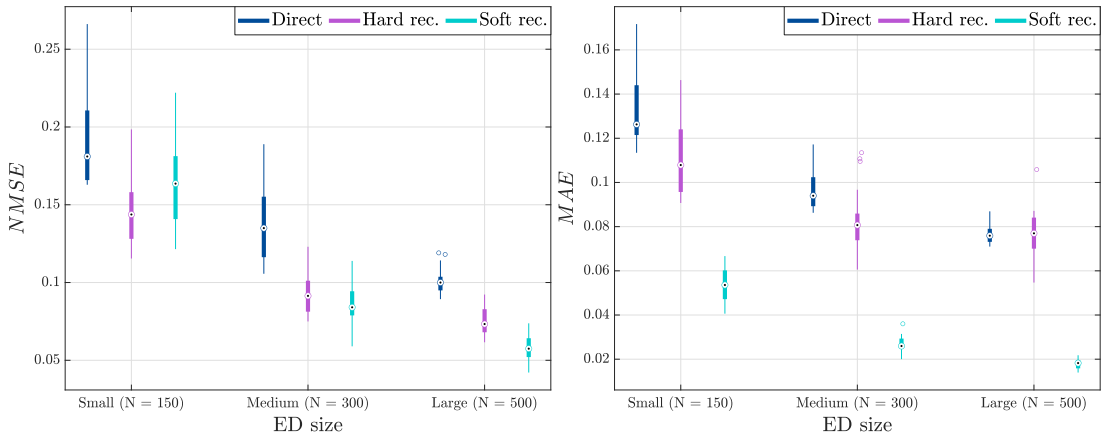
(a) Prob $[Y \in \mathcal{C}_1]$

(b) Prob $[Y \in \mathcal{C}_2]$

(c) Prob $[Y \in \mathcal{C}_3]$

(d) Prob $[Y \in \mathcal{C}_4]$

Figure 7: Example 1 - Partition of the space in the 4 regions using soft reconstruction.



(a) $NMSE$ (b) MAE

Figure 8: Example 1: Boxplots of the computed errors for various methods and experimental design sizes.

4.2 Snap-through instability problem

This example is a mechanical problem related to the snap-through instability of a two-bar truss structure. The structure is loaded at its tip and responds linearly with small displacements until a critical point is reached. Past that point, the structure suddenly snaps through a new equilibrium point and resumes its small displacements. In this example, we consider as quantity of interest the displacement w of the tip of the structure as illustrated in Figure 9 .

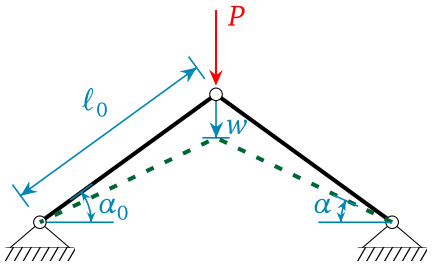


Figure 9: Illustration of the two-bar truss structure subject to snap-through.

The load at the deformed position can be expressed as a function of the inclination angles at the initial position and deformed one, respectively denoted by α_0 and α, the bars cross-sectional areas A and their constitutive material Youngs's modulus E

$$P = -2EA\tan(\alpha)\left(\cos(\alpha_0) - \cos(\alpha)\right). \tag{30}$$

The corresponding displacement of the tip of the truss can then be computed as follows:

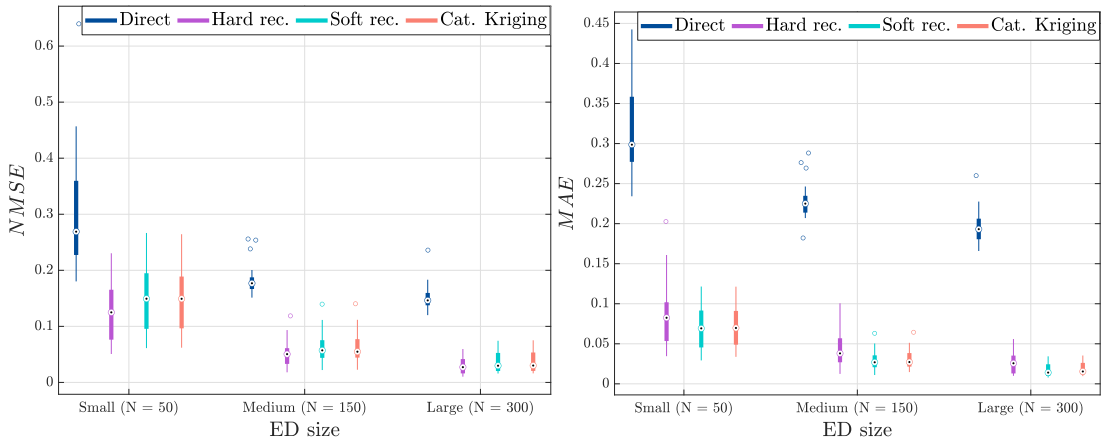
$$w = l_0\cos(\alpha_0)\left(\tan(\alpha_0) - \tan(\alpha)\right). \tag{31}$$

In this example, we assume that the length of the bar $l_0 = 5$ m and the initial inclination angle $\alpha_0 = 10°$ are deterministic. In contrast, the load, the Young's modulus and the cross section areas are assumed random and characterized by the distributions shown in Table 1.

Parameter	Distribution	Mean	C.o.V.
Load (P in N)	Gumbel	430	0.20
Young's modulus (E in GPa)	Lognormal	210	0.10
Cross sectional area (A in cm^2)	Gaussian	10	0.05

Table 1: Truss snap-through problem: probabilistic input model.

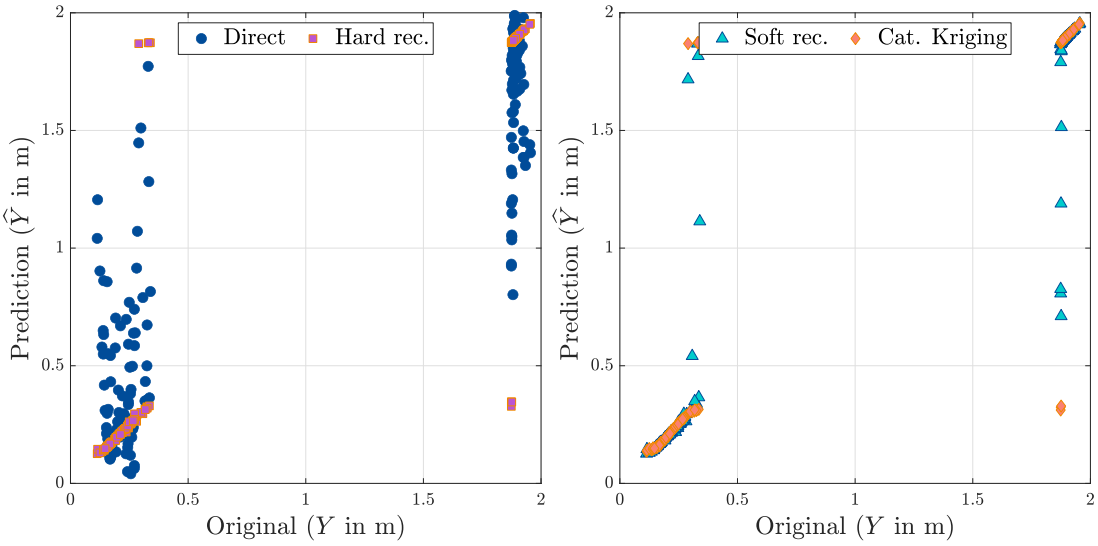
We run the analysis using the proposed method and considering three different experimental design sizes and 20 repetitions. The resulting errors are summarized as boxplots shown in Figure 10. The first observation is that the difference between the results obtained by the proposed method and a direct Kriging model (*i.e.* a single Kriging model built using the entire data set) is much more important than in the previous case, often by orders of magnitude. This is due to the fact that the two regimes of non-linear structure behaviours are prominently different as shown in Figure 11. Furthermore in this example, categorical Kriging performs quite well. It is not clear however which recombination approach is the best. When looking at the normalized mean square error, the hard recombination is slightly better. This is the opposite when looking at the mean absolute error, *i.e.*, the soft and categorical recombination are slightly better.



(a) $NMSE$ (b) MAE

Figure 10: Example 2: Boxplots of the computed errors for various methods and experimental design sizes.

Figure 11 shows the original *vs.* predicted vertical displacement for the four approximations using a random subset of the validation set of size 200. The left panel of this figure shows how a single model (called "direct") spans the entire range between the two regimes of the truss and leads to huge errors. In contrast, the multi-stage approaches properly detect the discontinuities. It is also clear from this figure how the recombination scheme affects the final prediction when there are classification errors. The soft recombination reduces the error for those cases when there is uncertainty in the classification. Note that the same outlier points are observed in Figures 11a and 11b when hard reconstruction and categorical Kriging are used: these outliers only stem from classification error.



(a) Direct Kriging and hard reconstruction

(b) Soft reconstruction and categorical Kriging

Figure 11: Example 2: Original *vs.* predicted vertical displacement for different approximation techniques.

4.3 Tensile fabric structure

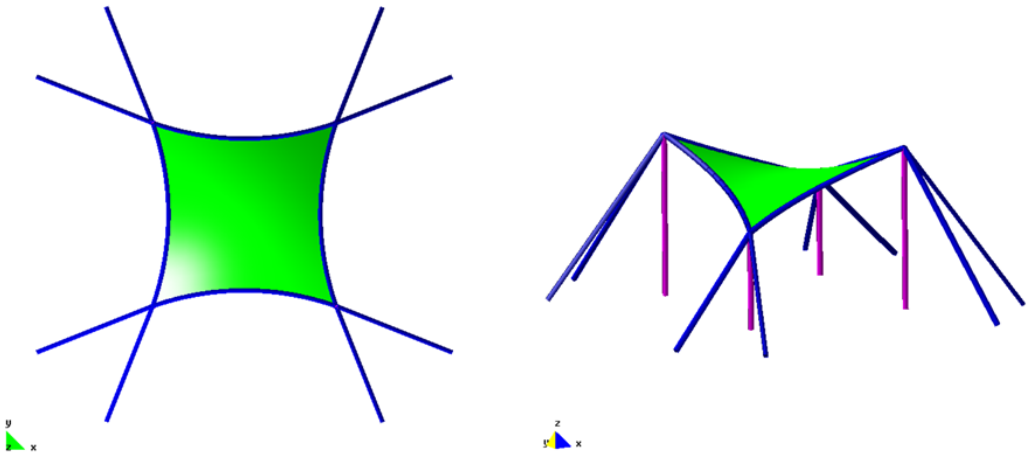
In this final example, we investigate a model that simulates the behaviour of a tensile membrane structure (TMS) under extreme loading (Valdés-Vázquez et al., 2020, 2021). TMS are flexible lightweight structures made of composite fabric spanning long distances. They have many advantages in terms of architectural sophistication but are yet challenging to design. By their very nature, they are unable to carry out-of-plane moments and shear forces that may result from the extreme wind loads they are expected to withstand. They further require careful pre-stressing to keep a stable form.

Special codes are designed to simulate the response of complex tensile membrane structures. Comet is one such in-house finite element code developed at the University of Gua (Valdés-Vázquez et al., 2021). In this work, we consider a hypar (hyperbol-paraboloid), which is one of the most common shapes for TMS, designed using Comet and illustrated in Figure 12. The probabilistic model is described using the random variables presented in Table 2. There are various quantities of interest for such a design model. We consider here the maximum reaction forces on the supports of the system (cables or mast). It turns out that according to the boundary conditions, the maximum reaction force occurs in two different locations with entirely different magnitudes. This is shown by the bi-modality of the kernel density estimate of the model response in Figure 13.

The underlying mechanisms leading to each of two model response modes are different and

Parameter	Distribution	Mean	C.o.V.
Wind load (V_w - m/s)	Gumbel	36.11	0.132
Cable pre-stress (S_{xx} - N/m^2)	Gaussian	$5.09 \cdot 10^8$	0.06
Young's modulus (E_{wf} - N/m)	Lognormal	$8 \cdot 10^5$	0.07
Poisson modulus (ν -)	Gaussian	0.4	0.05
Fabric prestress warp (F_w - N/m^2)	Gaussian	$4 \cdot 10^6$	0.05
Fabric prestress fill (F_f - N/m^2)	Gaussian	$4 \cdot 10^6$	0.05
Mast Young's modulus (E_m - N/m^2)	Lognormal	$2.1 \cdot 10^{11}$	0.03
Cables Young's modulus (E_c - N/m^2)	Lognormal	$2.1 \cdot 10^{11}$	0.03
Mast cross-sectional area (A_m - m^2)	Gaussian	$1.7 \cdot 10^{-3}$	0.032
Cable cross-sectional area (A_c - m^2)	Gaussian	$7.854 \cdot 10^{-5}$	0.032

Table 2: Hypar structure: probabilistic input model.



(a) Top view

(b) Side view

Figure 12: Hypar structure considered in this study.

building a single surrogate model to account for both leads to inaccurate results. We consider then the three-stage approach proposed in this paper, with an experimental design of size 500 and a validation set of size $1,000$. The experimental design is split into five different subsets of sizes 100, 200, 300, 400 and 500. In each of these, the DPMM clustering rightly identifies that there are two sets of responses.

Figure 14 shows the resulting NMSE and MAE for each experimental design size. As ex-

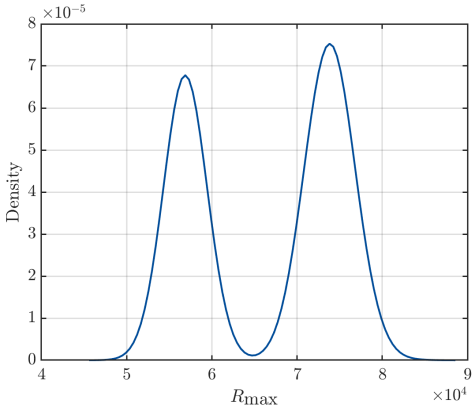
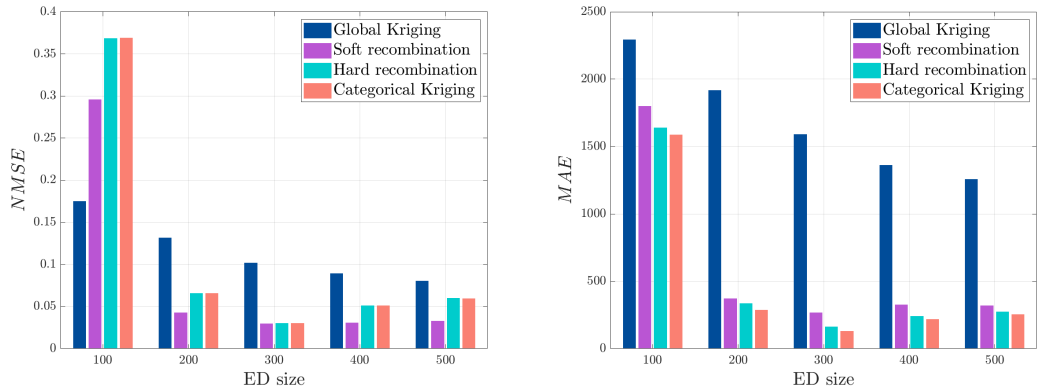


Figure 13: Example 3: Kernel smoothing density of the maximum reaction force of the hypar.

pected, the error decreases with increasing ED size and our proposed workflow yields more accurate approximations than a global single Kriging model, except for NMSE when $N = 100$ due to the large weight of misclassification errors. The soft recombination is slightly better than hard recombination and categorical Kriging which have very similar predictions.

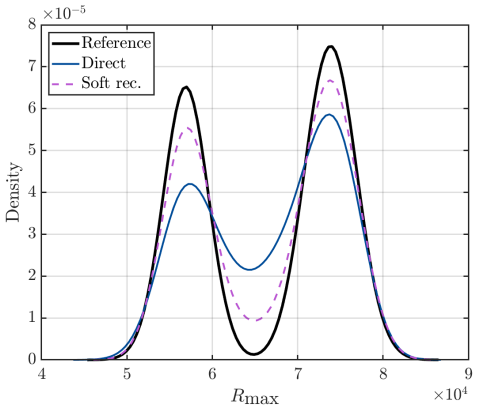


(a) $NMSE$ (b) MAE

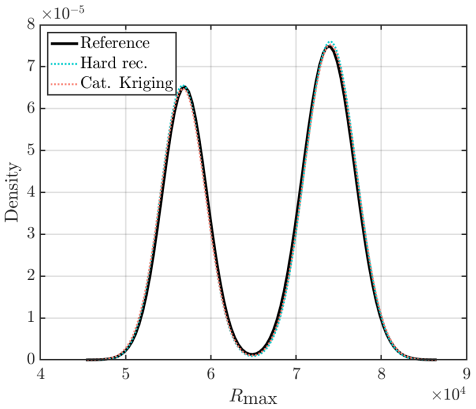
Figure 14: Example 3: Computed errors for the hypar structure for increasing experimental design sizes.

Finally, Figure 15 shows PDFs of the responses for different models with ED sizes of 100 and 300. We can see that even for 100 samples, the densities with the hard recombinations are extremely similar to those obtained from the original model. This shows that the reconstructed surrogate models are extremely accurate except for a few outliers which are due to misclassification in the second step of the workflow. The soft recombination puts more mass in the middle of the density support, due to the weighted recombination. This mass reduces as the ED size

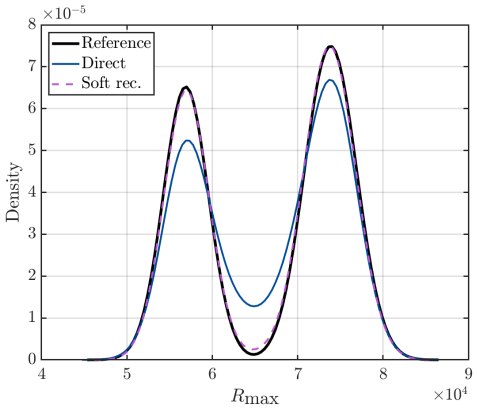
increases.



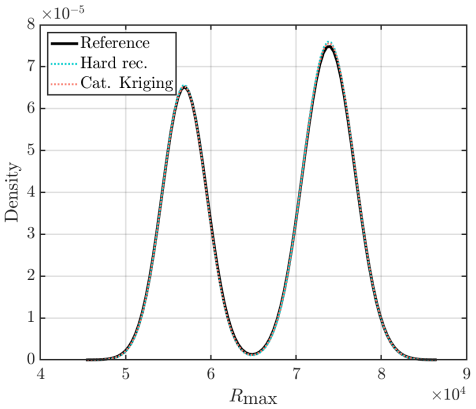
(a) $N = 100$ - Direct and soft recombination



(b) $N = 100$ - Hard recombination and categorical Kriging



(c) $N = 300$ - Direct and soft recombination



(d) $N = 300$ - Hard recombination and categorical Kriging

Figure 15: Example 3: Computed errors for the hypar structure for increasing experimental design sizes.

5 Conclusion

Surrogate modelling is now a well-established method that allows one to reduce the computational burden of simulation intensive methods that require multiple evaluations of a costly computational model. Building an accurate surrogate model with limited data generally requires that the functions to approximate are smooth and regular. This is however not always the case in many applications, *e.g.* crash simulation or computational fluid dynamics.

In this paper, we propose a three-stage approach for the approximation of non-smooth functions for systems exhibiting multiple behaviours and/or discontinuities. The problem is tackled by dividing the task into three complementary parts: i. a joint input-output clustering stage that identifies the different patterns exhibited by the system using a non-parametric Bayesian approach, namely a Dirichlet process mixture model, ii. a partition of the input space according to the identified clusters using support vector machines, and eventually iii. the construction of local surrogates, herein Kriging models, using data from each of the partitions. For any new point, the prediction is made by appropriately recombining the predictions made by each of the Kriging models, according to the assigned class of the new point.

The proposed approach is validated on two analytical examples and an engineering application (FE-based tensile membrane structure). It is shown to be both accurate and efficient compared to a traditional surrogate modelling approach ignoring the non-smoothness.

The three methods selected for each stage all provide probabilistic predictions. While the posterior probabilities of the support vector machines classifiers have been used within the soft reconstruction scheme, the ones provided by the Dirichlet process mixture models have not been exploited yet. However, as seen in the examples, mislabelling the initial data leads to large errors. These could be reduced by accounting for the uncertainties in the clustering stage. In a future work, we intend to account for the latter so as to provide a fully probabilistic prediction scheme that propagates the epistemic uncertainties from one step to the next.

References

Aldous, D. J. (1985). *Exchangeability and related topics*, Volume 117 of *École d'été de probabilités de Saint-Flour XIII — 1983. Lecture Notes in Mathematics*. Springer, Berlin, Heidelberg.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics 2*(6), 1152–1174.

Arnold, D. V. and N. Hansen (2012). A (1+1)-CMA-ES for constrained optimisation. In T. Soule and J. H. Moore (Eds.), *Proc. of the Genetic and Evolutionary Computation Conference 2012 (GECCO 2012)*, pp. 297–304.

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

Basudhar, A. and S. Missoum (2008). Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. *Computers & Structures 86*(19-20), 1904–1917.

Bernholdt, D., M. R. Cianciosa, D. L. Green, and J. M. Park (2019). Cluster, classify, regress: A general method for learning discontinuous functions. *Foundation of Data Science 1*, 491 – 506.

Blackwell, D. and J. MacQueen (1973). Ferguson distribution via Pólya urn schemes. *The Annals of Statistics 1*(2), 353–355.

Blatman, G. and B. Sudret (2011). Adaptive sparse polynomial chaos expansion based on Least Angle Regression. *Journal of Computational Physics 230*, 2345–2367.

Blei, D. and M. I. Jordan (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis 1*, 121 – 144.

Boroson, E. and S. Missoum (2017). Stochastic optimization of nonlinear energy sinks. *Structural and Multidisciplinary Optimization 55*, 633–646.

Calandra, R., J. Peters, C. E. Rasmussen, and M. P. Deisenroth (2016). Manifold Gaussian process regression. In *Proc. of the 2016 international Joint Conference on Neural Networks (IJCNN) , Vancouver, Canada, July 24th-29th, 2016*. Institute of Electrical and Electronics Engineers (IEEE).

Chapelle, O., V. Vapnik, and Y. Bengio (2002). Model selection for small sample regression. *Machine Learning 48*(1), 9–23.

Chatterjee, T., S. Chakraborty, and R. Chowdhury (2019). A critical review of surrogate assisted robust design optimization. *Archives of Computational Methods in Engineering 26*(1), 245–274.

Dieterich, T. G. and G. Bakiri (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research 2*, 263 – 286.

Dupuis, R., J.-C. Jouhaud, and P. Sagaut (2018). Surrogate modeling of aerodynamic simulations for multiple operating conditions using machine learning. *AIAA Journal 56*(9), 3622–3635.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*, 209–230.

Ghahramani, Z. and M. J. Beal (2000). Propagation algorithms for variational Bayesian learning. In M. Papadrakakis, V. Papadopoulos, and G. Stefanou (Eds.), *Advances in Neural Information Processing Systems 13, Denver, Colorado, USA, November 28-30*.

Gorodetsky, A. A. (2012). A learning method for the approximation of discontinuous functions for stochastic simulations. Msc thesis, Massachusetts Institute of Technology.

Hastie, T. and R. Tibshirami (1997). Classification by pairwise coupling. In M. Jordan, M. Kearns, and S. Solla (Eds.), *Advances in Neural Information Processing Systems 10, Denver, Colorado, USA*.

Heinonen, M., H. Mannerström, J. Rousu, S. Kaski, and H. Lähdesmäki (2016). Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. In A. Gretton and C. C. Robert (Eds.), *Proc. of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS), Cadiz, Spain, May 9th-11th, 2016*.

Ishwaran, H. and L. F. James (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association 96*(453), 161–173.

Konomi, B. A., A. A. Hanandeh, P. Ma, and E. L. Kang (2019). Computationnally efficient nonstationary nearest-neighbor Gaussian process models using data-driven techniques. *Econometrics 30*, 1–20.

Kuleshov, A., A. Bernstein, and E. Burnaev (2018). Manifold learning regression with non-stationary kernels. In L. Pancioni, F. Schwenker, and E. Trentin (Eds.), *Proc. of the 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19–21, 2018*.

Lataniotis, C., S. Marelli, and B. Sudret (2018). The Gaussian process modeling module in UQLab. *Soft Computing in Civil Engineering 2*(3), 91–116.

Le Maître, O. P., O. M. Knio, N. H. Najm, and R. G. Ghanem (2004). Uncertainty propagation using Wienerhaar expansions. *Journal of Computational Physics 224*, 560 – 586.

Li, Y., O. Schofield, and M. Gönen (2019). A tutorial on Dirichlet process mixture modeling. *Journal of Mathematical Psychology 91*.

Lin, H.-T., C.-J. Lin, and R. C. Weng (2007). A note on Platt's probabilistic outputs for support vector machines. *Machine Learning 68*, 267–276.

Lloyd, S. P. (1982). Least squares optimization in PCM. *IEEE Transactions on Informatinon Theory 28*(2), 129–137.

Marmin, S. (2018). *Warping and sampling approaches to non-stationary Gaussian process modelling*. Ph. D. thesis, Ecole Centrale Marseille; Université de Berne.

Meeds, E. and S. Osindero (2005). An alternative infinite mixture of Gaussian process experts. In B. Weiss, B. Schölkpf, and J. Platt (Eds.), *Advances in Neural Information Processing Systems 18 (NIPS 2005), Vancouver, British Columbia, Canada, December 5th-8th, 2005*.

Moreira, M. and E. Mayoraz (1998). Improved pairwise coupling classification with correcting classifiers. In C. Nédellec and C. Rouveirol (Eds.), *Proc. 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23*.

Moustapha, M. (2016). *Adaptive surrogate models for the reliable lightweight design of automotive body structures*. Ph. D. thesis, Université Blaise Pascal, Clermont-Ferrand, France.

Moustapha, M., C. Lataniotis, S. Marelli, and B. Sudret (2021). UQLab user manual – Support vector machines for regression. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report # UQLab-V1.4-111.

Moustapha, M., S. Marelli, and B. Sudret (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety 96*, 102714.

Moustapha, M. and B. Sudret (2019a). Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework. *Structural and Multidisciplinary Optimization 60*, 2157–2176.

Moustapha, M. and B. Sudret (2019b). A two-stage surrogate modelling approach for the approximation of functions with non-smooth outputs. In M. Papadrakakis, V. Papadopoulos, and G. Stefanou (Eds.), *Proc. 3rd Int. Conf. Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP), Crete Island (Greece), June 24-26*.

Moustapha, M., B. Sudret, J.-M. Bourinet, and B. Guillaume (2018). Comparative study of Kriging and support vector regression for structural engineering applications. *ASCE-ASME Journal Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 4*(2). Paper #04018005.

Myers, R. H. and D. C. Montgomery (2002). *Response surface methodology: process and product optimization using designed experiments* (2nd ed.). J. Wiley & Sons.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics 9*(2), 249–265.

Niutta, C. B., E. J. Wehrle, F. Duddeck, and G. Belingardi (2018). Surrogate modeling in design optimization of sources with discontinuous responses. *Structural and Multidisciplinary Optimization 57*, 1857 – 1869.

Paciorek, C. J. and M. J. Schervish (2003). Nonstationary covariance functions for Gaussian process regression. In S. Thrun, L. K. Saul, and B. Schölkpf (Eds.), *Proc. of the 16th International Conference on Neural Information Processing Systems (NIPS 03), Whistler British Columbia, Canada, December 9th-13th, 2003.*

Pelematti, J., L. Brevault, M. Balesdent, E.-G. Talbi, and Y. Guerin (2020). Bayesian optimization of variable-size design space problems. *Optimization and Engineering 22*, 387–447.

Pham, D. T. and A. A. Afify (2017). Clustering techniques and their applications in engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 221*, 1445–1459.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans (Eds.), *Advances in large margin classifiers.*

Rai, P. (2015). *Sparse low rank approximation of multivariate functions - Applications in Uncertainty Quantification.* Ph. D. thesis, École Centrale de Nantes.

Rasmussen, C. E. and Z. Ghahramani (2001). Infinite mixture of Gaussian process experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Proc. of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS 01), Vancouver British Columbia, Canada, December 3-8.*

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian processes for machine learning* (Internet ed.). Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press.

Rokach, L. and O. Maimon (2005). *Clustering methods, Data Mining and Knowledge Discovery Handbook*. Springer.

Santner, T. J., B. J. Williams, and W. I. Notz (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Sargsyan, K., C. Safta, B. Debusschere, and H. Najm (2012). Uncertainty quantification given discontinuous model response and a limited number of model runs. *SIAM Journal on Scientific Computing 34*, B44–B64.

Serna, A. and C. Bucher (2009). Advanced surrogate models for multidisciplinary design optimization. In *6th Weimar Optimization and Stochastic Days 2009, Weimar, Germany, October 15th-16th*.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica 4*, 639–650.

Smola, A. J. and B. Schölkopf (2004). A tutorial on support vector regression. *Statistics and Computing 14*, 199–222.

Teixeira, R., M. Nogal, and A. O'Connor (2021, March). Adaptive approaches in metamodel-based reliability analysis: A review. *Structural Safety 89*, 102019.

Tresp, V. (2000). Mixture of Gaussian processes. In T. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Proc. Advances in Neural Information Processing Systems 13, Denver, CO, USA, 2000*.

Valdés-Vázquez, J. G., A. D. García-Soto, and M. Chiumenti (2021). Response of a double hypar fabric structure under varying wind speed using fluid-structure interaction. *Latin American Journal of Solids and Structures 18*(4).

Valdés-Vázquez, J. G., A. D. García-Soto, A. Hernández-Martínez, and J. L. Nava (2020). Fluid-structure interaction of a tensile fabric structure subjected to different wind speeds. *Wind and Structures 31*(6).

Vapnik, V. and O. Chapelle (2000). Bounds on error expectation for support vector machines. *Neural Computation 12*(9), 2013–2036.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Wainwright, M. and M. Jordan (2003). Graphical models, exponential families, and variational inference. Technical Report Technical Report 649, UC Berkeley, Dept. of Statistics.

Wang, X. (2008). Posterior probability reconstruction for multi-class support vector machines. In *2008 International Conference on Computational Intelligence and Security, December 13-18, Suzhou, China*.

Wu, T.-F., C.-J. Lin, and R. C. Weng (2004). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research 5*, 975–1005.

Xiu, D. and G. E. Karniadakis (2002, January). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing 24*(2), 619–644.

Yan, L. and Y.-X. Zhang (2017). Convergence analysis of surrogate-based methods for Bayesian inverse problems. *Inverse Problems 33*, 125001.

Zhang, Y., S. Ghosh, and I. Asher (2019). Learning non-stationary response using clustering and local Gaussian process. In *AIAA SciTech Forum, San Diego, California, USA, January 7th-11th, 2019*, pp. 1–12.