Model selection and adaptive sampling in surrogate modeling : Kriging and beyond.

Malek Ben Salem

N°d'ordre NNT : 2018LYSEM006

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Ecole des Mines de Saint-Etienne

Ecole Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat :
Mathématiques Appliquées

Soutenue publiquement le 19/03/2018, par :
Malek BEN SALEM

Model selection and adaptive sampling in surrogate modeling: Kriging and beyond

Devant le jury composé de :

Prieur, Clémentine	Prof. Université Grenoble Alpes (UGA)	Présidente
Kuhnt, Sonja	Prof. Dr. Fachhochschule Dortmund	Rapporteuse
Haftka, Raphael	Dist. Prof. University of Florida	Rapporteur
Helbert, Céline	Maître de Conférence École Centrale de Lyon	Examinatrice
Roustant, Olivier	Prof. Mines de Saint-Etienne	Directeur de thèse
Gamboa, Fabrice	Prof. Institut de Mathématiques de Toulouse	Co-directeur de thèse
Tomaso, Lionel	Ingénieur de recherche Ansys, inc	Co-Encadrant

À mes parents,

Remerciement

Tout d'abord, j'exprime ma profonde gratitude à mes directeurs de thèse Olivier Roustant et Fabrice Gamboa pour m'avoir donné la possibilité de réaliser cette thèse. Leurs expériences, leur complémentarité, leurs précieux conseils et le recul qu'ils apportaient m'ont été de grande utilité. En plus de leurs qualités académiques indéniables, je les remercie pour leurs qualités humaines, leur disponibilité, leur bienveillance et pour les discussions intéressantes qu'on a pu avoir au Tracteur et bigoudi. En parlant de bienveillance et de qualité humaine et scientifique (et de Tracteur et bigoudi), je suis très reconnaissant envers une autre personne pour son rôle dans l'encadrement de ce travail : il s'agit bien de Lionel Tomaso. Je le remercie chaleureusement pour sa bienveillance, sa patience, sa maîtrise technique, pour la sérénité et le sens l'organisation qu'il a apporté, et pour le fait qu'il peut se débrouiller en allemand ("*Zwei tickets, bitte!*"). Je remercie également François Bachoc pour son efficacité lors des discussions scientifiques intéressantes qu'on a pu avoir.

I would like to thank Sonja Kuhnt and Raphael Haftka for having accepted to review this manuscript, for the expertise they brought in the evaluation of this work and for their valuable comments. Je remercie également Clémentine Prieur et Céline Helbert pour avoir accepté de participer à ce jury et pour la pertinence de leurs remarques et suggestions.

Je suis également reconnaissant envers ceux qui m'ont donné l'opportunité de réaliser ce travail. Je remercie chaleureusement Stéphane Marguerin qui était le premier à me proposer de faire une thèse malgré mon profil inhabituel. Je le remercie pour la persévérance qu'il a démontré pendant une période très délicate dans l'initiation ce travail. Je remercie également Mohamed Masmoudi pour son rôle dans l'initiation de ce projet. J'exprime ma gratitude également à Michel Rochette, Gilles Lebrogne et Phillippe Mahey et tous ceux qui ont contribué pour que ce travail ait lieu.

Je remercie ceux que j'ai eu le plaisir de rencontrer durant ces trois années, notamment les Lyonnais : Jérôme, Mathilde, Laurent, Anthony, Manu, Jean, Christine et les Stéphanois : Christine, Nicolas, Rodolphe, Mireille, Esperan, Hassan, Jean-Charles et Andres.

Je remercie infiniment mes parents pour avoir misé sur mon éducation et pour leur soutien inconditionnel et ininterrompu tout au long de cette thèse et bien avant. Je ne serais assurément pas là sans eux. Je suis également reconnaissant envers mon épouse Rihem pour sa patience, sa présence à mes côtés, ses encouragements et pour son obligeance pendant des situations, plutôt délicates. Et je tiens à remercier ma sœur Nesrine et mes frères Wassel et Nafaa pour leur soutien indéfectible.

J'ai une pensée pour tous mes amis ceux j'ai connu pendant la thèse et ceux de longue date. Je remercie en particulier Dali, Youssef, Adel, Aurélie, Safae et Yosra.

Merci à Tous !

Contents

List of Figures

List of Tables

Part I

Introduction and context

Chapter 1

Introduction

This chapter presents the general context of this work and an extended outline of the manuscript.

1.1 Context and motivations

1.1.1 Framework

Physics-based simulations are generally less expensive and relatively faster than prototyping and testing processes. Moreover, it is sometimes impractical to perform real world experiments (e.g. climate science, earthquakes, airfoil design). Therefore, computer simulations are very popular in applied research and industry. Competition and high standards of specifications fuels the need for more efficient, more robust and ultimately more optimized designs. Therefore, computer simulations are not only used to validate a model. But, they are also used to explore the design space looking for new designs with optimal performances. Both exploration and optimization require in general many evaluations of the simulator. However, high fidelity simulations of complex models remain computationally expensive despite the evolution of high performance computing.

To overcome such cost, surrogate models, also called meta-models or response surfaces, are used to speed-up the exploration of the design space. These functions aim at emulating the true function, here the computationally-intensive simulator, while being computationally cheaper. Surrogate models are commonly used in engineering design [Kle08, SWN13] and there are many construction methods of such approximations

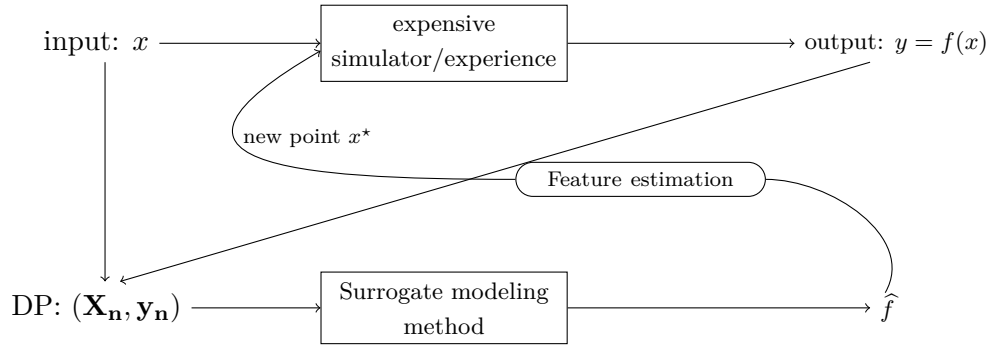


Figure 1.1: Illustration of surrogate-modeling

[Mat69, LS81, SS04, PG89].

Surrogate models are based on a given training set of n observations $Z_n = (z_1, \ldots, z_n)$, where $z_j = (x_j, y_j)$ for $1 \leq j \leq n$ and $y_j = f(x_j)$, called also design points. The main purpose of surrogate modeling is to replace the expensive-to-evaluate function f by a simple response surface $\widehat{f}_{Z_n}$ and then to speed-up the estimation of a feature of f using $\widehat{f}_{Z_n}$. The accuracy of the surrogate model relies, among others, on the relevance of the training set. Of course one is looking for the best trade-off between a good accuracy of the feature estimation and the number of calls of f. Consequently, the design of experiments (DOE), that is the sampling of $(x_j)_{1 \leq j \leq n}$, is a crucial step and an active research field. In Figure 1.1, a schema illustrating surrogate modeling is displayed.

1.1.2 The usages of surrogate modeling

Prediction Let us consider a design space Ω. Generally speaking, the main goal is to predict accurately f on Ω. The accuracy of a surrogate model $\widehat{f_{Z_n}}$ can be measured by a *loss function* that measures the errors between predictions and true values. A typical choice is the square error $\ell_2(x, y) = (x - y)^2$. The integral form of the mean square errors (MSE) is the ℓ_2-risk overall the parametric space.

$$\mathcal{R}_{\ell_2}(\widehat{f_{Z_n}}) = \int_\Omega \ell_2\big(\widehat{f_{Z_n}}(\mathbf{x}), f(\mathbf{x})\big) d\mathbf{x} \tag{1.1}$$

$$= \int_\Omega \big(\widehat{f_{Z_n}}(\mathbf{x}) - f(\mathbf{x})\big)^2 d\mathbf{x} \tag{1.2}$$

The first challenge of the use of surrogate models for design space exploration is to minimize the risk with the minimum number of design points. Thus, one of the research interests is how to sample Z_n to have the smallest risk on Ω. This objective is a partial step in a larger study or the main deliverable. For instance, some health-care applications aim at delivering a fast predictor of physiological properties following an intervention [PH16].

Feature estimation Surrogate models are also used to speed up the engineering process looking for a particular *feature* of the unknown function f. For instance, surrogate model-based techniques have been used for optimization [JSW98, ABDJ+00, FJ08]. In this context, we look for a good approximation of a global minimum of f using a limited number of evaluations. That is, we aim at finding $x^\star \in \Omega$ such that:

$$x^\star \in \arg\min_{x \in \Omega} f(x) \tag{1.3}$$

In other design problems, the goal can be the estimation of a level set of f [RBM08, BES+08]. That is, a threshold of a given value that can be used to estimate for instance a probability of failure. In Figure 1.2, an illustration of a surrogate-based technique for the estimation of a threshold is displayed. Notice that most of the added points by this technique have values around the threshold $T = 150$.

Surrogate models have also been used for other features estimation such as a Pareto front for multi-objective optimization [EDK11, SQMC10, BGR15a], a reliable optimum [NM81, DSB11], a robust optimum [ONL06, TSGB16].

1.2 Outline of the dissertation

The remainder of the manuscript is presented in 5 chapters. In Chapter 2, we present an extended review and the necessary background to set our contributions. Chapter 3, 4 and 5 present three different contributions that can be read separately. Each chapter corresponds to a journal article either published, in revision or submitted. Concluding remarks and perspectives are given in Chapter 6.

A summary of Chapters 2, 3, 4 and 5 is given below.

- Chapter 2 gives the necessary background to set our contributions. We briefly present several classical surrogate modeling techniques and model accuracy assess-

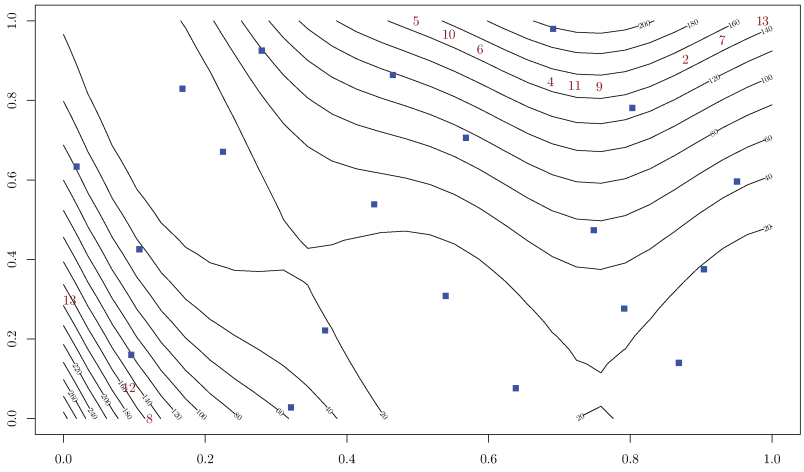


Figure 1.2: Example of sequence of points generated by a surrogate-based algorithm for level set estimation. Initial design points are in blue, the red numbers are the added points in the order of their generation.

ment techniques. A special attention is dedicated to Gaussian Process (GP) regression. We discuss kernels parameterization and how prediction uncertainty allows to perform relevant sequential design. This leads to the final part of the chapter dedicated to design techniques. Feature-oriented sequential design, such as optimization schemes, are also discussed.

- In chapter 3, we discuss model selection techniques. There are many surrogate model types and for each type there are various settings. It is difficult to select the most appropriate surrogate models for a given set of design of experiments. Moreover, it is difficult to assess the quality of a surrogate model and there is no optimal surrogate model for all the problems.

  A selection criterion that assesses the quality of surrogate models is introduced here. We call it the penalized predictive score (PPS). PPS can be computed for any surrogate model. By construction, PPS is especially suitable for response functions that have some regularity and smoothness characteristics. Generally, these

characteristics are implicitly expected within the meta-modeling framework. We show that PPS enables the construction of relevant aggregations of surrogate models called also ensembles. PPS-optimal ensembles are easily computed and avoid over-fitting. We present also two surrogate model selection schemes based on the PPS. The first one computes the PPS-optimal ensemble rather than selecting one surrogate model. The second one is based on a evolutionary framework that enables the exploration of the space of surrogate models.

- Chapter 4 gives a new tool to associate a prediction distribution to any surrogate model and as a result, to extend GP-based sequential design methods to any surrogate model. Recall that the main advantage of GP-based approach is that it provides everywhere a measure of uncertainty associated with the surrogate model prediction. This uncertainty is an efficient tool to construct strategies for various problems such as prediction enhancement, optimization or inversion.

  In this chapter, we propose a universal method to define a measure of uncertainty suitable for any surrogate model. It relies on Cross-Validation (CV) sub-models predictions and leads to a local empirical measure quantifying locally the uncertainty of the surrogate model. This empirical distribution may be computed in much more general frames than the Gaussian one. So that, it is called the Universal Prediction distribution (*UP distribution*). It allows the definition of many sampling criteria. We give and study adaptive sampling techniques to improve prediction accuracy and an extension of the so-called Efficient Global Optimization (EGO) algorithm. We also discuss the use of the *UP distribution* for inversion problems. The performances of these new algorithms are investigated both on toys models and on an engineering design problem.

- Nowadays, many design problems are complex and may involve a high number of variables. Performing design exploration in high dimension is a difficult task. There are several real-life problems where some variables are almost not influential. Chapter 5 presents an algorithm for joint feature estimation learning and dimension reduction. The method is based on Gaussian Process regression. Our method is called the *split-and-doubt* algorithm. The "split" step (model reduction) is based on a property of stationary Automatic Relevance Determination kernels of Gaussian

process regression. We prove that large correlation lengths correspond to inactive variables. We also show that classical estimators such maximum likelihood and cross-validation assign large correlation lengths to inactive variables.

The "doubt" step question the "split" step and helps correcting an initial erroneous estimation of the correlation lengths. It is possible to use this strategy for different feature learning purposes such as refinement, optimization or inversion. The optimization *Split-and-Doubt* algorithm has been evaluated on classical benchmark functions embedded in larger dimensional spaces by adding useless input variables. The results show that *Split-and-Doubt* is faster than classical EGO in the whole design space and outperforms it for most of the considered test case.

Chapters 3, 4 and 5 reproduce the following papers:

- M. Ben Salem and L. Tomaso. Automatic selection for general surrogate models. *Structural and Multidisciplinary Optimization*, Feb 2018 (Chapter 3).
- M. Ben Salem, O. Roustant, F. Gamboa, and L. Tomaso. Universal prediction distribution for surrogate models. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):1086–1109, 2017 (Chapter 4).
- M. Ben Salem, F. Bachoc, O. Roustant, F. Gamboa, and L. Tomaso. Sequential dimension reduction for learning features of expensive black-box functions. *Preprint available at hal-01688329*, 2017 (Chapter 5).

Chapter 2

Background and literature review

2.1 Surrogate-modeling

The term surrogate model encompasses different techniques that have been developed in various fields: regression analysis, response surface methodology, statistical learning, statistical inference, geostatistics. This broad definition includes, inter alia, the least squares method introduced by Legendre [Leg05] and Gauss [Gau09], the so-called response surface methodology, introduced by [BW92, BD87], polynomial chaos expansion [GS03] and artificial neural networks [PG89]. In order to highlight the model selection issue, we present in this section a collection of surrogates modeling techniques and we discuss how to assess their quality.

2.1.1 Notations

To begin with, let f denote a real-valued function defined on Ω, a nonempty subset of the Euclidean space $\mathbb{R}^d$, ($d \in \mathbb{N}^\star$). In order to estimate f, we have at hand a sample of size n ($n \geq 2$): $\mathbf{X_n} = \left(\mathbf{x_1}, \quad \ldots, \quad \mathbf{x_n}\right)^\top$ with $\mathbf{x_j} \in \Omega$, $j \in [\![1; n]\!]$ and $\mathbf{y_n} = \left(y_1, \quad \ldots, \quad y_n\right)^\top$ where $y_j = f(\mathbf{x_j})$ for $j \in [\![1; n]\!]$. We note $\mathbf{y_n} = f(\mathbf{X_n})$. Let $\mathbf{Z_n}$ denote the observations: $\mathbf{Z_n} := \{(\mathbf{x_j}, y_j), j \in [\![1; n]\!]\}$. Using $\mathbf{Z_n}$, we build a surrogate model $\widehat{f_{\mathbf{Z_n}}}$ to approximate f.

Statistical modeling of computer experiments embraces the set of methodologies for generating a surrogate model [VGH10]. Here, in order to avoid the possible confusion between a surrogate model and its construction technique, we introduce the so-called surrogate model builder to denote a method that generates surrogate models based on a

given set of data.

Definition 1. *m is a metamodel builder if: $\forall n \in \mathbb{N}^\star$, $\widehat{m}$ is an application from $\Omega^n \times \mathbb{R}^n$ to $\Omega^{\mathbb{R}}$.*

$$m : \left| \begin{array}{ccc} \Omega^n \times \mathbb{R}^n & \longrightarrow & \Omega^{\mathbb{R}} \\ \mathbf{Z_n} = (\mathbf{X_n}, \mathbf{y_n}) & \longmapsto & m(\mathbf{Z_n}) = \widehat{m}_{|\mathbf{Z_n}} \end{array} \right.$$

where the surrogate model $\widehat{m}_{\mathbf{Z_n}}$ is a an application from Ω to $\mathbb{R}$.

Example 1. *Let $\mathcal{P}^{(1)} : \mathbf{Z_n} = (\mathbf{X_n}, \mathbf{y_n}) \longmapsto \mathcal{P}^{(1)}_{|\mathbf{Z_n}} \in \Omega^{\mathbb{R}}$ denotes a linear polynomial regression such that:*

$$\forall \mathbf{x} = (x_1, \ldots, x_p) \in \Omega,\ \mathcal{P}^{(1)}_{|\mathbf{Z_n}}(\mathbf{x}) = \beta_0 + \sum_1^d \beta_i x_i,$$

where the vector $\widehat{\beta} = (\widehat{\beta}_0, \ldots, \widehat{\beta}_d)^\top$ is the least squares estimate (using $\widetilde{\mathbf{X_n}} = \left(\mathbf{1_n}, \mathbf{X_n}\right)$, here $\mathbf{1_n}$ denotes a vector of size n whose all components equal to 1, assuming further that $n \geq d$ and $\widetilde{\mathbf{X_n}}^\top \widetilde{\mathbf{X_n}}$ is invertible.)

$$\widehat{\beta} = \left(\widetilde{\mathbf{X_n}}^\top \widetilde{\mathbf{X_n}}\right)^{-1} \widetilde{\mathbf{X_n}}^\top \mathbf{y_n}.$$

2.1.2 Overview of some surrogate modeling techniques

There is a wide range of surrogate model building techniques. We give here a brief overview of linear regression models, support vector regression, neural networks and ensembles of surrogates. Gaussian Process regression is presented with more details in Section II.2.

Linear regression: Statistical regression aims at representing the relationships between the set of variables and a set of observed function outputs. It can be traced back to the least squares method [Leg05, Gau09]. In the polynomial regression context, f is assumed to be a polynomial function. The observations are noisy and assumed to be drawn "around" a trend f. More precisely, we have:

$$y_i = \epsilon_i + \sum_{i=1}^p \beta_i f_i(x_i),$$

where $\epsilon_1, \ldots, \epsilon_n$ with i.i.d $\mathcal{N}(0, \delta\sigma^2)$. The estimated response $\widehat{f}$ at any point $x \in \Omega$ is obtained by with the estimators $\widehat{\beta}$ of the coefficients b_i

$$\widehat{f}(x) = \sum_{i=1}^{p} \widehat{\beta}_i f_i(x).$$

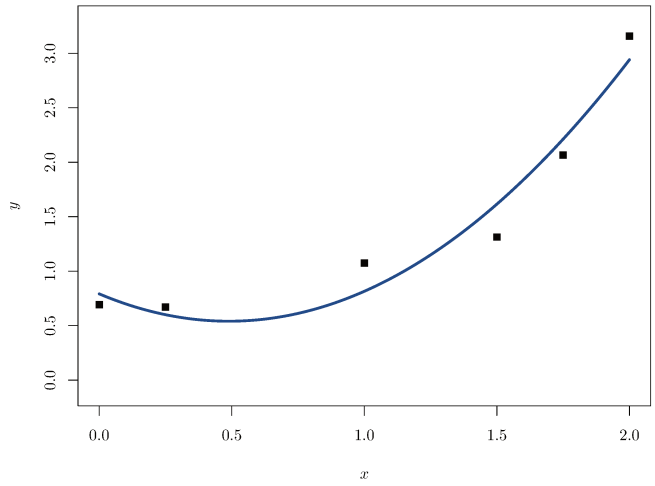


Figure 2.1: Illustration of second order polynomial regression. Solid line: $\widehat{f}$, black points: design points.

There exist many techniques to estimate β such as the ordinary least-squares (OLS). In OLS, the coefficients can be estimated to minimizing the squared errors of the design points. Assuming that the size of the data points n is greater than the size of the basis functions and that $F^\top F$ is invertible, the OLS estimate $\widehat{\beta}$ is given by [BHH78,MMAC16].

$$\widehat{\beta} = (F^\top F)^{-1} F^\top \mathbf{y_n},$$

where

$$F = \big(F_{ij} = f_j(x_i)\big).$$

There are several variants and enhancements for linear regression. For instance, it is common to perform transformation on the input and/or output values. Some popular ones include the Log-transformation, Box-Cox transformation [BC64], Tukey-Ladder

transformation [Tuk77]. Therefore, there is a wide range of possible combinations of different settings: the basis functions, the input transformation, the output transformation, the parameters of the transformations and the estimation method.

Support vector regression: Support vector regression is a particular use of support vector machine (SVM). In [Vap13, Chapter 5], the so-called ϵ-SVR regression aims at finding $f(x)$ that has ϵ as an upper bound for the errors on design points while being as smooth as possible. The approximations function is estimated as follows:

$$\widehat{f}(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^\star) K(x_i, x) + b,$$

where $k(.,.)$ is a kernel function. The parameters α_i, $\alpha_i^\star$ are estimated to minimize the following dual problem:

$$\begin{aligned} \min_{\alpha,\alpha^\star} \quad & -\tfrac{1}{2}(\alpha - \alpha^\star)^\top Q(\alpha - \alpha^\star) - \epsilon \mathbf{1}^\top(\alpha + \alpha^\star) + Y(\alpha - \alpha^\star) \\ \text{subject to :} \quad & \mathbf{1}^\top(\alpha - \alpha^\star) = 1, \\ & \alpha, \alpha^\star \in [0, C], \\ \text{where:} \quad & Q_{ij} = K(x_i, x_j). \end{aligned} \tag{2.1}$$

The parameter b is computed such that the Karush-Kuhn-Tucker (KKT) conditions are satisfied.

Note that there are formulations other than the ϵ-SVR. For instance, the so-called μ-SVR [SSWB00] controls the number of support vector rather than the errors. There are several possible settings to train a support vector regression model. We can cite for instance the kernel function, the parameter value ϵ.

Artificial neural network: Artificial neural network (ANN) [PG89, Lip87, MP69] is a learning method inspired by the biological neural networks modeling [MP43]. They have been used for clustering, classification or regression. It is modeled as a collection of connected nodes (neurons). Each connection is weighted and each node represents a

transfer function.

The weights are generally optimized during the learning process by algorithms such as back-propagation algorithm [Wer74]. The aim of the optimization being to minimize a predefined loss function. A schema representing a neural network is displayed in Figure 2.2.

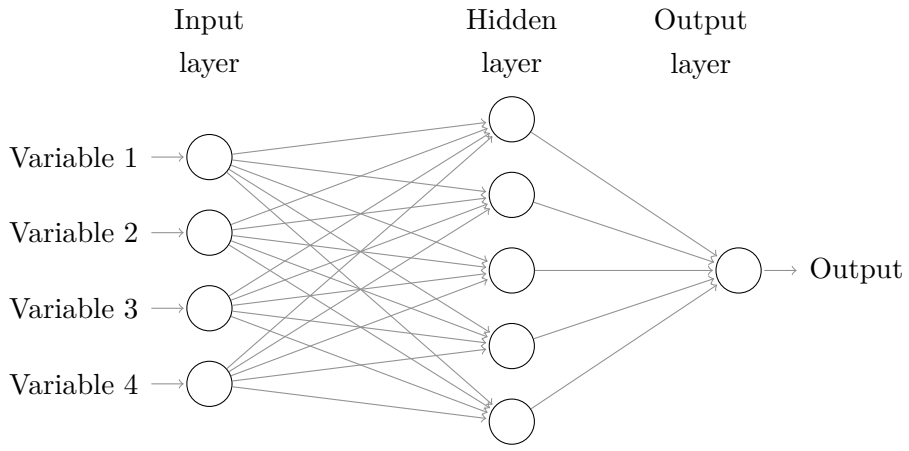


Figure 2.2: Neural network schema

It is important to note that several settings are possible for the artificial neural networks. We can cite as examples, the transfer function, the loss function, the optimization algorithm, the number of layers and the number of cells for each layer.

Aggregation of surrogates: Ensemble of surrogates, also called aggregations or mixtures, have been considered. There are two types of ensembles: local and global. Locally weighted ensemble consider the ensemble prediction as the local weighted prediction of the component surrogates $f_1, \ldots, f_m$ (Equation (2.2)). For instance, in [ZQPS05] the weights are based on the local expected variances.

$$\widehat{f}_{\text{ens}}(x) = \sum_{i=1}^{m} w_i(x)\widehat{f}_i(x). \tag{2.2}$$

However, most weighting methods uses constant weights $w_i(x) = w_i, \forall x \in \Omega$. For instance, Gorissen et al. [GDT09] used a simple average ensemble (all the weights are

equal). Muller et al. [MP11] proposed to weight the aggregation using the Dempster-Shafer theory where the error estimates are used as basic probability assignments. Viana et al. [VHS09] proposed to use an ensemble of surrogate models that minimize the cross-validation errors. Several heuristics to weight ensembles have been proposed in [GHSQ07, GHQS06].

2.1.3 Model accuracy assessment

2.1.3.1 General statistical framework for quality assessment

In statistical inference, the main objective is to estimate a given feature of an unknown distribution based on a given set of data. In regression framework, the feature to estimate is the unknown function $f \in \Omega^{\mathbb{R}}$, and the distribution is $P \sim \Big(X, Y = f(X)\Big)$. So, the performance of a surrogate model is related to its prediction capabilities. The assessment of this performance is extremely important in practice. A quantitative assessment is generally based on a *loss function* that measures the errors between the predictions and the observation of a given set. The risk defined by a loss function l is called *l-risk* (definition 2).

Definition 2. *Let $\Sigma = \Omega \times \mathbb{R}^d$. Let l be a measurable loss function, P be a probability measure on Σ, then for a measurable function $\widehat{m} : \Omega \to \mathbb{R}$*

$\mathcal{R}_{l,P}(\widehat{m}) = \int_\Sigma l(\widehat{m}(x), y) dP(x, y)$ is called the l-risk of $\widehat{m}$.

A popular *loss function* is the quadratic loss function $l_2(\hat{y}, y) = (y - \hat{y})^2$ where $\hat{y}$ is the prediction and y is the observation. The distribution P is generally unknown. In statistics, it is common to use an approximation of P. For instance, one can use the empirical distribution associated to a set of m observations $\mathbf{Z_n}$, $P_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{z_i}$ where δ_{z_i} is the Dirac measure at $z_i = (x_i, y_i)$. The empirical *l-risk* of a function $\hat{m}$ is then:

$$\mathcal{R}_{l,P_n} = \frac{1}{n} \sum_{i=1}^{n} l(\hat{m}(x_i), y_i) \tag{2.3}$$

If we assume that $\mathbf{Z_n}$ are generated (independently) by P and $\hat{m}$ satisfies $\mathcal{R}_{l,P}(\widehat{m}) < \infty$ then $\mathcal{R}_{l,P_n}(\widehat{m}) \to \mathcal{R}_{l,P}(\widehat{m})$ when $n \to \infty$ by the law of large numbers.

Notice that the surrogate model depends on a set of observations generated by P. Therefore, it is not convenient to use the same data to build the surrogate model and to

assess its accuracy. The use of an independent set of observation is more relevant.

2.1.3.2 Resampling techniques

Overfitting is a classical risk of an irrelevant use of statistical inference. Training an algorithm and evaluating its statistical performances on the same data leads to an optimistic result. Resampling techniques allow estimating the risk of a predictor without generating an extra set of observation. For instance, cross-validation [Sto74] or bootstrap [ET93] use a re-sampled sets of the available data $\mathbf{Z_n}$.

Cross-validation: The idea behind CV is to estimate the risk of an algorithm splitting the data once or several times. One part of the data (the training sample) is used for training and the remaining one (the validation sample) is used for estimating the risk of the algorithm. It is generally used to perform model selection or to estimate the accuracy of a meta-model.

Formally, for $i \in 1, \ldots, k$, let $\mathbf{Z}^{(i)}$ be a subset of $\mathbf{Z_n}$ such that $\cup_{i=1}^{k} \mathbf{Z}^{(i)} = \mathbf{Z_n}$. The kF-CV estimates of the l_2 errors (Equation (2.4)) by computing the loss of a point in the i^{th} fold $\mathbf{Z}^{(i)}$ compared to the prediction of the surrogate model built on the remaining folds ($\mathbf{Z}_n \setminus \mathbf{Z}^{(i)}$).

$$\mathcal{R}_{k-CV}(m) = \frac{1}{n} \sum_{i=1}^{k} \sum_{(x',y') \in \mathbf{Z}^{(i)}} l_2(\widehat{m_{|\mathbf{Z}_n \setminus \mathbf{Z}^{(i)}}}(x'), y'), \tag{2.4}$$

where

$$\mathbf{z} \in \mathbf{Z}_n \setminus \mathbf{Z}^{(i)} \text{ if and only if } \mathbf{z} \in \mathbf{Z}_n \text{ and } \mathbf{z} \notin \mathbf{Z}^{(i)}.$$

Several cross-validation procedures are possible. The one described previously is the so-called k-Fold-Cross-Validation (KFCV). Note that when $p = n$, it is called Leave-One-Out Cross-Validation (LOO-CV). Queipo et al [QHS+05] pointed out that the main advantage of CV is that it provides a nearly unbiased estimate. Further, Cross-Validation and Bootstrap performances on a large dataset are studied in studied in [Koh95]. Therein, the authors recommend using stratified 10-fold-cross-validation.

2.1.3.3 Error functions

Besides the limited number of available data. Another problem faces the assessment of surrogate quality. Let us assume that we have at hands a set of validation data of $p \in \mathbb{N}$ observations $Z_p^{(v)} = (z_1^{(v)} = (x_1^{(v)}, y_1^{(v)}), \ldots, z_p^{(v)} = (x_p^{(v)}, y_p^{(v)})))$. The best surrogate model could vary according to the used error measures. For instance, let us consider the root relative mean square error (equation (2.5)) the mean absolute error (equation (2.6)), the relative mean square error (equation (2.7)) and the maximum absolute error (equation (2.8)).

$$\text{rmse}(m(\mathbf{Z_n})) = \sqrt{\frac{1}{p}\sum_{i=0}^{p}(\widehat{m_{\mathbf{Z_n}}}(x_i^{(v)}) - y_i)^2} \tag{2.5}$$

$$\text{mae}(m(\mathbf{Z_n})) = \frac{1}{p}\sum_{i=0}^{p}|\widehat{m_{\mathbf{Z_n}}}(x_i^{(v)}) - y_i| \tag{2.6}$$

$$\text{rmae}(m(\mathbf{Z_n})) = \frac{1}{p}\sum_{i=0}^{p}\frac{|\widehat{m_{\mathbf{Z_n}}}(x_i^{(v)}) - y_i|}{y_i} \tag{2.7}$$

$$\text{Mae}(m(\mathbf{Z_n})) = \max\{|\widehat{m_{\mathbf{Z_n}}}(x_i^{(v)}) - y_i|, i \in 1, \ldots p\} \tag{2.8}$$

Example 2 (Toy example)**.** *We have at hands a set of* $p = 4$ *observations and* 4 *surrogate model builders* m_1, m_2, m_3 *and* m_4*. Table 2.1 shows the true values of the observation* y_i *and the predicted values by the surrogate models. The second part of the table 2.1 shows the error measures values for each surrogate model. The best value for each criterion is written in bold.*

The results show that for each error measure, we have a different optimal surrogate model. This means that the user should be careful when dealing with such measures. In fact, sometime the selection of the measure depends on the field. For instance, if we want to minimize the effect of outliers we would use some conservative measures for instance Huber loss function [Hub64].

2.1.4 Discussion

We presented a non-exhaustive list of surrogate modeling techniques. Generally, there are several settings for each type. However, no method is universally optimal and we

Table 2.1: Toy example of accuracy estimates of surrogate models

i	y_i	$m_1(x_i)$	$m_2(x_i)$	$m_3(x_i)$	(x_i)
1	0.1	0.4	0.5	-0.5	0.2
2	2.4	3.0	2.8	2.1	1.2
3	5.5	5.9	5.9	5.2	6.1
4	12	12	12.4	11.7	11.1
mae		**0.35**	0.4	0.375	0.7
Mae		0.6	**0.4**	0.7	1.2
rmse		0.41	0.4	**0.39**	0.8
rmae		0.83	1.06	1.55	**0.42**

showed that it is rather difficult to assess the quality of a surrogate model. We will tackle the problem of model selection in Chapter 3. This relies on a relevant definition of some assessment criterion and on the selection of the best surrogates or aggregation accordingly.

2.2 Focus on Gaussian Process Regression

2.2.1 Gaussian Process

Definition 3 (Gaussian distribution)**.** *A random variable follows a normal distribution with mean μ and variance σ^2 if its probability density function is:*

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \text{ for } x \in \mathbb{R}.$$

Observe that if $X \sim \mathcal{N}(\mu, \sigma^2)$ then $X = \mu + \sigma N$ with $N \sim \mathcal{N}(0, 1)$.

Multivariate random variables or random vectors are a generalization of random variables.

Definition 4 (Gaussian vector)**.** *A random vector $Y = (Y_1, \ldots, Y_d)$ is a said to be multivariate Gaussian if and only if:*

$$Y = A\varepsilon + \mu,$$

where μ is a $1 \times d$ vector, A a $d \times k$ matrix and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_d)$ is a Gaussian white noise i.e $\varepsilon_1, \ldots, \varepsilon_d)$ i.i.d $\mathcal{N}(0, 1)$.

Indeed, let $Y = (Y_1, \ldots, Y_d)$ be a multivariate Gaussian vector, μ its expected value (vector) $\mu = \mathbb{E}[Y]$ and $\Sigma = AA^\top$ the so-called covariance matrix of Y. The probability density function of Y is given in Equation (2.9):

$$\phi_Y(x) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right). \tag{2.9}$$

Conditional expectation of a random vector: Let $Y = (Y_1, Y_2)$ be the random vector where Y_1, Y_2 are random vectors such as:

$$Y \sim \mathcal{N}\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix} \quad , \quad \begin{pmatrix}\Sigma_{1,1} & \Sigma_{1,2}\\ \Sigma_{2,1} & \Sigma_{2,2}\end{pmatrix}\right).$$

Let us assume that $\Sigma_{2,2}$ is invertible. The conditional distribution of Y_1 knowing Y_2 is a Gaussian vector where:

$$\mathbb{E}[Y_1|Y_2] = \mu_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(Y_2 - \mu_2), \tag{2.10}$$

$$\mathrm{Cov}(Y_1|Y_2) = \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1}. \tag{2.11}$$

Gaussian Process: Stochastic processes or random processes can be seen as a generalization of multivariate random variables. There are several different types of random processes.

Definition 5 (Gaussian process)**.** *A random process Y over $\Omega \subset \mathbb{R}^d$ is Gaussian if*

$$\forall n \in \mathbb{N},\ \textit{for } i = 1, \ldots, n, \quad x_i \in \Omega, \quad \Big(Y(x_1), \ldots, Y(x_n)\Big) \textit{ is a Gaussian vector.}$$

We define the mean function $\mu(x) = \mathbb{E}[Y_x]$ and the covariance function $k(x, y) = \mathbb{E}\Big[\big(Y_x - \mu(x)\big)\big(Y_y - \mu(y)\big)\Big]$. This implicitly requires the process to be integrable at order 2. A Gaussian process Y_x is determined by its mean function and its covariance function [AT09, Chapter2]. Hence, we use the notation $Y \sim GP(\mu(.), k(.,.))$.

Definition 6 (Weak-sense stationarity)**.** *A random process Y is said to be second order stationary or weak-sense stationary if its mean and its covariance are invariant by translation. That is:*

$$\forall (x, y) \in \Omega^2,\ \mu(x) = \mu(y)\ \textit{and}\ \mathrm{Cov}(Y(x), Y(y)) = \kappa(x - y),$$

where κ is a function $\mathbb{R}^d \to \mathbb{R}$.

Since a Gaussian process is fully determined by its mean and its covariance function then second order stationarity is equivalent to the strong-sens stationarity (i.e the distribution of the process is invariant by any translation) for the Gaussian Process.

Covariance function The covariance matrix of a Gaussian vector is positive semi-definite. This notion is extended to covariance functions. A symmetric function $k(.,.)$ over $\Omega \times \Omega$ is positive semi-definite if it satisfies

$$\forall n \in \mathbb{N}, \forall (x_1, \ldots, x_n) \in \Omega^n, \forall \alpha \in \mathbb{R}^n,\ \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0.$$

Any covariance function is positive semi-definite. Conversely, any symmetric positive semi-definite functions is a covariance function of some random process. A review of covariance functions is given in [Abr97].

2.2.2 Kriging or Gaussian Process regression

2.2.2.1 Posterior distribution

Kriging or Gaussian process regression (GPR) is widely popular especially in spatial statistics. It is based on the early works of Krige [Kri51]. The mathematical framework can be found in [Mat63, Ste12, RW06]. Kriging models predict the outputs of a function f: $\Omega = [0,1]^d \to \mathbb{R}$, based on a set of n observations. Within the GP framework, the posterior distribution is given by the conditional distribution of Y given the observations $\mathbf{y_n} = (y_1, \ldots, y_n)^\top$ where $y_i = f(\mathbf{x}^{(i)})$ for $1 \leq i \leq n$. An illustration of a collection of random paths of a GP and their conditional counterpart is displayed in Figure 2.3.

The GPR framework uses a centered real-valued Gaussian Process (GP) Y over Ω as a prior distribution for f. We denote by $k_\theta : \Omega \times \Omega \to \mathbb{R}$ the covariance function (or kernel) of Y: $k_\theta(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[Y(\mathbf{x}), Y(\mathbf{x}')]$ $((\mathbf{x}, \mathbf{x}') \in \Omega^2)$, by ${\mathbf{X_n}}^\top = \left(\mathbf{x}^{(1)}, \ \ldots, \ \mathbf{x}^{(n)}\right) \in \Omega^n$

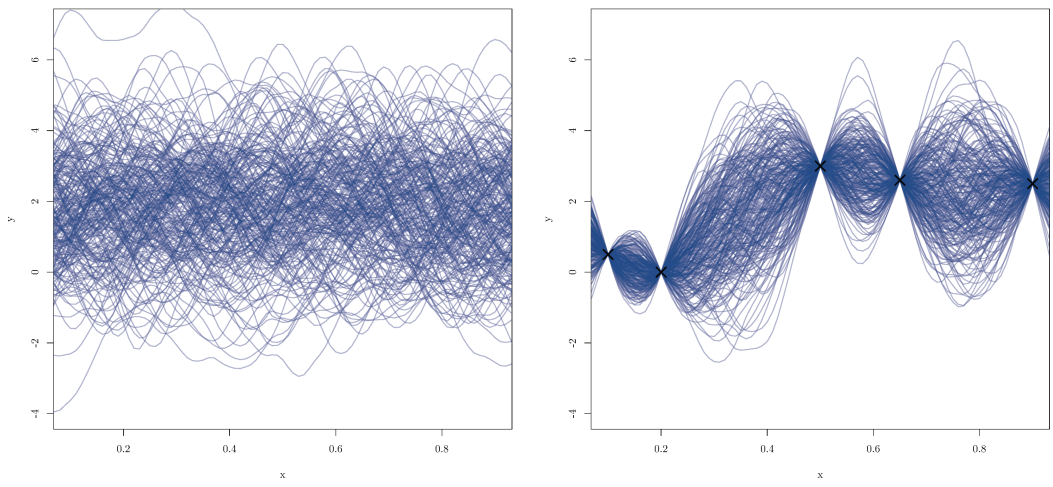


Figure 2.3: Left: Random paths from a Gaussian Process, Right: Conditional Gaussian random paths, black crosses: observations.

the matrix of observation locations and by $\mathbf{Z_n} = \begin{pmatrix} \mathbf{X_n} & \mathbf{y_n} \end{pmatrix}$ the matrix of observation locations and values where $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$ for $1 \leq i \leq n$. The general form assumes that the true model response is a realization of a Gaussian process described by the following Equation [SWN13]:

$$m_Z(\mathbf{x}) = \underbrace{\mu(\mathbf{x})}_{\text{trend function}} + \underbrace{\sigma^2 Y(\mathbf{x})}_{\text{Zero mean Gaussian Process Regression}} \tag{2.12}$$

In the literature, we can find three different types of kriging according to the trend function:

- Simple Kriging: when $\mu(\mathbf{x}) = \mu$ where μ is a known constant.
- Ordinary Kriging: when $\mu(\mathbf{x}) = \mu$ where μ is an unknown constant.
- Universal Kriging: when $\mu(\mathbf{x}) = \beta^\top \mathbf{h}(\mathbf{x})$ where (h_i) are known trend functions and β is unknown.

Simple Kriging: Without loss of generality, we consider the simple kriging framework. The *a posteriori* conditional mean $\widehat{m}_{\mathbf{Z_n}}$ and the *a posteriori* conditional variance $\widehat{\sigma}^2_{\mathbf{Z_n}}$ are

given by:

$$\widehat{m}_{\mathbf{Z_n}}(\mathbf{x}) = k(\mathbf{x}, \mathbf{X_n})^\top K_\theta^{-1} \mathbf{y_n}, \tag{2.13}$$

$$\widehat{\sigma}^2_{\mathbf{Z_n}}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X_n})^\top K_\theta^{-1} k(\mathbf{x}, \mathbf{X_n}). \tag{2.14}$$

Here, $k(\mathbf{x}, \mathbf{X_n})$ is the vector $(k(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, k(\mathbf{x}, \mathbf{x}^{(n)}))^\top$ and $K_\theta = k(X, X)$ is the invertible matrix with entries $\Big(k(\mathbf{X_n}, \mathbf{X_n})\Big)_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, for $1 \leq i, j \leq n$.

Universal Kriging: Let $(h_i)_{1 \leq i \leq p}$ be the basis functions of the trend function used in the universal kriging. Let us call $\mathbf{h}(\mathbf{x})$ the vector $(h_1(\mathbf{x}), .., h_p(\mathbf{x}))^\top$ and H the matrix defined as follows: $H = \Big(h_{i,j} = h_j(x_i)\Big)_{1 \leq i \leq n, 1 \leq j \leq p}$. The conditional mean and the conditional variance of the Gaussian process are given below [Cre93, Rip05]. .

$$\widehat{m}_{\mathbf{Z_n}}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \widehat{\beta} + k(\mathbf{x}, \mathbf{X_n})^\top K^{-1} (Y - H^\top \widehat{\beta}) \tag{2.15}$$

$$\begin{aligned} \widehat{\sigma}^2_{\mathbf{Z_n}}(\mathbf{x}) = {} & k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X_n})^\top K^{-1} k(\mathbf{x}, \mathbf{X_n}) \\ & + \Big(\mathbf{h}(\mathbf{x})^\top + k(\mathbf{x}, \mathbf{X_n})^\top K^{-1} H\Big)^\top \Big(H^\top K^{-1} H\Big)^{-1} \Big(\mathbf{h}(\mathbf{x})^\top + k(\mathbf{x}, \mathbf{X_n})^\top K^{-1} H\Big) \end{aligned} \tag{2.16}$$

where:

$$\widehat{\beta} = (H^\top K^{-1} H)^{-1} H^\top K^{-1} Y. \tag{2.17}$$

These equations can be derived from 2.10 and 2.11 by considering a Bayesian Framework with specific priors on β and σ [HDC09]. Finally notice that kriging predictions depend on several settings: the trend function, the prior distribution including the kernels parameters, the possible noise and the estimation method.

2.2.3 Link with Reproducing Kernel Hilbert Space

Reproducing kernel Hilbert space: A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ is a Hilbert Space of real-valued functions defined on Ω where evaluation functionals

$$T_x : f \mapsto f(x), \ \forall f \in \mathcal{H}$$

are continuous.

Definition 7 (Reproducing Kernel Hilbert Space)**.** *Let $\mathcal{H}$ be a Hilbert Space of real-valued functions defined on Ω. $\mathcal{H}$ is a RKHS if $\forall x \in \Omega$, T_x is a bounded operator*[1] *on $\mathcal{H}$ i.e there exists some $M > 0$ such that*

$$|T_x(f)| = |f(x)| \leq M||f||_{\mathcal{H}}, \ \forall f \in \mathcal{H}.$$

Theorem 1 (Fréchet-Riesz representation theorem)**.** *Let T be a continuous linear form on $\mathcal{H}$ then,*

$$\textit{If } \mathcal{H} \textit{ is an RKHS} \quad \exists \tau \in \mathcal{H}, \quad T(f) =< \tau, f >_{\mathcal{H}}, \forall f \in \mathcal{H}.$$

Every T_x is linear and continuous on $(\mathcal{H}, < ., . >_{\mathcal{H}})$. Thus, it can be represented by an element of $\mathcal{H}$ using $< ., . >_{\mathcal{H}}$ (Fréchet-Riesz Theorem):

$$\exists k_x \in \mathcal{H}, \ T_x(f) =< k_x, f >_{\mathcal{H}} .$$

Proposition 1. *Let k be the function defined on $\Omega \times \Omega \to \mathbb{R}$ defined by $k(x, y) = k_x(k_y)$. Then, k is positive semi-defined.*

Proof.

- $k(x, y) = k_x(k_y) =< k_x, k_y >_{\mathcal{H}} =< k_y, k_x >_{\mathcal{H}} = k_y(k_x) = k(y, x)$.
- $\forall n \in \mathbb{N}, \forall \alpha_1, \ldots, \alpha_n \in \mathbb{R}, \forall x_1, \ldots, x_n \in \Omega$,

$$\sum_{i=1,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) =< \sum_{i=1}^{n} \alpha_i k_{x_i}, \sum_{j=1}^{n} \alpha_j k_{x_j} >_{\mathcal{H}} = \Big|\Big| \sum_{i=1}^{n} \alpha_i k_{x_i} \Big|\Big|_{\mathcal{H}}^2 \geq 0.$$

□

$k(x, y) =< k_x, k_y >_{\mathcal{H}}$ is called a reproducing kernel. Moore–Aronszajn theorem states that if a function k is a symmetric, positive definite kernel on a Ω then there is a unique Hilbert space of functions on Ω for which k is a reproducing kernel, given by

$$\mathcal{H}_k = \overline{\text{Span}\{k(x, .), x \in \Omega\}}.$$

For more insight, on may refer to [Aro50, BTA11].

[1] This is equivalent to T_x is continuous at any $f \in \mathcal{H}$

Gaussian Process and RKHS

- A covariance function k of a centred Gaussian Process is also the reproducing kernel of the RKHS $\mathcal{H}_k$ [2].
- The simple kriging posterior mean is also the function in $\mathcal{H}$ with minimal norm that interpolates the data [MR85].
- Note that Gaussian Processes random paths are not generally in the corresponding RKHS [Dri73, BTA11].

2.2.4 Kernel functions

A common approach consists in assuming that the covariance function belongs to a parametric family. A review of classical covariance functions is given in [Abr97]. We are interested in the family of auto relevance determination (ARD) kernels of the form of Equation 2.18. The ARD kernels include most popular kernels such as the exponential kernel, the Matérn 5/2 kernel and the squared exponential (SE) kernel given in Table 2.2.

$$k_\theta(\mathbf{x}, \mathbf{y}) = \sigma^2 \prod_{p=1}^{d} k\Big(\frac{d(x_p, y_p)}{\theta_p}\Big), \text{for } \mathbf{x}, \mathbf{y} \in \Omega. \tag{2.18}$$

Here, $d(,)$ is a distance on $\Omega \times \Omega$ and $k : \mathbb{R} \to \mathbb{R}$ is a fixed stationary covariance function. The hyper-parameters σ and $\theta_1, \ldots, \theta_d$ have to be estimated. To do so, we use the Maximum Likelihood (ML) estimator or Cross Validation (CV). Both methods have interesting asymptotic properties [Bac13a, Bac14, BLN17]. Nevertheless, when the number of observations is relatively low, the estimation can be misleading. These methods are also computationally demanding when the number of observations is large.

2.2.4.1 Hyper-parameters estimation: maximum likelihood estimation

Without loss of generality, we consider the Gaussian process regression framework and we assume $\mu = 0$. The centred process depends only on its kernel that depends in turn on its hyper-parameters. The likelihood in this case is given in Equation 2.19:

[2] Henceforth, we use sometimes the term kernel to designate the covariance function of a Gaussian Process

Name	Expression
exponential	$k(x,y) = \sigma^2 \exp\left(-\frac{\lvert x-y\rvert}{\theta}\right)$
squared exponential	$k(x,y) = \sigma^2 \exp\left(-\frac{(x-y)^2}{2\theta^2}\right)$
Matern 5/2	$k(x,y) = \sigma^2 \left(1 + \frac{\sqrt{5}\lvert x-y\rvert}{\theta} + \frac{5\lvert x-y\rvert^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}\lvert x-y\rvert}{\theta}\right)$
Matern 3/2	$k(x,y) = \sigma^2 \left(1 + \frac{\sqrt{3}\lvert x-y\rvert}{\theta}\right) \exp\left(-\frac{\sqrt{3}\lvert x-y\rvert}{\theta}\right)$

Table 2.2: Examples of common kernels.

$$L(\sigma^2, \theta) = \frac{1}{|2\pi k(X,X)|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y_n}^\top K^{-1}\mathbf{y_n}\right), \tag{2.19}$$

where $k(.,.)$ depends on σ^2 and θ. To overcome numerical problems, it is common to consider the log-likelihood :

$$\log(L(\sigma^2, \theta)) = -\frac{n}{2}2\pi - \frac{1}{2}\log(|k(X,X)|) - \frac{1}{2}\mathbf{y_n}^\top\mathbf{y_n}. \tag{2.20}$$

2.2.4.2 Hyper-parameters estimation: Cross Validation

Without loss of generality, we consider the Leave-One-Out Cross-Validation (LOO-CV). LOO-CV consists in dividing the n point into n subsets of one point each. Then, each subset plays the role of test set while the remaining points are used together as the training set. Using Dubrule's formula [Dub83], the LOO-CV estimator is given in (2.21).

$$\widehat{\theta}^\star_{CV} \in \arg\min_\theta \frac{1}{n}\mathbf{y_n}^\top K^{-1}\,\mathrm{diag}(K^{-1})^{-1}K^{-1}\mathbf{y_n} \tag{2.21}$$

For more insight on these estimators, one can refer to [Bac13b].

2.2.5 Discussion

In Chapter 5, we will use some properties of the GPR. In fact, we will exploit some properties of the correlation lengths of ARD kernels to propose the so-called split-and-doubt algorithm. It consists in filtering some input variables while performing sequential design.

2.3 Design of experiments (DOE)

In surrogate modeling framework, the sampling of design points is a crucial step. Generally speaking, there are two ways to sample: either drawing the training points at once (one-shot design) or generating it sequentially (adaptive design). In this section, we give a brief overview of some design techniques.

2.3.1 Non-adaptive designs

The one-shot designs are methods that sample all the experiments independently of the values of the function output(s). These methods are also used in surrogate-based sequential designs to generate the initial DOE. Thus, it is a crucial issue in meta-modeling.

Design of experiments techniques can be roughly divided into three types: deterministic, random and quasi-random. Among the deterministic methods, we can cite the factorial designs, central composite designs, Box–Behnken designs [BB60] and orthogonal arrays [Owe92, Owe94]. Full factorial designs are basically d-dimensional grids of k levels in each dimension. Its main drawback is that the total number of design points $n = k^d$ grows exponentially with the dimension.

Lindely [Lin56] introduced a maximum entropy design technique. It is based on the amount of information provided by an experiment. Another type of deterministic model is when the model is specified. Among these particular designs, we may cite A-optimal design [Che53, Fed72], D-optimal design [Fed72, PW85, WP90] and so-called Bayesian designs. Shewry and Wynn [SW87] showed that if the design space is discrete then minimizing the expected posterior entropy is equivalent to maximize the prior entropy.

Geometric designs [JMY90] aim at optimizing a distance-based design criterion such as *minimax* or *maximin* criteria.

$$\min_{x\in\Omega}\max_{x_i\in X}||x-x_i||,$$

$$\max_{X=(x_1,\ldots,x_n)}\min_{i\neq j}||x_i-x_j||.$$

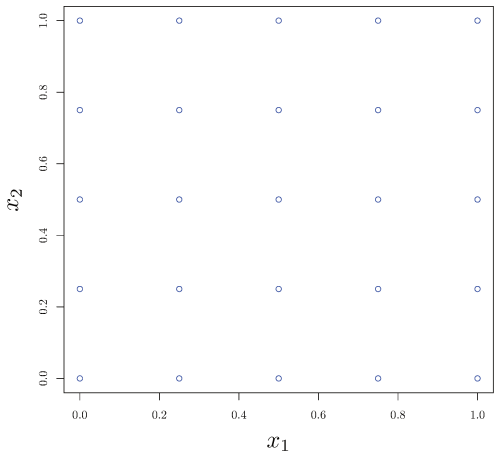
As shown in [PM12], the *minimax* criterion is equivalent to find the smallest balls centered in design points that cover the design space and *maximin* criterion seeks to maximize the radius of non-intersecting balls centered on design points.

Among the random designs, one of the most popular is the Latin Hypercube Sampling (LHS) developed by [MBC79]. It is proposed as an enhancement of Monte-Carlo Sampling. A square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A LHS is the generalization of this concept to an arbitrary number of dimensions. Among the designs displayed in Figure 2.4, three of them are LHS. Let us consider the singular case (Figure 2.4d). This example is of course one of the worst possible cases and its occurrence is not very likely. However, this highlights the limitation of LHS. Thus, there are many improvement to pure LHS: the so-called optimal Latin hypercube sampling (O-LHS[3]). The main idea is to keep the projection property of LHS and use another criterion such as maximin [MM95].

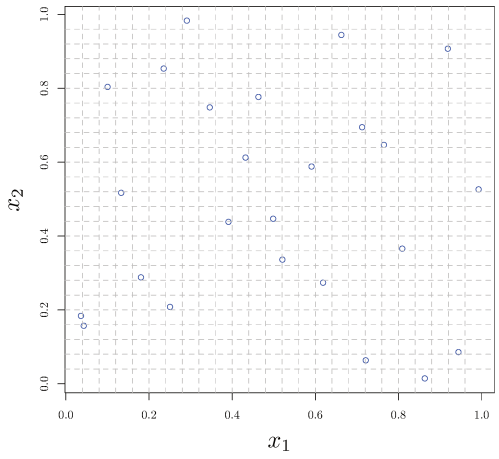
2.3.2 Adaptive design

A design is adaptive if the information from the experiments (inputs and responses) and/or information from the metamodel is used in selecting the next sample. "Adaptive approaches are typically superior to non-adaptive approaches" [ASA+13]. Generally, an adaptive approach begins with an initial design either deterministic, random or quasi-random. A metamodel is constructed using the initial experiments and then new samples are chosen by systematically evaluating the response and/or the metamodel the current design point. Such definition includes feature-oriented designs such as optimization algorithms and inversion algorithms. An example is the stepwise uncertainty reduction (SUR) strategy that has been applied for excursion set identification [CBG+14], constrained optimization [Pic14] and multi-objective optimization [Pic15].

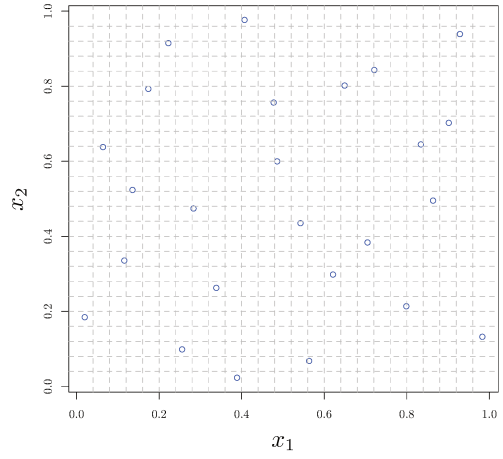
[3]The O-LHS in Figure 2.4 was realized with the package DiceDesign [DHF15]



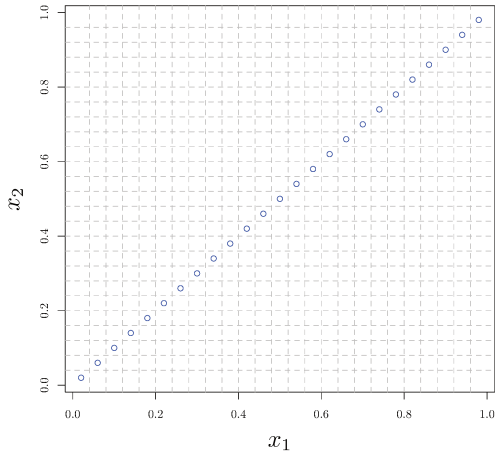
(a) Factorial Design



(b) Random LHS design



(c) Optimized LHS design



(d) Singular LHS design

Figure 2.4: Four different examples of design of experiments.

An efficient tool to construct such adaptive methods is the prediction distribution of a surrogate model. Hence, many methods are based on kriging. In the next section, we present a brief overview of two popular kriging-based strategies: Efficient Global Optimization (EGO) and stepwise uncertainty reduction (SUR). We also discuss other methods.

2.3.2.1 GPR-based adaptive design algorithms

Gaussian Process Regression provides a prediction distribution. Many surrogate-based sequential design methods take advantage of this tool. Sequential-based design for different features have been studied. For example, the inversion is considered in [BES$^+$08] and the context of optimization is studied in [FJ08,Sas98]. We take here a closer look on two strategies: EGO and SUR.

Efficient Global Optimization: Bayesian global optimization (BO) techniques have been successfully used in various problems [Moč75,Moč82,JSW98]. One of the most popular algorithms is the so-called Efficient Global Optimization (EGO) algorithm of [JSW98]. It consists in sampling the point that maximizes the so-called expected improvement (EI).

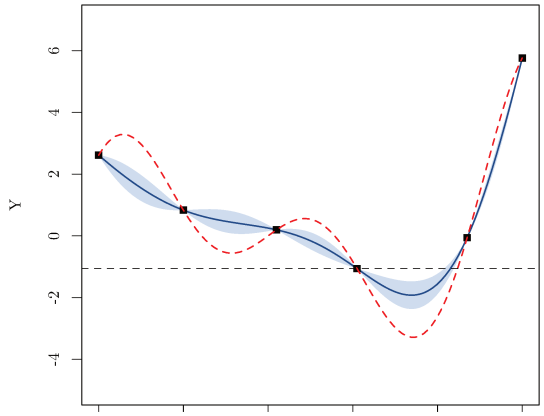
Let $(y(\mathbf{x}))_{\mathbf{x}\in\mathbb{X}}$ be a Gaussian process. Let further m_{GP} and σ^2_{GP} denote respectively the mean and the variance of the conditional process $y(\mathbf{x}) \mid \mathbf{Z}$. Last, let $y^\star$ be the minimum value of the response on the sample $\mathbf{Z} = (z_1, \ldots, z_n)$ where $z_i = (\mathbf{x_i}, y_i)$, that is $y^\star = \min_{i=1..n} y_i$. The EGO algorithm [JSW98] uses the expected improvement EI (Equation (2.22)) as sampling criterion:

$$EI(\mathbf{x}) = \mathbb{E}\Big[\max(y^\star - y(\mathbf{x}), 0) \mid \mathbf{Z}\Big] \tag{2.22}$$

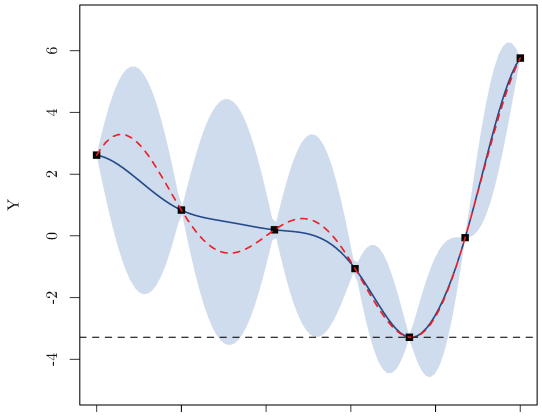
Using some Gaussian computations, $EI(\mathbf{x})$ can be explicitly computed:

$$EI_n(\mathbf{x}) = \begin{cases} (y^\star - m_{GP}(\mathbf{x}))\Phi\left(\dfrac{y_n^\star - m_{G_n}(\mathbf{x})}{\sigma_{GP}(\mathbf{x})}\right) \\ \qquad + \ \sigma_{GP}(\mathbf{x})\phi\left(\dfrac{y_n^\star - m_{G_n}(\mathbf{x})}{\sigma_{GP}(\mathbf{x})}\right) & \text{if } \sigma_{GP}(\mathbf{x}) \neq 0, \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

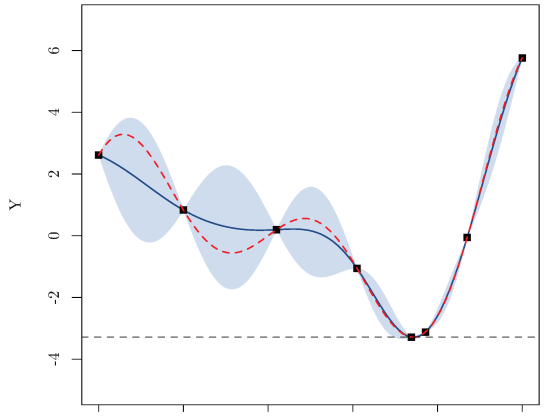
The EGO algorithm adds to the sample the point that maximizes EI. An illustration of 5 iterations of EGO on a toy example is displayed in Figure 2.5.



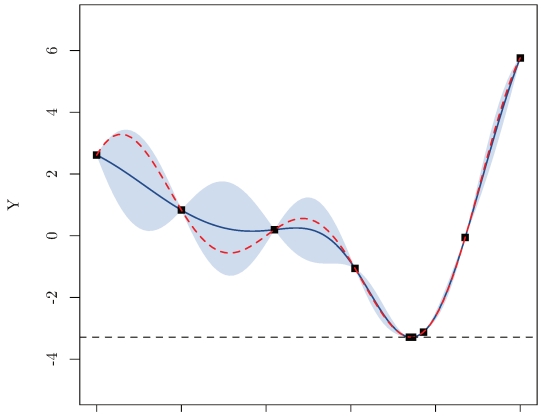
(a) initial



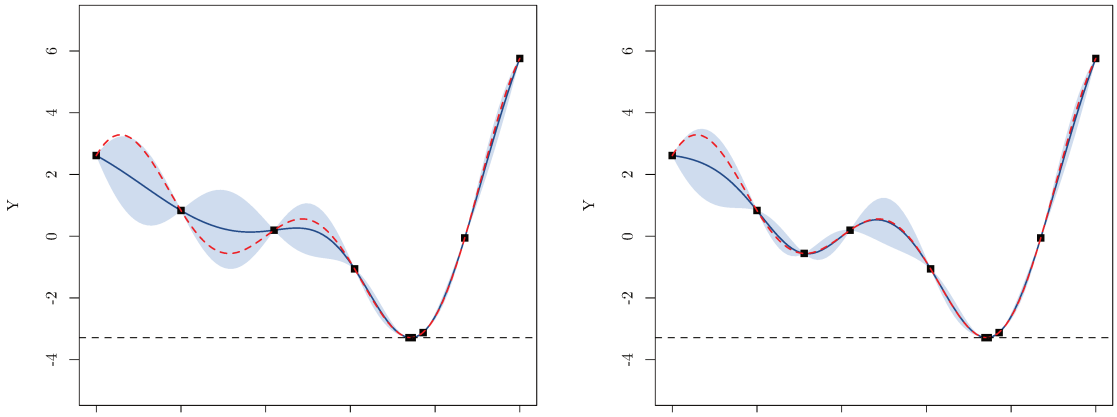
(b) Iteration 1



(c) Iteration 2



(d) Iteration 3



(e) Iteration 4 (f) Iteration 5

Figure 2.5: Example of 1-dimensional EGO algorithm. Dashed red line: Real function, dashed black line: value of the current minimum, Solide line: Kriging prediction, black squares: design points.

Stepwise Uncertainty Reduction (SUR): The expected improvement criterion is said to be myopic. As a matter of fact, it selects each point as if it would be the last one. Indeed, it selects at each iteration the point with the largest expected improvement value. Another surrogate-based sequential design, called the Stepwise Uncertainty Reduction (SUR) strategy has also been widely studied [BGL$^+$12, VB09, VPM07, CBG$^+$14]. It consists in generating a sequence of points that reduces the uncertainty of the posterior distribution of a quantity of interest. The Informational Approach to Global Optimization (IAGO) [VVW09] can be seen as an example of a SUR strategy. Roughly speaking, the SUR strategy requires an uncertainty measure $\mathcal{M}$ with regards to the quantity of interest. The SUR optimization strategy consists in adding the point that minimizes the uncertainty measure $\mathcal{M}$. It is also a straightforward strategy in terms of uncertainty reduction on the quantity of interest. For instance, in an optimization context the uncertainty measure can be the excursion set (above or below) the current optimum.

2.3.2.2 Other surrogate-based adaptive design strategies

Other surrogate-based adaptive designs, that are not based on a prediction uncertainty tool, have been developed. Some of these methods are based on hold-out sample, resam-

pling techniques or ensembles.

For example, [LMA+04] use Multivariate Adaptive Regression Splines and kriging models with Sequential Exploratory Experimental Design method. It consists in building a surrogate model to predict errors based on the errors on a test set. Goel et al. [GHSQ07] use a set of surrogate models to identify regions of high uncertainty by computing the empirical standard deviation of the predictions of the ensemble members.

In the literature, several cross-validation-based techniques have been discussed. Li and Azarm [LA06] propose to add the design point that maximizes the Accumulative Error (AE). The AE on $\mathbf{x} \in \mathbb{X}$ is computed as the sum of the LOO-CV errors on the design points weighted by influence factors. This method could lead to clustered samples. To avoid this effect, the authors [LAFMD06] propose to add a threshold constraint in the maximization problem. Busby, Farmer, and Iske [BFI07] propose a method based on a grid and CV. It affects the CV prediction errors at a design point to its containing cell in the grid. Then, an entropy approach is performed to add a new design point. More recently, Xu et al. [XLWJ14] suggest the use of a method based on Voronoi cells and CV. Kleijnen and Van Beers [KvB04] propose a method based on the Jackknife's pseudo values predictions variance. Jin, Chen, and Sudjianto [JCS02] present a strategy that maximizes the product between the deviation of CV sub-models predictions with respect to the master model prediction and the distance to the design points. Aute et al. [ASA+13] introduce the Space-Filling Cross-Validation Trade-off (SFCVT) approach. It consists in building a new surrogate model over LOO-CV errors and then add a point that maximizes the new surrogate model prediction under some space-filling constraints. In general, cross-validation-based approaches tend to allocate points close to each other resulting in clustering [ASA+13]. This is not desirable for deterministic simulations.

2.3.3 Discussion

In this section, we introduced several design of experiments techniques. The main take-home messages of this section are:

- It is natural to assume that adaptive designs strategies may give better results than non-adaptive designs. Indeed, these methods uses the response to sample the future points.

- Most of the popular surrogate-based design strategies are based on GPR. This is due to the fact that prediction distribution is given analytically. We know yet that several surrogate models techniques are available and useful (Section 2.1).

In Chapter 4, we will introduce the so-called universal prediction distribution that defines a prediction distribution for all surrogates.

Part II

Surrogate modeling

Chapter 3

Surrogate model selection

Abstract In design engineering problems, the use of surrogate models (also called meta-models) instead of expensive simulations have become very popular. Surrogate models include individual models (regression, kriging, neural network...) or a combination of individual models often called aggregation or ensemble. Since different surrogate types with various tunings are available, users often struggle to choose the most suitable one for a given problem. Thus, there is a great interest in automatic selection algorithms. In this paper, we introduce a universal criterion that can be applied to any type of surrogate models. It is composed of three complementary components measuring the quality of general surrogate models: internal accuracy (on design points), predictive performance (cross-validation) and a roughness penalty.

Based on this criterion, we propose two automatic selection algorithms. The first selection scheme finds the optimal ensemble of a set of given surrogate models. The second selection scheme further explores the space of surrogate models by using an evolutionary algorithm where each individual is a surrogate model. Finally, the performances of the algorithms are illustrated on 15 classical test functions and compared to different individual surrogate models. The results show the efficiency of our approach. In particular, we observe that the three components of the proposed criterion act all together to improve accuracy and limit over-fitting.

3.1 Introduction

Computer simulations are an efficient tool to study complex physical behaviors. However, high-fidelity simulations are generally computationally expensive. Therefore, surrogate models, also known as metamodels or response surfaces, are usually instead used. They provide an approximation of a response of interest based on a limited number of expensive simulations. There are several methods of construction of such approximations. Among the popular surrogate model types, we can cite for example Kriging [Mat63], support vector machines (SVM) [SS04], Moving least squares [LS81] and Multivariate Adaptive Regressive Splines (MARS) [Fri91]. Generally, a metamodel family comes with several possible tunings. In the same time, there is no universal optimal surrogate for all the problems. Some users face some difficulties in selecting the most suitable surrogate for their problem. Thus, there is a great interest in automatic model selection algorithms. The main purpose is to choose the surrogate that provides the best prediction performances on the whole parametric space.

In the literature, this problem is generally studied along three different approaches.

1) The first approach consists in using algorithms to optimize the settings of a particular surrogate model type. For instance, [CWL04, LSC06] work on SVM, [ZSL00] on neural networks, and [TNE07] deal with least squares regression.

2) A second approach consists in considering multiple surrogates or ensembles. The automatic surrogate selection is so a model selection method. Often, the selected model is a weighted sum of different surrogate models. For example, [VHS09, ZML11, ARR09, GHSQ07] discuss different ways to build such aggregations.

3) The last approach consists in selecting a *good* member among different types of surrogate models with different settings. We refer for instance to the works of [GDT09, SYZ12, ZJ16].

The main objective of our paper is to propose a new relevant surrogate model selection algorithm that can handle different type of surrogates. To achieve such a goal, we define a universal criterion. This criterion may evaluate the accuracy of any surrogate model.

The paper is organized as follows. We introduce and discuss in Section 3.2 our criterion called the Penalized Predictive Score (PPS). We show in Section 3.3 that PPS is

suitable to optimize weights of surrogate models ensembles. In Section 3.4, we present an evolutionary selection algorithm that explores the space of surrogate models. The algorithm is called PPS Genetic Aggregations (PPS-GA). Finally, the performances of the algorithm on 15 test cases are displayed in Section 3.5. The results show the efficiency of the PPS, the complementary role of its three components and the relevance of the proposed selection algorithms.

3.2 Penalized Predictive Score (PPS)

3.2.1 Definition

Assessing the quality of a surrogate is very challenging. It is desirable to use an independent set to assess the predictive capabilities of a given method. But, this is computationally expensive in practice. One can also estimate the errors by computing the errors on design points. Unfortunately, a small MSE does not imply good predictive capabilities. Therefore, resampling techniques such as Cross-Validation (CV) [Sto74] or bootstrap [ET93] are generally used. Such techniques reduce the bias of the estimation. Nevertheless, they does not prevent overparameterized models. We will introduce a criterion that will do this job. This criterion is called the Penalized Predictive Score (PPS Equation (3.1)). It combines three components:

a) The internal accuracy (or fit): we use the mean squared errors (MSE) on design points.

b) The predictive capability: we propose to use the 10F-CV PRESS errors.

c) A roughness penalty: we propose to use the Bending Energy Functional (BEF) ([Duc77]).

$$PPS(m, \mathbf{Z_n}) = \underbrace{\alpha \widehat{\mathcal{R}}_{l_2,\mathbf{Z_n}}(m)}_{a} + \underbrace{\beta \mathcal{R}_{10-CV}(m)}_{b} + \underbrace{\gamma E_n(m)}_{c} \tag{3.1}$$

Here, as it will be described below, $\widehat{\mathcal{R}}_{l_2,\mathbf{Z_n}}(m)$ denotes the MSE criterion, $\mathcal{R}_{10-CV}(m)$ the 10-Fold cross-validation estimate of the errors and $E_n(m)$ a roughness penalty. Further, α, β, γ are weights in $\mathbb{R}_+$. In all our implementations, we use $\alpha = 2\beta$ and $\beta = 2\gamma$.

3.2.2 Internal accuracy

Let $\Omega = [0,1]^d$ be the parametric space of dimension d. $\mathbf{X_n} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^\top \in \Omega^n$ and $\mathbf{Y_n} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ form the set of design points $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n})$ where $y_i = f(\mathbf{x_i})$ for $i = 1, \ldots, n$ and $f \in \mathbb{R}^\Omega$ is an expensive-to-evaluate function. A surrogate model $\widehat{m}_{|\mathbf{Z_n}} \in \Omega^\mathbb{R}$ is used to replace f based on the design $\mathbf{Z_n}$. We call the construction method a "surrogate model builder". For instance, if m is a surrogate model builder, then we build the surrogate model $\widehat{m}_{|\mathbf{Z_n}} \in \Omega^\mathbb{R}$ based on the design $\mathbf{Z_n}$.

The assessment of the performance of a surrogate model is extremely important in practice [HTF09]. It relies on the evaluation on the set of design points of the prediction capabilities of the surrogate model. It is generally based on a *contrast function* (or *loss function*) that measures the errors between the predicted and the true models. A typical choice is the square error $l_2(x, y) = (x-y)^2$. The integral form of the MSE is the $l_2 - risk$ overall the parametric space.

$$\mathcal{R}_{l_2,\mathbf{Z_n}}(m) = \int_\Omega l_2\big(\widehat{m_{|\mathbf{Z_n}}}(\mathbf{x}), f(\mathbf{x})\big) d\mathbf{x} \tag{3.2}$$

Since f is unknown, we can only use an approximation to estimate this risk. Ideally, the performance of the surrogate model would be evaluated on an extra set of points. However, generating such set is sometimes computationally expensive. Therefore, one use an empirical distribution associated to the set of design points. Computing the mean square errors (MSE) (Equation (3.3)) on the set of design points for the surrogate model $\widehat{m_{|\mathbf{Z_n}}}$ is an empirical approximation of $\mathcal{R}_{l_2,\mathbf{Z_n}}(m)$ defined in Equation (3.2).

$$\begin{aligned} \widehat{\mathcal{R}}_{l_2,\mathbf{Z_n}}(m) &= \frac{1}{n}\sum_{i=1}^{n} l_2(\widehat{m_{|\mathbf{Z_n}}}(x_i), y_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} (\widehat{m_{|\mathbf{Z_n}}}(x_i) - y_i)^2 \end{aligned} \tag{3.3}$$

Note that computing the MSE on the set of design points is a biased estimate of the error in the whole space. In fact, for any interpolating surrogate model m, $\widehat{\mathcal{R}}_{l_2,\mathbf{Z_n}}(m) = 0$. This does not necessarily mean that the surrogate model fits the real function in the whole space.

3.2.3 Predictive capabilities

On one hand, the use of design points to estimate the errors yields an optimistic result [AC10]. On the other hand, using a validation set can be expensive. Therefore, it is convenient to use re-sampling techniques such as Cross-Validation (CV) [Sto74] and bootstrap [ET93] to estimate the predicted errors. Resampling techniques estimate the errors by using subsets of the design points to build several *sub-surrogate models*. For instance, computing the Leave-One-Out Cross-Validation (LOO-CV) errors of a surrogate model $\widehat{m_{|\mathbf{Z_n}}}$ consists in computing the errors of an observation $(\mathbf{x_i}, y_i)$ based on the surrogate model $\widehat{m_{|\mathbf{Z_{n,-i}}}}$ built on the subset of all the design points except the i^{th} design point ($\mathbf{Z_{n,-i}} = (\mathbf{x_j}, \mathbf{y_j})_{j \neq i}$). In the same way, k-fold cross-validation (kF-CV) consists in dividing the data into k subsets. Each subset plays the role of validation set while the remaining $k-1$ subsets are used together as the training set. If k is the number of folds, for $i \in 1, \ldots, k$ let $\mathbf{Z}^{(i)} \in \mathcal{P}(Z_n)$ be a subset of Z_n such that $\cup_{i=1}^{k} \mathbf{Z}^{(i)} = Z_n$. The kF-CV estimates of the l_2 errors (Equation (3.4)) by computing the loss of a point in the i^{th} fold $\mathbf{Z}^{(i)}$ compared to the prediction of the surrogate model built on the remaining folds ($\mathbf{Z}_n \setminus \mathbf{Z}^{(i)}$).

$$\mathcal{R}_{k-CV}(m) = \frac{1}{n} \sum_{i=1}^{k} \sum_{(x',y') \in \mathbf{Z}^{(i)}} l_2(\widehat{m_{|\mathbf{Z}_n \setminus \mathbf{Z}^{(i)}}}(x'), y'), \tag{3.4}$$

where

$$\mathbf{z} \in \mathbf{Z}_n \setminus \mathbf{Z}^{(i)} \text{ if and only if } \mathbf{z} \in \mathbf{Z}_n \text{and} \mathbf{z} \notin \mathbf{Z}^{(i)}.$$

Queipo et al. [QHS+05] pointed out that the main advantage of CV is that it provides a nearly unbiased estimate. Further, Kohavi et al. [Koh95] studied Cross-Validation and Bootstrap performances on a large dataset and recommended using stratified 10-fold-cross-validation. [JWHT13] stated that kF-CV with $k = 5$ or $k = 10$ yield test error estimates that suffer neither from excessively high bias nor from very high variance.

3.2.4 Penalization

Penalties are used in several model selection frameworks in order to prevent over-fitting. Selection criteria such as the Bayesian Information Criterion (BIC) [Sch78] or Akaike Information Criterion (AIC) [Aka74] penalize the models by their degrees of freedom. Most penalties are designed for a particular family of surrogates. Here, we are interested in universal methods. So that, we prefer to deal with the smoothness of the surrogate

model rather than with its structural complexity. For instance, [NCK+11] introduce a criterion called Linear Reference Model (LRM). It scores a surrogate model by computing the deviation between its predictions and a local linear model $\widehat{l_{rm}}$. The LRM is computed over a set of N points $x^{(k)}$ for $k = 1, \ldots, N$ (see Equation (3.5)).

$$\mathcal{R}_{LRM}(m) = \frac{1}{N} \sum_{k=1}^{N} l_2(\widehat{m_{|\mathbf{Z_n}}}(x^{(k)}), \widehat{l_{rm}}(x^{(k)})) \tag{3.5}$$

Computationally, this last criteria needs the construction of a Delaunay tessellation [Wat81] to compute $\widehat{l_{rm}}$. The computational cost of such construction in high dimension is too expensive. We suggest to use a criterion that penalize the roughness of surrogate models: the thin plate spline (TPS) [Duc77] Bending Energy Functional (BEF). It is a second order partial derivatives-based penalty. For a dimension d, the roughness penalty E_n is the integral of the squared term of the Hessian (Equation (3.6)).

$$E_n(\hat{f}) = \int_{\Omega} \sum_{i=1}^{d} \sum_{j=1}^{d} \Big(\frac{\partial^2 \hat{f}}{\partial x_i \partial x_j} \Big)^2 dx \tag{3.6}$$

LRM can be used in place of the BEF in the selection criterion PPS. It penalizes the deviation from a linear model regardless of its roughness. It still gives good predictive capabilities also. Nevertheless, some rough surrogates may be selected.

3.3 Surrogate model ensemble: PPS-OS

3.3.1 Overview

Surrogate model selection consists in selecting a surrogate model among a collection of them. This means that we evaluate the performances of several surrogate models and then choose one of them. Acar et al. [ARR09] stated that *this practice has some shortcomings as it does not take full advantage of the resources devoted to constructing different metamodels*. In fact, it is possible to consider a weighted combination of surrogates without any significant extra computational cost. These combinations are called: ensembles, aggregations and multiple surrogates.

Forrester and Keane [FK09] show that these aggregation methods drastically improve the performances of the surrogate models. In general, ensembles require small computational resources compared to the cost of the simulations [QHS+05]. The general form of an aggregation of p surrogate models $\widehat{m^{(i)}}_{|\mathbf{Z_n}}$, for $i = 1, \ldots, p$ is given in Equation (3.7):

$$\widehat{A}_{|\mathbf{Z_n}}(x) = \sum_{i=0}^{p} w_i(x)\widehat{m^{(i)}}_{|\mathbf{Z_n}}(x) \tag{3.7}$$

For instance, Zerpa et al. [ZQPS05] considered a local combination called weighted average model where the weights are based on the local expected variances of the surrogate models. Goel et al. [GHSQ07] extended the use of ensembles to the identification of region with high error. They presented also several heuristics to weight ensembles.

However, Gorissen et al. [GDT09] used a simple average ensemble (all the weights are equal). Muller et al. [MP11] proposed to weight the aggregation using the Dempster-Shafer theory where the error estimates are used as basic probability assignments. Viana et al. [VHS09] proposed to use an ensemble of surrogate models that minimize the CV errors. In fact, if for $k = 1, \ldots, n$, $\mathbf{v_k}$ is the vector of CV errors of the surrogate model $\widehat{m^{(i)}}_{|\mathbf{Z_n}}$, the CV errors of the aggregation is then $W^\top CW$. The weights are selected to minimize the CV errors of the aggregation under the constraint $\sum_{i=1}^{p} w_i = 1$. The optimal weighted surrogate OWS is obtained using the weights of Equation (3.8).

$$\mathbf{W} = \frac{\mathbf{C}^{-1}\mathbf{1}}{\mathbf{1}^\top\mathbf{C}^{-1}\mathbf{1}} \tag{3.8}$$

where the elements of the matrix $\mathbf{C}$, $c_{ij} =< \mathbf{v_i}, \mathbf{v_j} >$. Viana et al. [VHS09] noticed that the solution may include negative values. They stated that this additional freedom to the weights estimation amplify errors. In fact, the matrix $\mathbf{C}$ is an approximation of the covariance of the errors of the surrogate models. To overcome the problem, the authors suggested to use only the diagonal elements of $\mathbf{C}$. Then, the weights are $w_i = \frac{c_{ii}^{-1}}{\sum_{k=1}^{n} c_{kk}^{-1}}$. This formulation is close to the weights of the PRESS weighted surrogate (PWS) given in [GHSQ07] (equation (3.9)), with $\alpha = 0, \beta = -2$.

$$w_i = \frac{(\sqrt{c_{ii}} + \frac{\alpha}{n}\sum_{j=1}^{n}\sqrt{c_{jj}})^{\beta}}{\sum_{k=1}^{n}(\sqrt{c_{kk}} + \frac{\alpha}{n}\sum_{j=1}^{n}\sqrt{c_{jj}})^{\beta}} \tag{3.9}$$

3.3.2 PPS-optimal ensemble

Let us consider $(\widehat{m^{(1)}}_{|\mathbf{Z_n}}, \ldots, \widehat{m^{(n)}}_{|\mathbf{Z_n}})$ a set of p surrogate models. Let A be an aggregation of these surrogate models weighted by the vector $\mathbf{W} = (w_1, .., w_n)$ (Equation (3.10)).

$$\widehat{A}(x) = \sum_{k=1}^{p} w_k \widehat{m^{(k)}}_{|\mathbf{Z_n}}(x) \tag{3.10}$$

In our formulation, we compute the weights of the aggregations by optimizing the PPS of the aggregation under the constraint $\sum_{i=1}^{p} w_i = 1$. The PPS-Optimal aggregation is then the aggregation in which the weights are the solution of the optimization Problem (3.11).

$$\begin{aligned} \min_{W} \quad & PPS(A, \mathbf{Z_n}) \\ \text{u.c.} \quad & \sum_{i=1}^{p} w_i = 1 \end{aligned} \tag{3.11}$$

For each k in $1, \ldots, p$, let:

- $\mathbf{e_k}$ be the vector of errors on design points.
- $\mathbf{v_k}$ the vector of cross-validation error of the surrogate model $\widehat{m^{(k)}})_{|\mathbf{Z_n}}$.

Notice then that the MSE of the aggregation is a quadratic form of the weights

$$\mathcal{R}_{l_2,\widehat{P_n}}(A) = \left\| \sum_{i=1}^{p} w_i e_i \right\|^2 = \mathbf{W}^T \mathbf{E} \mathbf{W}, \tag{3.12}$$

where the elements of $\mathbf{E}$, $E_{ij} =< e_i, e_j >$. Similarly, the cross validation errors of the aggregation is also a quadratic form of the weights (Equation (3.13)) where $\mathbf{C}$ is the same defined in the previous section.

$$\mathcal{R}_{CV}(A) = \mathbf{W}^T \mathbf{C} \mathbf{W} \tag{3.13}$$

Last, the energy functional is also a quadratic form of the weights (Equation 3.14).

$$\begin{aligned} E_n(\widehat{A}) &= \int_\Omega \sum_{i=1}^{d}\sum_{j=1}^{d} \Big(\frac{\sum_{k=1}^{p} w_k \partial^2 \widehat{m^{(k)}}|_{\mathbf{Z_n}}(x)}{\partial x_i \partial x_j} \Big)^2 dx \\ &= \mathbf{W}^T \mathbf{K} \mathbf{W}, \end{aligned} \tag{3.14}$$

where:

$$\mathbf{K} = \Big[k_{kl} = \sum_{i=1}^{d}\sum_{j=1}^{d} \int_\Omega \Big(\frac{\partial^2 \widehat{m^{(k)}}|_{\mathbf{Z_n}}(x)}{\partial x_i \partial x_j} \Big) \Big(\frac{\partial^2 \widehat{m^{(l)}}|_{\mathbf{Z_n}}(x)}{\partial x_i \partial x_j} \Big) dx \Big].$$

Let $\mathbf{R} = \alpha\mathbf{E} + \beta\mathbf{C} + \gamma\mathbf{K}$. The PPS of the aggregation is then a quadratic form of the weights $\mathbf{W}$: $PPS(\widehat{A}) = \mathbf{W}^T\mathbf{R}\mathbf{W}$. The PPS-Optimal aggregation is then the aggregation that minimizes the PPS under the constraint $\sum_{i=1}^{n} w_i = 1$. The solution is defined in Equation (3.15):

$$\mathbf{W}^\star = \frac{\mathbf{R}^{-1}\mathbf{1}}{\mathbf{1}^\top \mathbf{R}^{-1}\mathbf{1}} \tag{3.15}$$

Similarly to Equation (3.8), the solution of Equation (3.15) may include negative weights as well as weights greater than one. Unlike, in [VHS09] in which the writers suggested to use only the diagonal terms in the matrix to ensure the positivity, here we tolerate such weights since this freedom is controlled by the BEF penalization. As a matter of fact, the BEF penalization prevents to artificial oscillations on the aggregated surrogate.

3.3.3 Illustrative example

We consider the example in Figure 3.1. The ensemble is the optimal trade-off defined by the PPS parameters. The ensemble is relatively smoother than the interpolating ones of the initial collection. Further, its CV error is lesser than the best prediction of this collection.

3.3.4 One shot metamodel selection: PPS-OS

We suppose that we have at hands p possible surrogate model builders where p is relatively small (typically $p \leq 35$). One select the model that has the best PPS. In order to improve the result, we select the PPS-Optimal ensemble. We consider this procedure (described in Algorithm 1) as a model selection algorithm. Notice that the aggregation does not

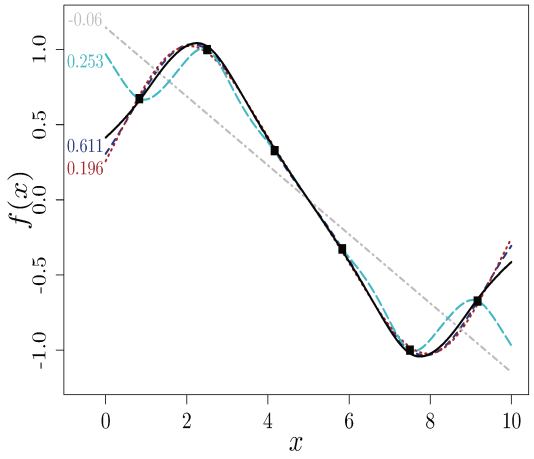


Figure 3.1: Example of PPS-Optimal ensemble, Dashed lines: 4 meta-models predictions. Solid line: PPS-optimal ensemble predictions. Black squares: design points

increase significantly the computational cost of the procedure as the errors have been generally previously evaluated.

Algorithm 1: PPS One Shot (PPS-OS) model selection algorithm

Data: Design Points $\mathbf{Z_n}$

Generate the list of first population of surrogate models builder

$$L = (m_1, m_2, \ldots, m_p)$$

Compute the PPS-Optimal aggregation $\widehat{A}$;

Result: $\widehat{A}$.

In our implementation, PPS-OS selects the PPS-optimal aggregation of 32 surrogate models from 4 different surrogate types (Kriging, SVM, Polynomial regression and MLS).

3.4 PPS-based Genetic Aggregation for model selection : (PPS-GA)

As discussed in the previous section, the use of PPS to perform model selection is straightforward if the number of the available surrogate model is moderate. In that case, one can consider a weighted PPS-Optimal aggregation of all the possible surrogate models. However, there are many types of surrogate models and each type has several possible settings. For instance, to tune a universal kriging surrogate model, there are various possible choices for covariance function and trend function. Consequently, one cannot evaluate the PPS for all the possible combinations. Even with a good selection criterion, one need to explore the space of available surrogate models to select the best one.

Gorissen et al. [GDT09] proposed an evolutionary algorithm to perform surrogate model selection and to explore the space of surrogate models. The surrogate models are considered as the individuals of the population. The settings of the surrogate models are considered as the genetic information of the individuals. The mutation and cross-over operators between two surrogate models of the same type are performed by modifying or exchanging the surrogate models settings. Further, they generate an equally weighted surrogate model ensemble when the cross-over is between two surrogate models of different types. Their algorithm uses the island model of evolutionary algorithms.

We now introduce our selection algorithm based on the genetic aggregation called PPS-GA. Similarly to [GDT09]'s heuristic, the mutation and cross-over operators are performed over surrogate model builders' settings. In our algorithm, all the aggregation weights are now optimized according to the PPS. Moreover, we add new aggregations at each iteration. The members of these aggregations are generated randomly. Further, we do not adopt the island model. We consider that the heterogeneous set of surrogate model builders "lives" together in the same space. The selection method is designed to conserve the diversity.

In our implementation, we consider several surrogate types with various settings: Kriging, moving least squares, polynomial regression and support vector machines regression. PPS-GA has another interesting property. It is easy to enrich the set of surrogate model builders. In fact, the algorithm does not require any particular assumption. It is in part due to the universality of PPS.

Algorithm 2: *PPS* Genetic Aggregation (*PPS*-GA) model selection algorithm

Inputs: Design Points $\mathbf{Z_n}$, $l = 10$.

Generate the list of first surrogate models builders $L = (m_1, m_2, \ldots, m_k)$.

for *Generation = 1 to MaxGeneration* **do**

m_{agg} = Compute the optimal aggregation of the l best surrogate models according to PPS

L_{new} = Perform mutation and cross-over operations

$L = L \cup L_{new} \cup m_{agg}$

L = Select the best k surrogate models according to PPS.

end

$m^{\star}_{|\mathbf{Z_n}}$ = Select the best surrogate model of L.

Outputs: $m^{\star}_{|\mathbf{Z_n}}$

3.5 Numerical examples

3.5.1 Benchmark problems

In order to check the efficiency of *PPS*-OS and *PPS*-GA, we tested their performances on a benchmark of 15 functions (see Table 3.1 and formula given in Appendix. 3.8).

For each function, we generated 10 different optimized maximin Latin hypercube sampling (LHS) [MBC79] of size N. We generated an extra test set of size $n_t = 1000 \times N$ by a fast optimized LHS algorithm [VVB10]. We use the $RMSE$ criterion (Equation (3.16)) to evaluate the performances on the set of verification points.

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}(y_i - \widehat{y}_i)^2} \tag{3.16}$$

For each function, we compare the performance of the selection algorithms (*PPS*-OS and *PPS*-GA) to the performances of 4 witness surrogate models:

a) A kriging surrogate model using an an-isotropic Matérn 5/2 kernel and a linear trend function.

b) A support vector regression using a Gaussian kernel and ϵ-regression paradigm.

Name	Dimension d	Number of design points N	Number of test points n_t
1. Wing weight	10	45	45000
2. Borehole	8	40	40000
3. Dette & Pepelyshev (8-Dim)	8	75	75000
4. Piston simulation	7	60	60000
5. OTL circuit	6	35	35000
6. Gramacy & Lee (2009)	6	85	85000
7. Friedman	5	35	35000
8. Dette & Pepelyshev exponential	3	16	16000
9. Dette & Pepelyshev curved	3	18	18000
10. Lim non-polynomial	2	12	12000
11. Currin exponential	2	20	20000
12. Franke's	2	10	10000
13. Gramacy & Lee (2008)	2	45	45000
14. Sasena	2	10	10000
15. Gramacy & Lee (2012)	1	15	15000

Table 3.1: Test functions

c) A moving least squares surrogate model using a Gaussian weighting function and second order polynomial regression.

d) Full second order polynomial regression, we use least-norm when the equation system is undetermined.

These surrogates are selected among the 32 surrogates of $PPS - OS$ as follows: We consider the 150 functions (15 $\times$10 repetitions). For each surrogate $\widehat{m}$, we compute $N_{best}(\widehat{m})$: the number of times where $\widehat{m}$ is the best individual surrogate. Each witness surrogate models is the one with highest N_{best} among its type. The surrogate with the highest N_{best} is the kriging using an an-isotropic Matérn 5/2 kernel and a linear trend function. It is the best individual surrogate in 25 test (16%).

3.5.2 Results

We display the results of the benchmark in Table 3.2 and in Figures 3.2-3.16:

- In Table 3.2, the median and the standard deviation of the RMSE of each surrogate model are given. The best median value is in bold.
- In Figures 3.2-3.16, the box-plots illustrate the variability with respect to the design set.

The results show the efficiency of the selection algorithms: the models selected by PPS-OS and PPS-GA outperform each individual surrogate models in the predictive capabilities for at least one function. Generally, the RMSE of the selected surrogates is generally either the best or close to the best one.

3.5.3 PPS-based ensembles

We also use the same test bench to compare the PPS-optimal ensemble, the OWS ensemble and the PWS ensemble with $\alpha = 0.05$ and $\beta = -1$. Here, we have at hands 10 surrogate models and we compute the weights by these three different techniques. The results are given in Figure 3.17. In order to display all the results in the same figure, we have rescaled the values of all the bench functions in [0,1].

Generally, PPS-optimal ensemble give the best result except for the Dette & Pepelyshev Exp function where the PWS is better and for the Dette & Pepelyshev 8-Dim

	MLS		SVM		Poly		Kriging		*PPS*-OS		*PPS*-GA	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
Wing Weight	6.646	0.500	12.89	0.225	15.890	4.332	5.800	1.076	3.873	0.708	**3.701**	0.560
Borehole	12.08	1.933	13.27	0.442	1341	2050	9.014	2.128	**3.197**	0.418	3.627	0.467
Dette & Pepelyshev 8-Dim	14.57	11.52	5.236	0.134	10.82	9.006	**1.771**	0.780	1.995	0.902	3.609	0.162
Piston Simulation	0.037	0.002	0.040	0.001	0.087	0.083	0.016	0.006	**0.011**	0.001	0.014	0.003
OTL Circuit	0.287	0.141	0.312	0.004	0.303	0.172	0.112	0.037	**0.036**	0.011	0.055	0.013
Gramacy & Lee 2009	1.421	0.498	0.669	0.012	1.223	0.667	0.410	0.092	**0.243**	0.139	0.380	0.179
Friedman	4.215	1.607	1.522	0.107	4.218	1.714	1.251	0.244	**0.634**	0.284	0.854	0.195
Dette & Pepelyshev Exp	**0.955**	0.038	2.860	0.147	0.998	0.032	3.280	0.175	1.139	0.362	1.293	0.665
Dette & Pepelyshev Curved	1.765	0.129	3.330	0.146	2.034	0.048	2.466	0.796	**1.414**	0.409	1.821	0.592
Lim Non Polynomial	0.395	0.044	0.374	0.048	0.433	0.037	**0.251**	0.033	0.441	0.187	0.460	0.095
Currin Exp	0.970	0.142	1.049	0.098	1.331	0.050	0.692	0.324	0.554	0.268	**0.438**	0.199
Franke	0.093	0.007	0.062	0.004	0.132	0.002	0.060	0.016	**0.052**	0.010	0.062	0.013
Gramacy & Lee 2008	0.058	0.002	0.069	0.001	0.074	0.001	0.040	0.006	**0.035**	0.008	**0.035**	0.006
Sasena	2.942	0.056	3.512	0.119	4.423	0.358	2.434	0.399	2.341	0.608	**2.138**	0.504
Gramacy & Lee 2012	**0.426**	0.067	0.527	0.097	0.508	0.034	0.456	0.071	0.458	0.073	0.471	0.127

Table 3.2: Mean and Standard deviation of RMSE

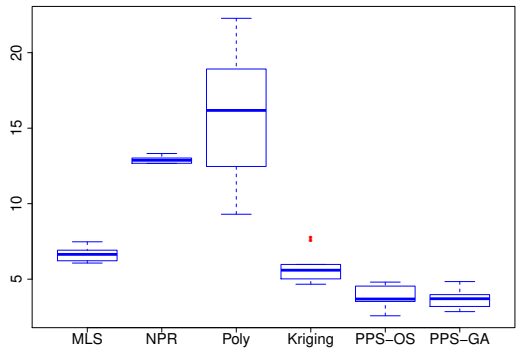


Figure 3.2: Wing weight function

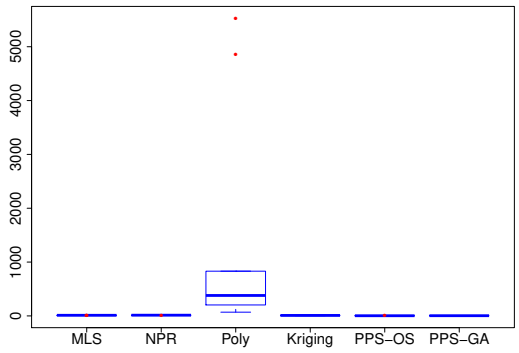


Figure 3.3: Borehole function

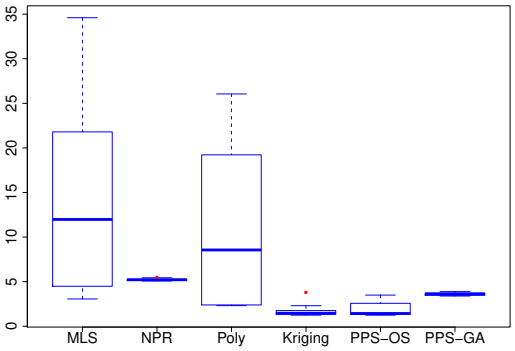


Figure 3.4: D & P (8-Dim) function

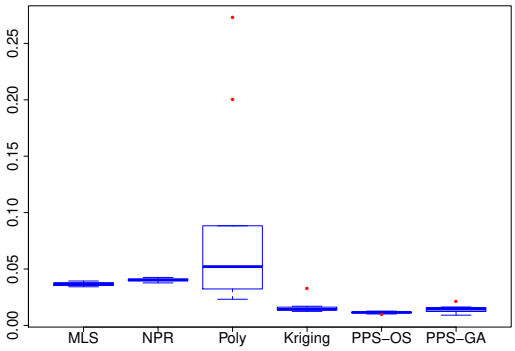


Figure 3.5: Piston simulation function

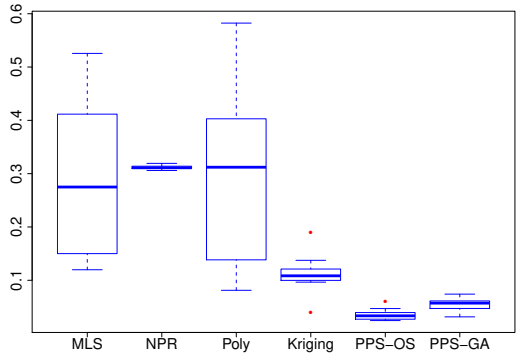


Figure 3.6: OTL circuit function

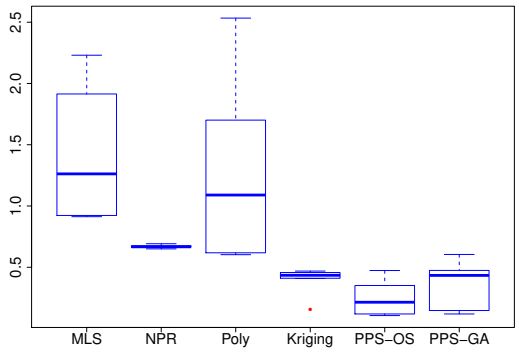


Figure 3.7: G & L 2009 function

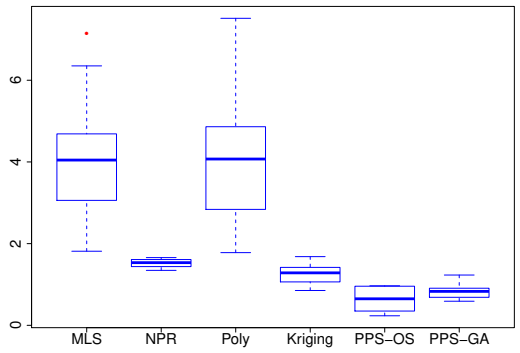


Figure 3.8: Friedman function

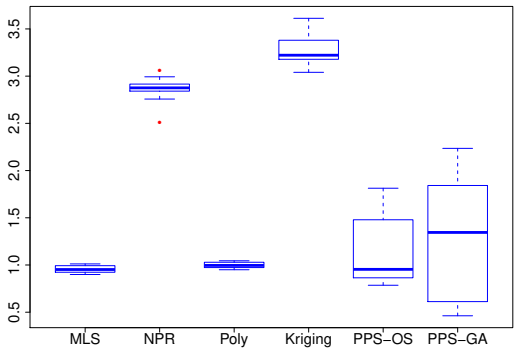


Figure 3.9: D & P exponential function

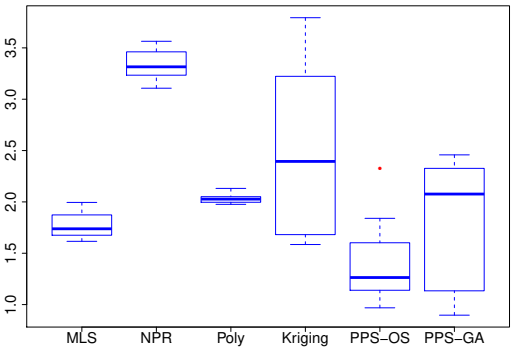


Figure 3.10: D & P curved function

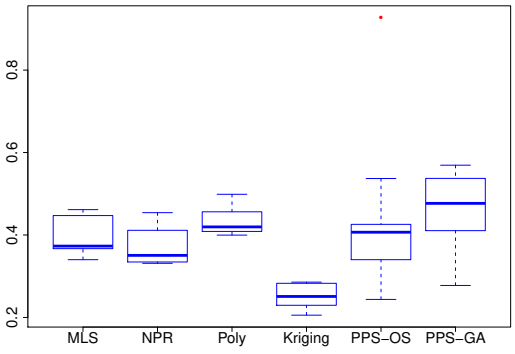


Figure 3.11: Lim non-polynomial function

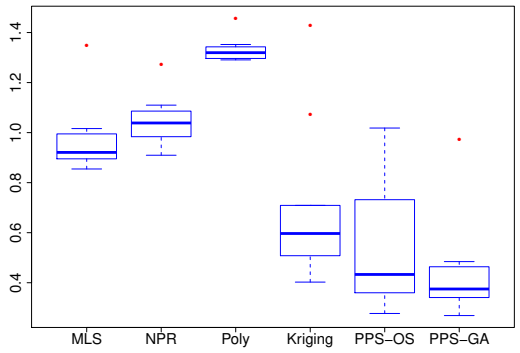


Figure 3.12: Currin exponential function

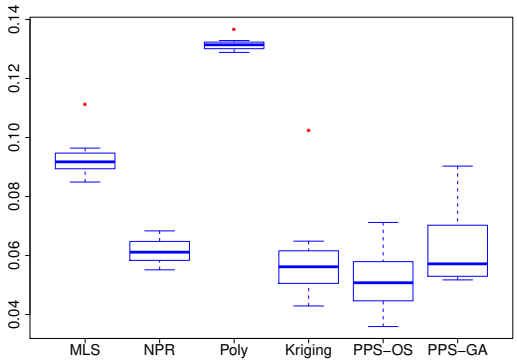


Figure 3.13: Franke function

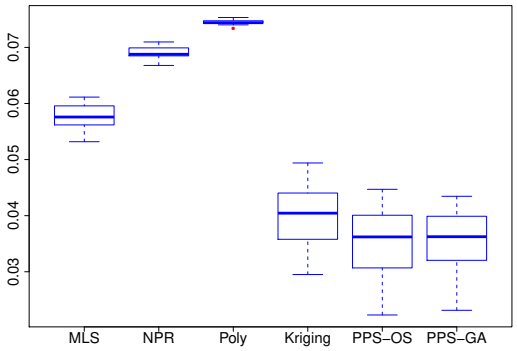


Figure 3.14: G & L (2008) function

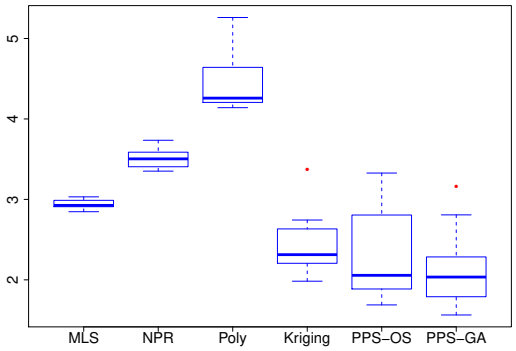


Figure 3.15: Sasena function

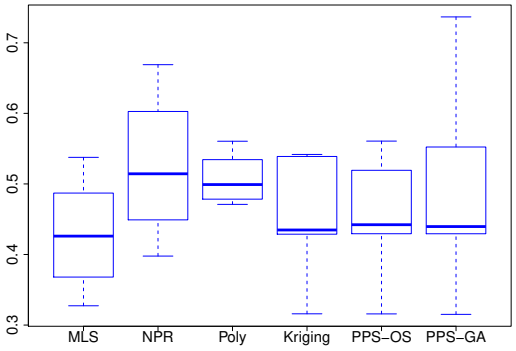
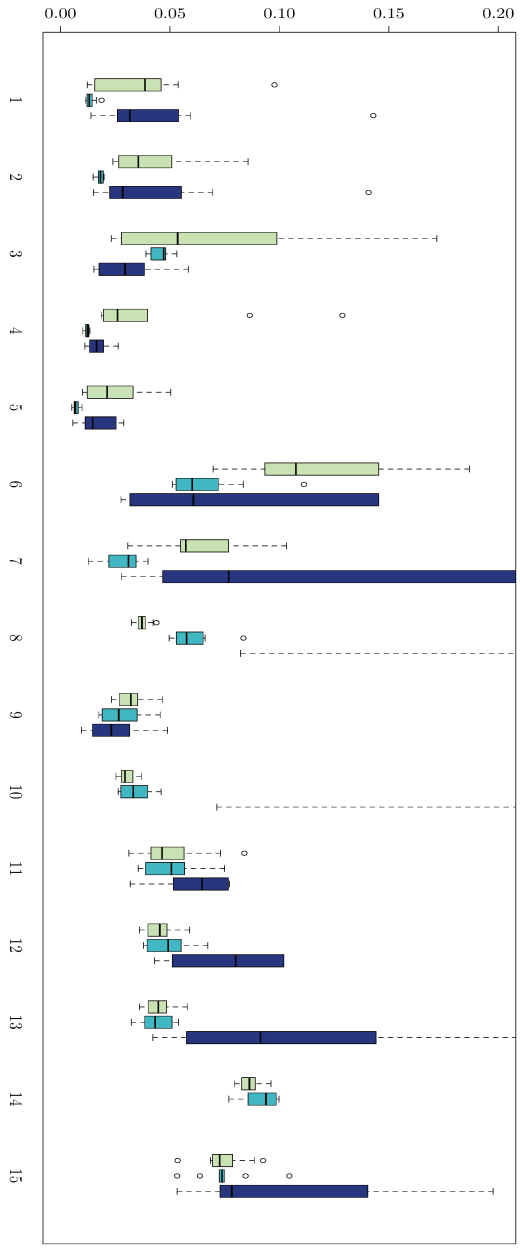


Figure 3.16: G & L 2012 function

Figure 3.17: For each function: Left: PWS method in light green. Middle: PPS-optimal ensemble in light blue. Right: OWS ensemble in dark blue. The function number is as in Table (3.1)



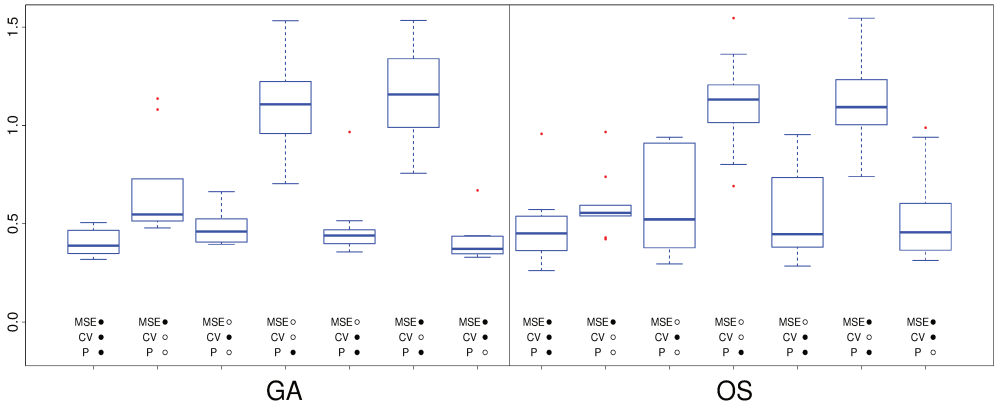


Figure 3.18: Several selection criteria: Currin function

function where OWS is better. Moreover, for all the functions, PPS-optimal ensemble never has the worst RMSE. This shows the suitability of PPS to construct ensembles.

3.5.4 On the choice of α, β and γ

Further, recall that the PPS criterion is composed of 3 components: MSE, CV and Pen. It is then useful to understand the role played by each component in the performances. To do so, we compared the performance of the algorithms based on the PPS (PPS-GA and PPS-OS) to same algorithms in which we replaced the selection criterion by only one or a sum of two components of the PPS. Therefore, the entire set of tested algorithms are:

- PPS-based algorithms (PPS-GA, PPS-OS) MSE● CV● P●
- MSE only MSE● CV○ P○
- CV only MSE○ CV● P○
- Penalty only MSE○ CV○ P●
- CV + Penalty MSE○ CV● P●
- MSE + Penalty MSE● CV○ P●
- MSE + CV MSE● CV● P○

Here, we used LRM as penalty because it gives a reasonable prediction quality when used as a selection criterion. BEF cannot be used in this context. It is only a penalty

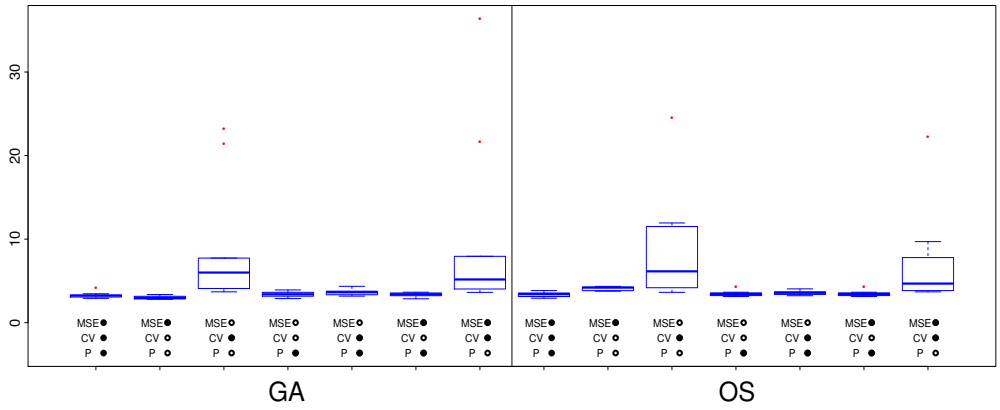


Figure 3.19: Several selection criteria: Sasena function

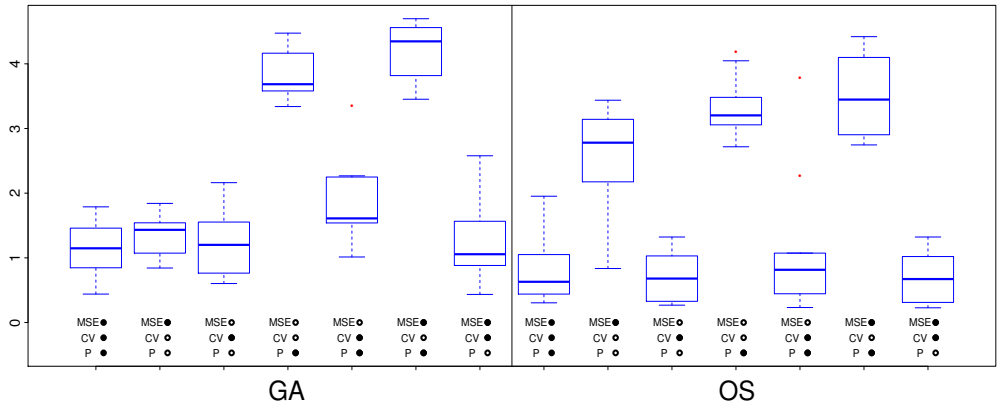


Figure 3.20: Several selection criteria: D&P curved function

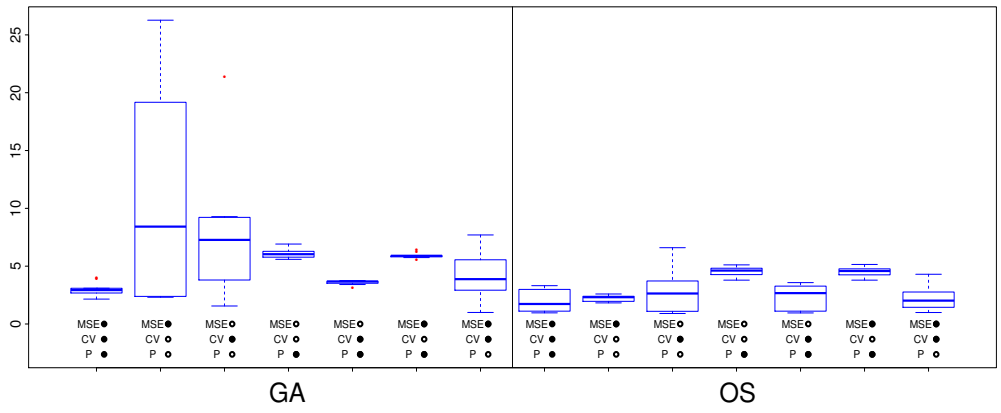


Figure 3.21: Several selection criteria: D&P 8-dim function

that favors monomials and constant functions regardless of the output value. We display in Figures 3.18, 3.19, 3.20 and 3.21 the results of the comparison between the different algorithms for some functions of Table 3.1: Currin function, Sasena function, Dette & Pepelyshev curved function and Dette & Pepelyshev 8-dim function. For each function, 10 different maximin LHS design are generated of size $N = 10d$ where d is the space dimension.

For these functions, we can notice how the different components of the PPS act together to select a convenient surrogate model in different scenarios. In fact, the results highlight the effect of each component. Obviously, neither a single criterion nor any combination of two criteria is better than PPS in all the cases. This is due to:

- Any interpolating surrogate model is MSE-optimal. It is a misleading criterion to the overall errors.
- CV is a convenient estimate of the predictive capabilities. But, it is a pessimistic one.

We also study the choice of the values of the parameters of the PPS on the benchmark. We used ten surrogate models and we computed the sum of RMSE for each value of β and γ, α being fixed to 1. Let $(\beta^\star, \gamma^\star)$ denotes the global minimum. We display the contour plot of the sum of mean square errors (MSE) in Figure 3.22. Notice that the proposed values of Section 3.2 $(1, 0.5, 0.25)$, are close to the optimum. Further, they give

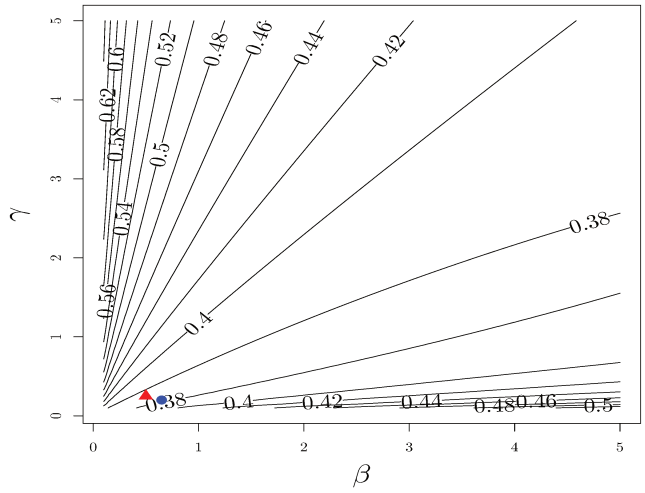


Figure 3.22: Contour plot of the sum of scaled MSE of 150 test functions (15×10 repetitions), Blue circle: optimum of sum of RMSE, Red triangle: our proposed value.

better sum of MSE error 0.3760 if compared to $(\beta^\star, \gamma^\star)$ that leads to 0.3771. We display also the errors of the ensembles using these two parameters in Appendix 3.7. Notice that the prediction errors are close.

3.5.5 On the relevance of ensembles

Finally, we display in Figure 3.23 the number of surrogate models in the selected surrogate of PPS-GA. We notice that the algorithm selects generally a PPS-optimal ensemble. This is due in part to the PPS suitability to ensemble construction and it shows that aggregations are relevant in metamodel selection. This shows the usefulness of the ensemble approach.

3.5.6 Computational cost

We give in Table 3.3, the quantiles and the sum of the computing time of all the 150 benchmark functions. It is expected that the selection methods needs more time than individual surrogates. We can notice also that PPS-GA is computationally more expensive than PPS-OS. This is due to the cost of exploration. Finally, notice that these values

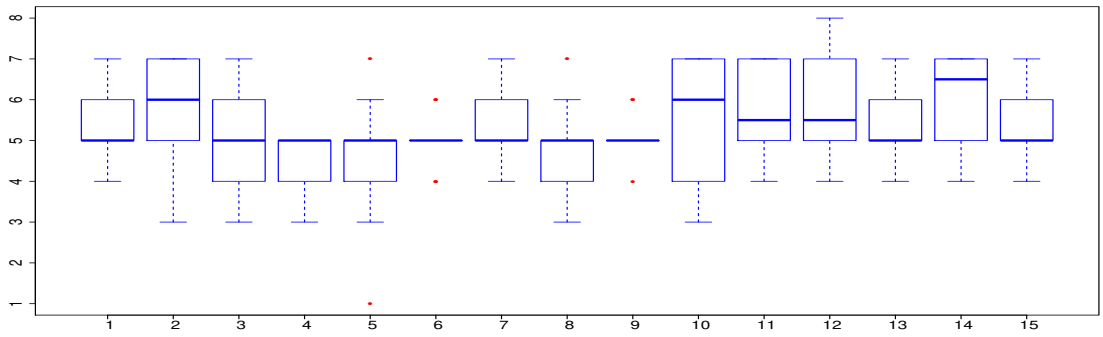


Figure 3.23: Number of members in the best ensemble

are negligible compared to the computing time of one complex simulation.

	MLS	NPR	Poly	Kriging	PPS-OS	PPS-GA
0%	0.000	0.000	0.000	0.000	0.012	0.386
25%	0.000	0.001	0.000	0.001	0.065	0.933
50%	0.000	0.001	0.000	0.014	0.336	1.769
75%	0.000	0.003	0.003	0.068	0.886	2.865
100%	0.001	0.027	0.022	0.136	1.884	5.055
Sum	0.007	0.394	0.406	5.398	79.758	297.509

Table 3.3: Elapsed time in seconds to construct each surrogate model

3.6 Conclusion

In this paper, we propose a new selection criterion called the penalized predictive score. PPS can be computed for all the types of surrogate models. By construction, PPS is especially suitable for functions that have specific characteristics such as regularity and smoothness. Generally these characteristics are implicitly expected with the meta-modeling framework. We showed also that it enables the construction of relevant ensembles. The PPS-optimal ensemble are easily computed and avoid over-fitting.

We study also two surrogate model selection schemes based on the PPS. The first one compute the PPS-optimal ensemble rather than selecting one surrogate model. The

second one is based on a evolutionary framework that enables the exploration of the space of surrogate models. Tests shows that the proposed algorithms give very good results. It remains important to notice that this algorithm does not necessarily give an accurate approximation in all the cases. For instance, the algorithm will fail if we use a small amount of observations for a highly nonlinear behavior. It aims at selecting the best surrogate among the possible choices. Assessing the level of confidence of a prediction is left for future research.

Acknowledgments Malek Ben Salem is funded by a CIFRE grant from the ANSYS company, subsidized by the French National Association for Research and Technology (ANRT, CIFRE grant number 2014/1349). We gratefully thank Olivier Roustant and Fabrice Gamboa for the help in writing this paper and for their valuable remarks. We also warmly thank two anonymous reviewers for their valuable comments.

3.7 Appendix A: Comparison between the proposed PPS parameters and optimal parameters

In Figure 3.24, we use the same test functions to compare the proposed PPS parameters and the optimal parameters.

3.8 Appendix B: Test functions

The equations and the input parameter space of the functions of Table 3.1 are defined below:

1/ Wing weight function:
Parameters: $S_w \in [150, 200]$, $W_{fw} \in [220, 300]$, $A \in [6, 10]$, $\gamma \in [-10, 10]$,
$q \in [16, 45]$, $\lambda \in [0.5, 1]$, $t_c \in [0.08, 0.18]$, $N_z \in [2.5, 6]$,
$W_{dg} \in [1700, 2500]$, $W_p \in [0.025, 0.08]$

For $\mathbf{x} = (S_w, W_{fw} A, \gamma, q, \lambda, t_c, N_z, W_{dg}, W_p)$

$$f_1(\mathbf{x}) = 0.036 S_w^{0.758} W_{fw}^{0.758} \Big(\frac{A}{\cos^2(\gamma)}\Big)^{0.6} q^{0.006} \lambda^{0.04} \quad \Big(\frac{100 t_c}{\cos(\gamma)}\Big)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p \tag{3.17}$$

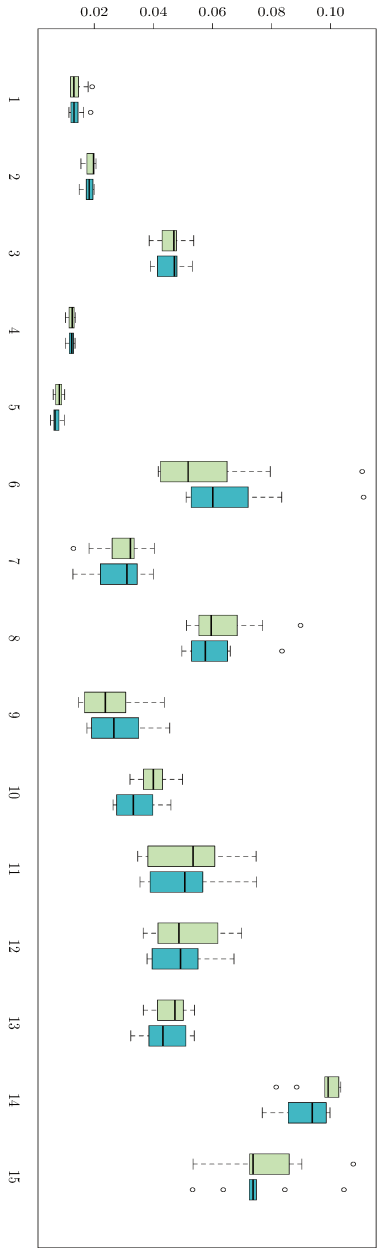


Figure 3.24: The scaled RMSE for each PPS-optimal ensemble, For each function: Left: using $(\alpha, \beta, \gamma) = (1, \beta^\star, \gamma^\star)$ in light green. Right: using $(\alpha, \beta, \gamma) = (1, 0.5, 0.25)$ in light blue. The function number is as in Table (3.1)

2/ Borehole function:
Parameters: $r_w \in [0.05, 0.15]$, $r \in [100, 50000]$, $T_u \in [63070, 115600]$, $H_u \in [990, 1110]$, $T_l \in [63.1, 116]$, $H_l \in [700, 820]$, $L \in [1120, 1680]$, $K_w \in [9855, 12045]$

$$\begin{aligned} &\text{For } \mathbf{x} = (r_w, r, T_u, H_u, T_l, H_l, L, K_w) \\ &f_2(\mathbf{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln(\frac{r}{r_w})\Big(1 + \frac{2LT_u}{ln(\frac{r}{r_w})r_w^2 K_w} + \frac{T_u}{T_l}\Big)} \end{aligned} \tag{3.18}$$

3/ Dette & Pepelyshev (2010a):
Parameters: for all $i = 1, \ldots, 8$, $x_i \in [0, 1]$

$$\begin{aligned} f_3(\mathbf{x}) = &\, 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 \\ &+ 16\sqrt{x_3 + 1}(2x_3 - 1)^2 + \sum_{i=4}^{8} i \ln(1 + \sum_{j=3}^{i} x_j) \end{aligned} \tag{3.19}$$

4/ Piston simulation function:
Parameters: $M \in [30, 60]$, $S \in [0.005, 0.020]$, $V_0 \in [0.002, 0.010]$, $k \in [1, 5] \times 10^3$, $P_0 \in [9, 11] \times 10^4$, $T_a \in [290, 296]$, $T_0 \in [340, 360]$

$$f_4(\mathbf{x}) = 2\pi \sqrt{\frac{M}{k + S^2 \frac{P_0 V_0}{T_0} \frac{T_a}{V^2}}} \tag{3.20}$$

where:

$$V = \frac{S}{2k}\Big(\sqrt{A^2 + 4k\frac{P_0 V_0}{T_0} T_a} - A\Big)$$

and

$$A = P_0 S + 19.62M - \frac{kV_0}{S}.$$

5/ OTL circuit function:
Parameters: $R_{b1} \in [50, 150]$, $R_{b2} \in [25, 70]$, $R_f \in [0.5, 3]$, $R_{c1} \in [1.2, 2.5]$, $R_{c1} \in [0.25, 1.2]$, $\beta \in [50, 300]$

$$\begin{aligned} f_5(\mathbf{R}, \beta) = &\, \frac{(\frac{12R_{b2}}{R_{b1}+R_{b2}} + 0.74)\beta(R_{c2} + 9)}{\beta(R_{c2} + 9) + R_f} \\ &+ \frac{11.35R_f}{\beta(R_{c2} + 9) + R_f} + \frac{0.75R_f\beta(R_{c2} + 9)}{(\beta(R_{c2} + 9) + R_f)R_{c1}} \end{aligned} \tag{3.21}$$

6/ Gramacy & Lee (2009) function:
Parameters: for all $i = 1, \ldots, 6$, $x_i \in [0, 1]$

$$f_6(\mathbf{x}) = \exp[\sin((0.9(x_1 + 0.48))^{10})] + x_2 x_3 + x_4$$

7/ Friedman function:
Parameters: for all $i = 1, \ldots, 5$, $x_i \in [0, 1]$

$$f_7(\mathbf{x}) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$$

8/ Dette & Pepelyshev exponential function:
Parameters: for all $i = 1, \ldots, 3$, $x_i \in [0, 1]$

$$f_8(\mathbf{x}) = 100(e^{-2/x_1^{1.75}} + e^{-2/x_2^{1.5}} + e^{-2/x_3^{1.25}})$$

9/ Dette & Pepelyshev curved function:
Parameters: for all $i = 1, \ldots, 3$, $x_i \in [0, 1]$

$$f_9(\mathbf{x}) = 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2 \tag{3.22}$$

10/ Lim non-polynomial function:
Parameters: $x_1, x_2 \in [0, 1]$

$$f_{10}(\mathbf{x}) = \frac{1}{6}[(30 + 5x_1 \sin(5x_1))(4 + \exp(-5x_2)) - 100]$$

11/ Currin exponential function:
Parameters: $x_1, x_2 \in [0, 1]$

$$f_{11}(\mathbf{x}) = [1 - \exp(-\frac{1}{2x_2})] \times \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \tag{3.23}$$

12/ Franke function:
Parameters: $x_1, x_2 \in [0, 1]$

$$\begin{aligned} f_{12}(\mathbf{x}) = {} & 0.75 \exp(-\frac{(9x_1 - 2)^2 + (9x_2 - 2)^2}{4}) + 0.75 \exp(-\frac{(9x_1 + 2)^2}{49} - \frac{9x_2 + 1}{10}) \\ & + 0.5 \exp(-\frac{(9x_1 - 7)^2}{4} - \frac{(9x_2 - 3)^2}{4}) + 0.2 \exp(-(9x_1 - 4)^2 - (9x_2 - 7)^2) \end{aligned} \tag{3.24}$$

13/ Gramacy & Lee (2008) function:
Parameters: $x_1, x_2 \in [-2, 6]$

$$f_{13}(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2)$$

14/ Sasena function:
Parameters: $x_1, x_2 \in [0.0, 5]$

$$f_{14}(\mathbf{x}) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 + 7 \sin(0.5x_1) \sin(0.7x_1 x_2) \tag{3.25}$$

15/ Gramacy & Lee (2012) function:
Parameters: $x \in [0.5, 2.5]$

$$f_{15}(x) = \frac{\sin(10\pi x)}{2x} + (x-1)^4.$$

Chapter 4

Universal Prediction distribution for surrogate models

This chapter is a reproduction of the article Universal Prediction distribution for surrogate models published in SIAM/ASA journal of uncertainty quantification [BSRGT17].

Abstract The use of surrogate models instead of computationally expensive simulation codes is very convenient in engineering. Roughly speaking, there are two kinds of surrogate models: the deterministic and the probabilistic ones. These last are generally based on Gaussian assumptions. The main advantage of probabilistic approach is that it provides a measure of uncertainty associated with the surrogate model in the whole space. This uncertainty is an efficient tool to construct strategies for various problems such as prediction enhancement, optimization or inversion.

In this paper, we propose a universal method to define a measure of uncertainty suitable for any surrogate model either deterministic or probabilistic. It relies on Cross-Validation (CV) sub-models predictions. This empirical distribution may be computed in much more general frames than the Gaussian one. So that it is called the Universal Prediction distribution (*UP distribution*). It allows the definition of many sampling criteria. We give and study adaptive sampling techniques for global refinement and an extension of the so-called Efficient Global Optimization (EGO) algorithm. We also discuss the use of the *UP distribution* for inversion problems. The performances of these new algorithms are studied both on toys models and on an engineering design problem.

4.1 Introduction

Surrogate modeling techniques are widely used and studied in engineering and research. Their main purpose is to replace an expensive-to-evaluate function s by a simple response surface $\hat{s}$ also called surrogate model or meta-model. Notice that s can be a computation-intensive simulation code. These surrogate models are based on a given training set of n observations $z_j = (x_j, y_j)$ where $1 \leq j \leq n$ and $y_j = s(x_j)$. The accuracy of the surrogate model relies, *inter alia*, on the relevance of the training set. The aim of surrogate modeling is generally to estimate some features of the function s using $\hat{s}$. Of course one is looking for the best trade-off between a good accuracy of the feature estimation and the number of calls of s. Consequently, the design of experiments (DOE), that is the sampling of $(x_j)_{1\leq j\leq n}$, is a crucial step and an active research field.

There are two ways to sample: either drawing the training set $(x_j)_{1\leq j\leq n}$ at once or building it sequentially. Among the sequential techniques, some are based on surrogate models. They rely on the feature of s that one wishes to estimate. Popular examples are the EGO [JSW98] and the Stepwise Uncertainty Reduction (SUR) [BGL$^+$12]. These two methods use Gaussian process regression also called kriging model. It is a widely used surrogate modeling technique. Its popularity is mainly due to its statistical nature and properties. Indeed, it is a Bayesian inference technique for functions. In this stochastic frame, it provides an estimate of the prediction error distribution. This distribution is the main tool in Gaussian surrogate sequential designs. For instance, it allows the introduction and the computation of different sampling criteria such as the Expected Improvement (EI) [JSW98] or the Expected Feasibility (EF) [BES$^+$08].

Away from the Gaussian case, many surrogate models are also available and useful. Notice that none of them including the Gaussian process surrogate model are the best in all circumstances [GDT09]. Classical surrogate models are for instance support vector machine [SS04], linear regression [HHB78], moving least squares [LS81]. More recently a mixture of surrogates has been considered in [VHS09, GHSQ07]. Nevertheless, these methods are generally not naturally embeddable in some stochastic frame. Hence, they do not provide any prediction error distribution. To overcome this drawback, several empirical design techniques have been discussed in the literature. These techniques are generally based on resampling methods such as bootstrap, jackknife, or cross-validation. For instance, Gazut et al. [GMDO08] and Jin et al. [JCS02] consider a population of surrogate models constructed by resampling the available data using bootstrap or cross-validation.

Then, they compute the empirical variance of the predictions of these surrogate models. Finally, they sample iteratively the point that maximizes the empirical variance in order to improve the accuracy of the prediction. To perform optimization, Kleijnen et al. [KvBvN12] use a bootstrapped kriging variance instead of the kriging variance to compute the expected improvement. Their algorithm consists in maximizing the expected improvement computed through bootstrapped kriging variance. However, most of these resampling method-based design techniques lead to clustered designs [ASA$^+$13, JCS02].

In this paper, we give a general way to build an empirical prediction distribution allowing sequential design strategies in a very broad frame. Its support is the set of all the predictions obtained by the cross-validation surrogate models. The novelty of our approach is that it provides a prediction uncertainty distribution. This allows a large set of sampling criteria.

The paper is organized as follows. We start by presenting in Section 4.2 the background and notations. In Section 4.3 we introduce the Universal Prediction (UP) empirical distribution. In Sections 4.4 and 4.5, we use and study features estimation and the corresponding sampling schemes built on the UP empirical distribution. Section 4.4 is devoted to the enhancement of the overall model accuracy. Section 4.5 concerns optimization. In Section 4.6, we study a real life industrial case implementing the methodology developed in Section 4.4. Section 4.7 deals with the inversion problem. In Section 4.8, we conclude and discuss the possible extensions of our work. All proofs are postponed to Section 4.9.

4.2 Background and notations

4.2.1 General notation

To begin with, let s denote a real-valued function defined on $\mathbb{X}$, a nonempty compact subset of the Euclidean space $\mathbb{R}^p$ ($p \in \mathbb{N}^\star$). In order to estimate s, we have at hand a sample of size n ($n \geq 2$): $\mathbf{X_n} = \begin{pmatrix} \mathbf{x_1}, & \ldots, & \mathbf{x_n} \end{pmatrix}^\top$ with $\mathbf{x_j} \in \mathbb{X}$, $j \in [\![1; n]\!]$ and $\mathbf{Y_n} = \begin{pmatrix} y_1, & \ldots, & y_n \end{pmatrix}^\top$ where $y_j = s(\mathbf{x_j})$ for $j \in [\![1; n]\!]$. We note $\mathbf{Y_n} = s(\mathbf{X_n})$.

Let $\mathbf{Z_n}$ denote the observations: $\mathbf{Z_n} := \{(\mathbf{x_j}, y_j), j \in [\![1; n]\!]\}$. Using $\mathbf{Z_n}$, we build a surrogate model $\hat{s}_n$ that mimics the behaviour of s. For example, $\hat{s}_n$ can be a second order polynomial regression model. For $i \in \{1 \ldots n\}$, we set $\mathbf{Z}_{n,-i} := \{(\mathbf{x_j}, y_j), j = 1, \ldots, n, j \neq i\}$ and so $\hat{s}_{n,-i}$ is the surrogate model obtained by using only the dataset $\mathbf{Z}_{n,-i}$. We will

call $\hat{s}_n$ the master surrogate model and $(\hat{s}_{n,-i})_{i=1\dots n}$ its sub-models.

Further, let $d(.,.)$ denote a given distance on $\mathbb{R}^p$ (typically the Euclidean one). For $\mathbf{x} \in \mathbb{X}$ and $A \subset \mathbb{X}$, we set: $\underline{d}_A(\mathbf{x}) = \inf\{d(\mathbf{x}, \mathbf{x}') : \mathbf{x}' \in A\}$ and if $A = \{\mathbf{x'_1}, \dots, \mathbf{x'_m}\}$ is finite ($m \in \mathbb{N}^\star$), for $i \in 1, \dots, m$ let A_{-i} denote $\{\mathbf{x'_j}, j = 1 \dots m, j \neq i\}$. Finally, we set $\bar{d}(A) = \max\{\underline{d}_{A_{-i}}(\mathbf{x'_i}) : i = 1, \dots, m\}$, the largest distance of an element of A to its nearest neighbor.

4.2.2 Cross-validation

Training an algorithm and evaluating its statistical performances on the same data yields an optimistic result [AC10]. It is well known that it is easy to over-fit the data by including too many degrees of freedom and so inflate the fit statistics. The idea behind Cross-validation (CV) is to estimate the risk of an algorithm splitting the dataset once or several times. One part of the data (the training set) is used for training and the remaining one (the validation set) is used for estimating the risk of the algorithm. Simple validation or hold-out [DW79] is hence a cross-validation technique. It relies on one splitting of the data. Then one set is used as training set and the second one is used as validation set. Some other CV techniques consist in a repetitive generation of hold-out estimator with different data splitting [Gei75]. One can cite, for instance, the Leave-One-Out Cross-Validation (LOO-CV) and the K-Fold Cross-Validation (KFCV). KFCV consists in dividing the data into k subsets. Each subset plays the role of validation set while the remaining $k-1$ subsets are used together as the training set. LOO-CV method is a particular case of KFCV with $k = n$.

For $i = 1, \dots, n$, the LOO error is $\varepsilon_i = \hat{s}_{n,-i}(\mathbf{x_i}) - y_i$ where the sub-models $\hat{s}_{n,-i}$ are introduced in paragraph 4.2.1. In our study, we are interested in the distribution of the local predictor for all $\mathbf{x} \in \mathbb{X}$ ($\mathbf{x}$ is not necessarily a design point). As explained in the next section, the CV paradigm provides sub-models allowing the definition of a local *uncertainty* measure for the master surrogate model $\hat{s}_n$. This distribution is estimated by using LOO-CV predictions. This is one of the easiest ways to build an uncertainty measure based on resampling.

4.3 Universal Prediction distribution

4.3.1 Overview

As discussed in the previous section, cross-validation is used as a method for estimating the prediction error of a given model. In our case, we introduce a novel use of cross-validation in order to estimate the local uncertainty of a surrogate model prediction. Hence, for a given surrogate model $\hat{s}$ and for any $\mathbf{x} \in \mathbb{X}$, $\hat{s}_{n,-1}(\mathbf{x}), \ldots, \hat{s}_{n,-n}(\mathbf{x})$ define an empirical distribution of $\hat{s}(\mathbf{x})$ at $\mathbf{x}$. In the case of an interpolating surrogate model and a deterministic simulation code s, it is natural to enforce a zero variance at design points. Consequently, when predicting on a design point $\mathbf{x_i}$ we have to neglect the prediction $\hat{s}_{n,-i}$. This can be achieved by introducing weights on the empirical distribution. These weights avoid the pessimistic sub-model predictions that might occur in a region while the global surrogate model fits the data well in that region.

Let $\hat{F}^{(0)}_{n,\mathbf{x}} = \sum_{i=1}^{n} w^0_{i,n}(\mathbf{x})\delta_{\hat{s}_{n,-i}(\mathbf{x})}(dy)$ be the weighted empirical distribution based on the n different predictions of the LOO-CV sub-models $\{\hat{s}_{n,-i}(\mathbf{x})\}_{1\leq i\leq n}$ and weighted by $w^0_{i,n}(\mathbf{x})$ defined in Equation (4.1):

$$w^0_{i,n}(\mathbf{x}) = \begin{cases} \dfrac{1}{n-1} & \text{if } \mathbf{x_i} \neq \arg\min\{d(\mathbf{x}, \mathbf{x_j}), j = 1, \ldots, n\} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

For $i = 1, \ldots, n$, let R_i be the Voronoi cell of the point $\mathbf{x_i}$. The weights can be written as $w^0_{i,n}(\mathbf{x}) = \frac{1-\mathbf{1}_{R_i}(\mathbf{x})}{\sum_{j=1}^{n}(1-\mathbf{1}_{R_j}(\mathbf{x}))}$ where $\mathbf{1}_{R_i}$ is the indicator function on R_i. Such binary weights lead to unsmooth design criteria. In order to avoid this drawback, we smooth the weights. Direct smoothing based on convolution would lead to the computations of Voronoi cells. We prefer to use a simpler technique. Indeed, $w^0_{i,n}(\mathbf{x})$ can be seen as a Nadaraya-Watson weight with the kernel $1 - \mathbf{1}_{R_j}(\mathbf{x}) = \mathbf{1}_{R_i}(\mathbf{x_i}) - \mathbf{1}_{R_j}(\mathbf{x})$. Instead of the unsmooth indicator function $\mathbf{1}_{R_i}$, we use the Gaussian kernel but other smooth kernels could also be used. This leads to the following weights:

$$w_{i,n}(\mathbf{x}) = \frac{1 - e^{-\frac{d(\mathbf{x},\mathbf{x_i})^2}{\rho^2}}}{\sum_{j=1}^{n}\left(1 - e^{-\frac{d(\mathbf{x},\mathbf{x_j})^2}{\rho^2}}\right)} \tag{4.2}$$

Notice that $w_{i,n}(\mathbf{x})$ increases with the distance between the i^{th} design point $\mathbf{x_i}$ and

$\mathbf{x}$. In fact, the least weighted predictions is $\hat{s}_{n,-p_{\text{nn}}(\mathbf{x})}$ where $p_{\text{nn}}(\mathbf{x})$ is the index of the nearest design point to $\mathbf{x}$. In general, the prediction $\hat{s}_{n,-i}$ is locally less reliable in a neighborhood of $\mathbf{x_i}$. The proposed weights determine the local relative confidence level of a given sub-model predictions. The term "relative" means that the confidence level of one sub-model prediction is relative to the remaining sub-models predictions due to the normalization factor in Equation (4.2). The smoothing parameter ρ tunes the amount of uncertainty of $\hat{s}_{n,-i}$ in a neighborhood of $\mathbf{x_i}$. Several options are possible to choose ρ. It can be either related to the distance of a point to its nearest neighbor or common for all the points. We suggest to use $\rho^\star = \bar{d}(\mathbf{X_n})$. Indeed, this is a well suited choice for practical cases.

Definition 8. *The Universal Prediction distribution (UP distribution) is the weighted empirical distribution:*

$$\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\delta_{\hat{s}_{n,-i}(\mathbf{x})}(dy). \tag{4.3}$$

This probability measure is nothing more than the empirical distribution of all the predictions provided by cross-validation sub-models weighted by local smoothed masses.

Definition 9. *For $\mathbf{x} \in \mathbb{X}$ we call $\hat{\sigma}_n^2(\mathbf{x})$ (Equation (4.5)) the local UP variance and $\hat{m}_n(\mathbf{x})$ (Equation (4.4)) the UP expected value.*

$$\hat{m}_n(\mathbf{x}) = \int y\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\hat{s}_{n,-i}(\mathbf{x}) \tag{4.4}$$

$$\hat{\sigma}_n^2(\mathbf{x}) = \int (y - \hat{m}_n(\mathbf{x}))^2 \, \mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})(\hat{s}_{n,-i}(\mathbf{x}) - \hat{m}_n(\mathbf{x}))^2 \tag{4.5}$$

4.3.2 Illustrative example

Let us consider the Viana function defined over $[-3, 3]$

$$f(\mathbf{x}) = \frac{10\cos(2x) + 15 - 5x + x^2}{50} \tag{4.6}$$

Let $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n})$ be the design of experiments such that $\mathbf{X_n} = (x_1 = -2.4, \mathbf{x}_2, = -1.2, x_3 = 0, x_4 = 1.2, x_5 = 1.4, x_6 = 2.4, x_7 = 3)$ and $\mathbf{Y_n} = (y_1, \ldots, y_7)$ their image by f. We used a Gaussian process regression [Mat63, Kri51, Kle09] with constant trend function

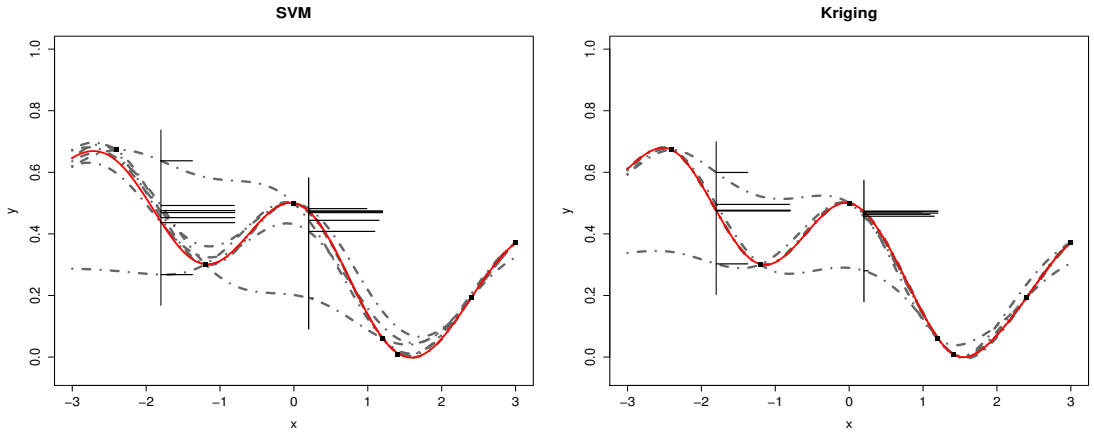


Figure 4.1: Illustration of the *UP distribution* for an SVM surrogate (left) and a kriging surrogate (right). Dashed lines: CV sub-models predictions, solid red line: master model prediction, horizontal bars: local UP distribution at $x_a = -1.8$ and $x_b = 0.2$, black squares: design points.

and Matérn 5/2 covariance function $\hat{s}$ and a SVM regression [SS04]. We display in Figure 4.1 the design points, the cross-validation sub-models predictions $\hat{s}_{n,-i}$, $i = 1, \ldots, 7$ and the master model prediction $\hat{s}_n$ of each surrogate model.

Notice that in the interval $[1, 3]$ (where we have 4 design points) the discrepancy between the master model and the CV sub-models predictions is smaller than in the remaining space. Moreover, we displayed horizontally the *UP distribution* at $x_a = -1.8$ and $x_b = 0.2$ to illustrate the weighting effect. One can notice that:

- At x_a the least weighted predictions are $\hat{s}_{n,-1}(x_a)$ and $\hat{s}_{n,-2}(x_a)$. These predictions do not use the two closest design points to x_a: (x_1, respectively x_2).
- At x_b, $\hat{s}_{n,-3}(x_b)$ is the least weighted prediction.

Furthermore, we display in Figure 4.2 the master model prediction and the region delimited by $\hat{s}_n(\mathbf{x}) + 3\hat{\sigma}_n(\mathbf{x})$ and $\hat{s}_n(\mathbf{x}) - 3\hat{\sigma}_n(\mathbf{x})$. Contrary to the Gaussian case, this region cannot be interpreted as the 99.7% traditional prediction interval. Nevertheless, it can be interpreted as a prediction interval with level greater or equal than 88,8%. Indeed, Chebyshev's inequality states that for any squared integrable random variable X and $k > 0$, $\Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$. In particular, $\Pr(|X - \mu| < 3\sigma) \geq 1 - \frac{1}{3^2} \approx 88.8\%$

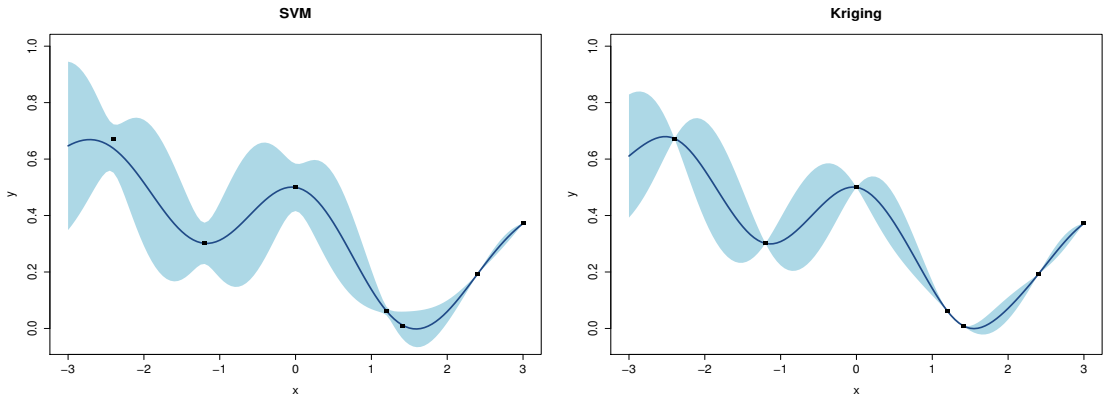


Figure 4.2: Uncertainty quantification based on the *UP distribution* for an SVM surrogate (left) and a kriging surrogate (right). Blue solid line: master model prediction $\hat{s}_n(\mathbf{x})$, light blue area: region delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_n(\mathbf{x})$.

Notice that here the UP standard deviation is null at design points for the interpolating surrogate model. In addition, its local maxima in the interval $[1, 3]$ (where we have more design points density) are smaller than its maxima in the remaining space region.

4.3.3 UP distribution in action

Case of the kriging surrogate model Without loss of generality, let us consider the simple kriging framework. Recall that the conditional mean and variance are given by:

$$\begin{aligned} m(\mathbf{x}) &= \mathbf{k}(\mathbf{x})^\top K_n^{-1} \mathbf{Y_n} \\ \hat{\sigma}^2_{GP}(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k_n}(\mathbf{x})^\top K_n^{-1} \mathbf{k_n}(\mathbf{x}) \end{aligned} \tag{4.7}$$

where $k(\mathbf{x}, \mathbf{x}')$ is a covariance function, $\mathbf{k_n}(\mathbf{x})$ is the vector $(k(\mathbf{x}, \mathbf{x_1}), \ldots, k(\mathbf{x}, \mathbf{x_n}))^\top$ and K_n is the invertible matrix with entries $k_{i,j} = k(\mathbf{x_i}, \mathbf{x_j})$, for $1 \leq i, j \leq n$.

Notice that for an interpolating kriging, both kriging variance and UP-variance vanish at design points. Further, kriging variance does not depend on the output values once the kernel parameters are fixed. However, UP-variance does. It is not everywhere smaller or larger than kriging variance. For instance, consider the toy example in Figure 4.3.

On one hand UP-variance is maximum in the interval $[0.4, 1.3]$. Indeed, if we remove one point in that region we significantly increase the variability of the sub-models predictions. Similarly, UP-variance is minimum in the nearly linear region $[0, 0.4] \cup [1.3, 2]$.

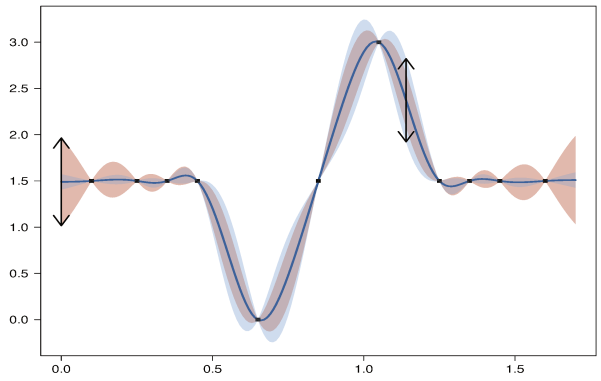


Figure 4.3: UP-variance for Kriging. Black squares: design points, blue line: $\hat{s}_n(\mathbf{x})$, light blue: area delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_n(\mathbf{x})$, light red: area delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_{GP}(\mathbf{x})$.

On the other hand, kriging variance is maximum in that region precisely at $x = 0$. This example highlights how the UP-variance of an interpolating kriging is sensitive to output values.

Surrogate inheritance The UP distribution depends, among others, on the structure of the surrogate model. It inherits some of its properties. For instance, in Figure 4.4 we consider a set of points sampled from a noisy sinusoid and an SVM that filters the noise. In terms of bias-variance dilemma, the SVM surrogate model may have higher bias and lower variance than interpolating surrogates. As a consequence, the LOO sub-models predictions do not vary significantly and the UP-variance is low in the whole design space.

Further, let us consider a degenerated regression problem e.g. 3 aligned points for a first order regression model. The LOO sub-models predictions are all the same. The UP-distribution is then degenerated everywhere.

On the role of the bandwidth parameter ρ Notice that the bandwidth parameter ρ has no effect on the support of the UP distribution. However, it impacts its statistical features. For instance, we notice in Figure 4.5 that ρ does not modify the general shape of the variance but controls its magnitude.

In Section 3.1, we suggested to use $\rho^\star = \bar{d}(\mathbf{X_n})$ which depends on the inter-distance between design points. Such value has the desirable property: $\rho^\star \to 0$ as $n \to +\infty$ iif the design of experiment is dense in the design space. This choice guarantees the convergence of the sequential algorithms presented in Sections 4 and 5.

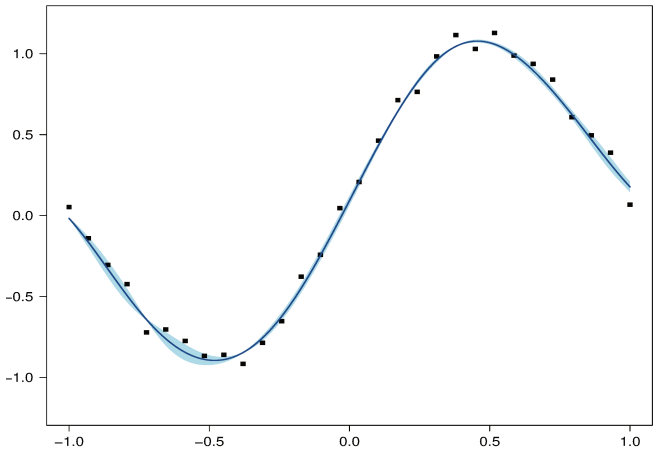


Figure 4.4: Illustration of the UP-variance using a low variance surrogate model (SVM). Black squares: design points, Blue solid line: master model prediction $\hat{s}_n(\mathbf{x})$, shaded area: delimited by $\hat{s}_n(\mathbf{x}) \pm 3\hat{\sigma}_n(\mathbf{x})$.

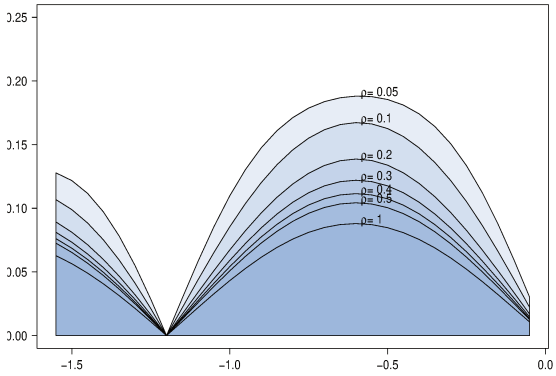


Figure 4.5: ρ effect on the UP-variance locally for the Viana function using kriging.

Computational aspects When the number of design points is large, the computational cost can be a drawback. In fact, the construction time of the sub-models is $O(nT)$ where n is number of the data and T is the construction time of one sub-model. Nevertheless, for some surrogate models, closed form formula are available for the LOO-sub-models predictions. For instance, Chevalier, Ginsbourger, and Emery [CGE14] presented a formula for kriging. Another way to reduce the computational cost is to use parallel computing where each sub-models is computed on a separate thread. Finally, we can replace the use of LOO-CV by the k-fold CV.

4.4 Sequential Refinement

In this section, we use the *UP distribution* to define an adaptive refinement technique called the Universal Prediction-based Surrogate Modeling Adaptive Refinement Technique UP-SMART.

4.4.1 Introduction

The main goal of sequential design is to minimize the number of calls of a computationally expensive function. Gaussian surrogate models [Kle09] are widely used in adaptive design strategies. Indeed, Gaussian modeling gives a Bayesian framework for sequential design. In some cases, other surrogate models might be more accurate although they do not provide a theoretical framework for uncertainty assessment. We propose here a new universal strategy for adaptive sequential design of experiments. The technique is based on the *UP distribution*. So, it can be applied to any type of surrogate model.

In the literature, many strategies have been proposed to design the experiments (for an overview, the interested reader is referred to [GWE03, WS07, SK08]). Some strategies, such as Latin Hypercube Sampling (LHS) [MBC79], maximum entropy design [SW87], and maximin distance designs [JMY90] are called one-shot sampling methods. These methods depend neither on the output values nor on the surrogate model. However, one would naturally expect to design more points in the regions with high nonlinear behavior. This intuition leads to adaptive strategies. A DOE approach is said to be adaptive when information from the experiments (inputs and responses) as well as information from surrogate models are used to select the location of the next point.

By adopting this definition, adaptive DOE methods include for instance surrogate model-based optimization algorithms, probability of failure estimation techniques and sequential refinement techniques. Sequential refinement techniques aim at creating a more accurate surrogate model. For example, Lin et al. [LMA+04] use Multivariate Adaptive Regression Splines (MARS) and kriging models with Sequential Exploratory Experimental Design (SEED) method. It consists in building a surrogate model to predict errors based on the errors on a test set. Goel et al. [GHSQ07] use a set of surrogate models to identify regions of high uncertainty by computing the empirical standard deviation of the predictions of the ensemble members. Our method is based on the predictions of the CV sub-models. In the literature, several cross-validation-based techniques have been discussed. Li and Azarm [LA06] propose to add the design point that maximizes the Accumulative Error (AE). The AE on $\mathbf{x} \in \mathbb{X}$ is computed as the sum of the LOO-CV errors on the design points weighted by influence factors. This method could lead to clustered samples. To avoid this effect, the authors [LAFMD06] propose to add a threshold constraint in the maximization problem. Busby, Farmer, and Iske [BFI07] propose a method based on a grid and CV. It affects the CV prediction errors at a design point to its containing cell in the grid. Then, an entropy approach is performed to add a new design point. More recently, Xu et al. [XLWJ14] suggest the use of a method based on Voronoi cells and CV. Kleijnen and Van Beers [KvB04] propose a method based on the Jackknife's pseudo values predictions variance. Jin, Chen, and Sudjianto [JCS02] present a strategy that maximizes the product between the deviation of CV sub-models predictions with respect to the master model prediction and the distance to the design points. Aute et al. [ASA+13] introduce the Space-Filling Cross-Validation Trade-off (SFCVT) approach. It consists in building a new surrogate model over LOO-CV errors and then add a point that maximizes the new surrogate model prediction under some space-filling constraints. In general cross-validation-based approaches tend to allocate points close to each other resulting in clustering [ASA+13]. This is not desirable for deterministic simulations.

4.4.2 UP-SMART

The idea behind UP-SMART is to sample points where the *UP distribution* variance (Equation (4.5)) is maximal. Most of the CV-based sampling criteria use CV errors. Here, we use the local predictions of the CV sub-models. Moreover, notice that the UP variance is null on design points for interpolating surrogate models. Hence, UP-SMART

does not naturally promote clustering.

However, $\hat{\sigma}_n^2(\mathbf{x})$ can vanish even if $\mathbf{x}$ is not a design points. To overcome this drawback, we add a distance penalization. This leads to the UP-SMART sampling criterion γ_n (Equation (4.8)).

$$\gamma_n(\mathbf{x}) = \hat{\sigma}_n^2(\mathbf{x}) + \delta \underline{d}_{\mathbf{X_n}}(\mathbf{x}) \tag{4.8}$$

where $\delta > 0$ is called exploration parameter. One can set δ as a small percentage of the global variation of the output. UP-SMART is the adaptive refinement algorithm consisting in adding at step n a point $x_{n+1} \in \arg\max_{\mathbf{x}\in\mathbb{X}}(\gamma_n(\mathbf{x}))$.

4.4.3 Performances on a set of test functions

In this subsection, we present the performance of the UP-SMART. We present first the used surrogate-models.

4.4.3.1 Used surrogate models

Kriging Kriging [Mat63] or Gaussian process regression is an interpolation method. Universal Kriging fits the data using a deterministic trend and governed by prior covariances. Let $k(\mathbf{x}, \mathbf{x}')$, be a covariance function on $\mathbb{X} \times \mathbb{X}$, and let $(h_i)_{1\leq i\leq p}$ be the basis functions of the trend. Let us denote $\mathbf{h}(\mathbf{x})$ the vector $(h_1(\mathbf{x}), .., h_p(\mathbf{x}))^\top$ and let H be the matrix with entries $h_{ij} = h_j(\mathbf{x_i}), 1 \leq i, j \leq n$. Furthermore, let $\mathbf{k_n}(\mathbf{x})$ be the vector $(k(\mathbf{x}, \mathbf{x_1}), .., k(\mathbf{x}, \mathbf{x_n}))^\top$ and K_n the matrix with entries $k_{i,j} = k(\mathbf{x_i}, \mathbf{x_j})$, for $1 \leq i, j \leq n$.

Then, the conditional mean of the Gaussian process with covariance $k(\mathbf{x}, \mathbf{x}')$ and its variance are given in Equations ((4.9),(4.10))

$$m_{G_n}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \hat{\beta} + \mathbf{k_n}(\mathbf{x})^\top K_n^{-1}(Y - H^\top \hat{\beta}) \tag{4.9}$$

$$\sigma^2_{GP_n}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k_n}(\mathbf{x})^\top K_n^{-1}\mathbf{k_n}(\mathbf{x})^\top + \mathbf{V}(\mathbf{x})^\top (H^\top K_n^{-1} H)^{-1}\mathbf{V}(\mathbf{x}) \tag{4.10}$$

$$\hat{\beta} = (H^\top K_n^{-1} H)^{-1} H^\top K_n^{-1} Y \text{ and } \mathbf{V}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top + \mathbf{k_n}(\mathbf{x})^\top K_n^{-1} H \tag{4.11}$$

Note that the conditional mean is the prediction of the Gaussian process regression. Further, we used two kriging instances with different sampling schemes in our test bench. Both use constant trend function and a Matérn 5/2 covariance function. The first design is obtained by maximizing the *UP distribution* variance (Equation (4.5)). And the second one is obtained by maximizing the kriging variance $\sigma^2_{GP_n}(\mathbf{x})$.

Genetic aggregation The genetic aggregation response surface is a method that aims at selecting the best response surface for a given design of experiments. It uses several surrogate models. In our examples, we use several kriging and SVM with different settings (kernels, trend functions…) and select the best weighted aggregation

$$\hat{A}_m(\mathbf{x}) = \sum_{l=1}^{m} \omega_l \hat{s}^{(l)}(\mathbf{x}). \tag{4.12}$$

$\hat{s}^{(l)}$ are the surrogate models and the weights ω_l are computed in order to minimize a criterion combining CV errors, surrogate model errors and a smoothness penalty. The use of such response surface, in this test bench, aims at checking the universality of the *UP distribution*: the fact that it can be applied for all types of surrogate models.

4.4.3.2 Test bench

In order to test the performances of the method we launched different refinement processes for the following set of test functions:

- Branin: $f_b(x_1, x_2) = (x_2 - (\frac{5.1}{4\pi^2})x_1^2 + (\frac{5}{\pi})x_1 - 6)^2 + 10(1 - (\frac{1}{8\pi}))\cos(x_1) + 10$.
- Six-hump camel: $f_c(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1x_2 + x_2^2(4x_2^2 - 4)$.
- Hartmann6: $f_h(\mathbf{X} = (x_1, \ldots, x_6)) = -\sum_{i=1}^{4} \alpha_i \exp\Big(-\sum_{j=0}^{6} A_{ij}(x_j - P_{ij})^2\Big)$. A,P and α can be found in [DS78].
- Viana: (Equation (4.6))

For each function we generated by optimal Latin hyper sampling design the number of initial design points n_0, the number of refinement points N_{max}. We also generated a set of N_t test points and their response $Z^{(t)} = (X^{(t)}, Y^{(t)})$. The used values are available in Table 4.1.

We fixed n_0 in order to get non-accurate surrogate models at the first step. Usually, one follows the rule-of-thumb $n_0 = 10 \times d$ proposed in [LSW09]. However, for Branin and Viana functions, this rule leads to a very good initial fit. Therefore, we choose lower values.

- Kriging variance-based refinement process (Equation (4.10)) as refinement criterion.

Table 4.1: Used test functions

Function	dimension d	n_0	N_{max}	N_t
Viana	1	5	7	500
Branin	2	10	10	1600
Camel	2	20	10	1600
Extended Rosenbrock	6→10	60	100	10000
Hartmann6	6→10	100	100	10000

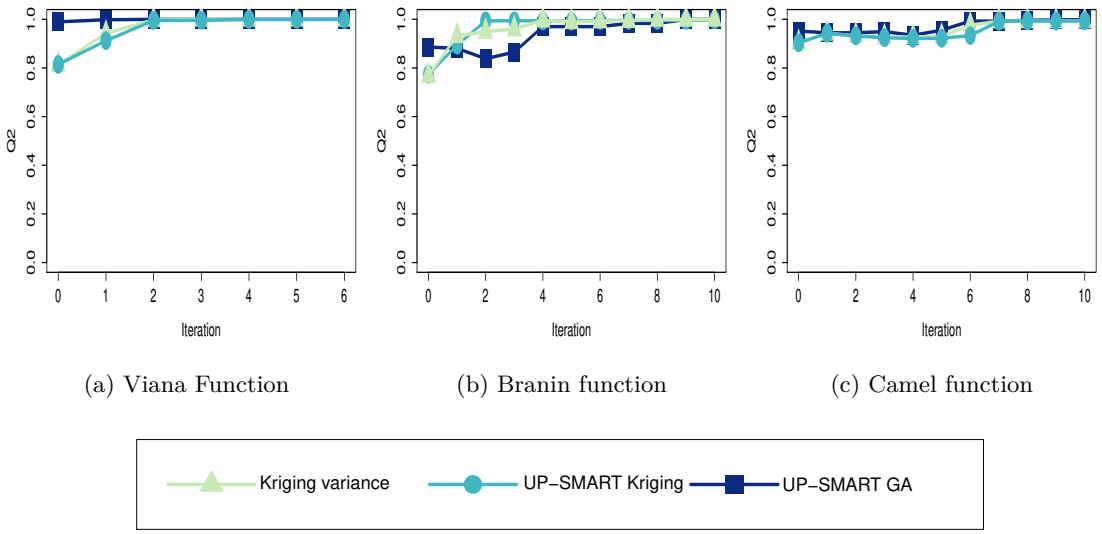
- Kriging using the UP-SMART: UP-variance as refinement criterion (Equation (4.8)).
- Genetic aggregation using the UP-SMART: UP-variance as refinement criterion (Equation (4.8)).

4.4.3.3 Results

For each function, we compute at each iteration the Q squared (Q^2) of the predictions of the test set $Z^{(t)}$ where $Q^2(\hat{s}, Z^{(t)}) = 1 - \frac{\sum_{i=1}^{N_t}(y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)}))^2}{\sum_{i=1}^{N_t}(y_i^{(t)} - \bar{y})^2}$ and $\bar{y} = \frac{1}{N_t}\sum_{i=1}^{N_t} y_i^{(t)}$. We display in Figure 4.6 the performances of the three different techniques described above for Viana (Figure 4.6a), Branin (Figure 4.6b) and Camel (Figure 4.6c) functions measured by Q^2 criterion.

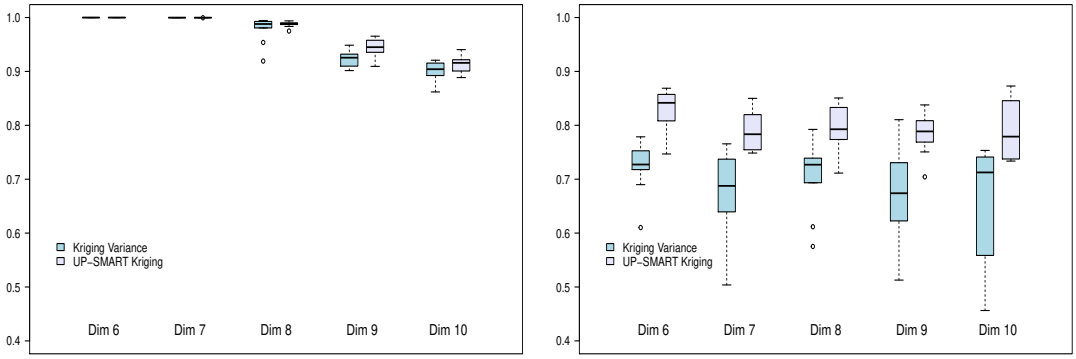
For these tests, the three techniques have comparable performances. The Q^2 converges for all of them. It appears that the UP variance criterion refinement process gives as good a result as the kriging variance criterion. In higher dimensions, we perform for each dimension (from 6 to 10) 10 tests with different initial design of experiments. The sequential algorithm based on kriging variance generates more points on the boundaries. This may be a good strategy when there is significant variability on the boundaries. On one hand consider the extended Rosenbrock function 4.7a. As the function varies significantly on the boundaries, UP-SMART and kriging variance strategies have comparable performances. On the other hand, when the function does not vary much on the boundaries such as Hartman 4.7b, UP-SMART outperforms the kriging variance strategy.

The results show that:



(a) Viana Function (b) Branin function (c) Camel function

Figure 4.6: Performance of three refinement strategies on three test functions measured by the Q^2 criterion on a test set. x axis: number of added refinement points.



(a) Rosenbrock Function (b) Hartmann function

Figure 4.7: Performance of refinement strategies for different dimension on two test functions measured by the Q^2 on a test set UP-SMART with kriging in blue and kriging variance-based technique in violet.

- UP-SMART gives for some problems better global response surface accuracy than the maximization of the kriging variance. This shows the usefulness of the method.
- UP-SMART is a universal method. Here, it has been applied with success to an aggregation of response surfaces. Such usage highlights the universality of the strategy.

4.5 Empirical Efficient Global optimization

In this section, we introduce *UP distribution*-based Efficient Global Optimization (UP-EGO) algorithm. This algorithm is an adaptation of the well known EGO algorithm.

4.5.1 Overview

Surrogate model-based optimization refers to the idea of speeding optimization processes using surrogate models. In this section, we present an adaptation of the well-known efficient global optimization (EGO) algorithm [JSW98]. Our method is based on the weighted empirical distribution *UP distribution*. We show that asymptotically, the points generated by the algorithm are dense around the optimum. For EGO, such result was proved by Vazquez et al. [VB10].

The basic unconstrained surrogate model-based optimization scheme can be summarized as follows [QHS+05]:

- Construct a surrogate model from a set of known data points.
- Define a sampling criterion that reflects a possible improvement.
- Optimize the criterion over the design space.
- Evaluate the true function at the criterion optimum/optima.
- Update the surrogate model using new data points.
- Iterate until convergence.

Several sampling criteria have been proposed to perform optimization. The Expected Improvement (EI) is one of the most popular criteria for surrogate model-based optimization. Sasena, Papalambros, and Goovaerts [SPG00] discussed some sampling criteria such as the threshold-bounded extreme, the regional extreme, the generalized expected improvement and the minimum surprises criterion. Almost all of the criteria

are computed in practice within the frame of Gaussian processes. Consequently, among all possible response surfaces, Gaussian surrogate models are widely used in surrogate model-based optimization. Recently, Viana, Haftka, and Watson [VHW13] performed multiple surrogate-based optimization by importing Gaussian uncertainty estimate.

4.5.2 UP-EGO Algorithm

Here, we use the *UP distribution* to compute an empirical expected improvement. Then, we present an optimization algorithm similar to the original EGO algorithm that can be applied with any type of surrogate models. Without loss of generality, we consider the minimization problem:

$$\underset{\mathbf{x}\in\mathbb{X}}{\text{minimize}} \quad s(\mathbf{x})$$

Let $(y(\mathbf{x}))_{\mathbf{x}\in\mathbb{X}}$ be a Gaussian process model. m_{G_n} and $\sigma^2_{GP_n}$ denote respectively the mean and the variance of the conditional process $y(\mathbf{x}) \mid \mathbf{Z_n}$. Further, let $y^\star_n$ be the minimum value at step n when using observations $\mathbf{Z_n} = (z_1, \ldots, z_n)$ where $z_i = (\mathbf{x_i}, y_i)$. $(y^\star_n = \min_{i=1..n} y_i)$. The EGO algorithm [JSW98] uses the expected improvement EI_n (Equation (4.13)) as sampling criterion:

$$EI_n(\mathbf{x}) = \mathbb{E}[\max(y^\star_n - y(\mathbf{x}), 0) \mid \mathbf{Z_n}] \tag{4.13}$$

The EGO algorithm adds the point that maximizes EI_n. Using some Gaussian computations, Equation (4.13) is equivalent to Equation (4.14).

$$EI_n(\mathbf{x}) = \begin{cases} (y^\star_n - m_{G_n}(\mathbf{x}))\Phi\left(\dfrac{y^\star_n - m_{G_n}(\mathbf{x})}{\sigma_{GP_n}(\mathbf{x})}\right) \\ \qquad + \ \sigma_{GP_n}(\mathbf{x})\phi\left(\dfrac{y^\star_n - m_{G_n}(\mathbf{x})}{\sigma_{GP_n}(\mathbf{x})}\right) & \text{if } \sigma_{GP_n}(\mathbf{x}) \neq 0, \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

We introduce a similar criterion based on the *UP distribution*. With the notations of Sections 4.2 and 4.3, EEI_n (Equation (4.15)) is called the empirical expected improvement.

$$\begin{aligned} EEI_n(\mathbf{x}) &= \int \max(y^\star_n - y, 0)\mu_{(n,\mathbf{x})}(dy) \\ &= \sum_{i=1} w_{i,n}(\mathbf{x}) \max(y^\star_n - \hat{s}_{n,-i}(\mathbf{x}), 0) \end{aligned} \tag{4.15}$$

We can remark that $EEI_n(\mathbf{x})$ can vanish even if $\mathbf{x}$ is not a design point. This is one of the limitations of the empirical *UP distribution*. To overcome this drawback, we suggest the use of the Universal Prediction Expected Improvement (UP-EI) κ_n (Equation (4.16))

$$\kappa_n(\mathbf{x}) = EEI_n(\mathbf{x}) + \xi_n(\mathbf{x}) \tag{4.16}$$

where $\xi_n(\mathbf{x})$ is a distance penalization. We use $\xi_n(\mathbf{x}) = \delta \underline{d}_{\mathbf{X_n}}(\mathbf{x})$ where $\delta > 0$ is called the exploration parameter. One can set δ as a small percentage of the global variation of the output for less exploration. Greater value of δ means more exploration. δ fixes the wished trade-off between exploration and local search.

Furthermore, notice that κ_n has the desirable property also verified by the usual EI:

Proposition 2. $\forall n > 1, \forall \mathbf{Z_n} = (\mathbf{X_n} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^\top, \mathbf{Y_n} = s(\mathbf{X_n}))$, *if the used model interpolates the data then* $\kappa_n(\mathbf{x_i}) = 0$, *for* $i = 1, \ldots, n$

The *UP distribution*-based Efficient Global Optimization (UP-EGO) (Algorithm 3) consists in sampling at each iteration the point that maximize κ_n. The point is then added to the set of observations and the surrogate model is updated.

Algorithm 3: UP-based Efficient Global Optimization

UP-EGO($\hat{s}$)

Inputs: $\mathbf{Z}_{n_0} = (X_{n_0}, Y_{n_0})$, $n_0 \in \mathbb{N} \setminus \{0, 1\}$ and a deterministic function s

(1) $m := n_0$, $\mathbf{S}_m := X_{n_0}$, $Y_m := Y_{n_0}$

(2) Compute the surrogate model $\hat{s}_{\mathbf{Z}_m}$

(3) *Stop_conditions* := False

(4)**While** Stop_conditions are not satisfied

(4.1) Select $\mathbf{x_{m+1}} \in arg\max_{\mathbb{X}}(\kappa_m(\mathbf{x}))$

(4.2) Evaluate $y_{m+1} := s(\mathbf{x_{m+1}})$

(4.3) $\mathbf{S}_{m+1} := \mathbf{S}_m \cup \{\mathbf{x_{m+1}}\}$, $Y_{m+1} := Y_m \cup \{y_{m+1}\}$

(4.4) $\mathbf{Z}_{m+1} := (\mathbf{S}_{m+1}, Y_{m+1})$,$m := m + 1$

(4.5) Update the surrogate model

(4.6) Check *Stop_conditions*

end loop

Outputs: $\mathbf{Z}_m := (\mathbf{S}_m, Y_m)$, surrogate model $\hat{s}_{\mathbf{Z}_m}$

4.5.3 UP-EGO convergence

We first recall the context. $\mathbb{X}$ is a nonempty compact subset of the Euclidean space $\mathbb{R}^p$ where $p \in \mathbb{N}^\star$. s is an expensive-to-evaluate function. The weights of the *UP distribution* are computed as in Equation (4.2) with $\rho > 0$ a fixed real parameter. Moreover, we consider the asymptotic behaviour of the algorithm so that, here, the number of iterations goes to infinity. Let $\mathbf{x}^\star \in \arg\min\{s(\mathbf{x}), \mathbf{x} \in \mathbb{X}\}$ and $\hat{s}$ be a continuous interpolating surrogate model bounded on $\mathbb{X}$. Let $\mathbf{Z}_{n_0} = (X_{n_0} = (\mathbf{x_1}, \ldots, \mathbf{x_{n_0}})^\top, Y_{n_0})$ be the initial data. For all $k > n_0$, $\mathbf{x_k}$ denotes the point generated by the UP-EGO algorithm at step $k - n_0$. Let $\mathbf{S}_m$ denote the set $\{\mathbf{x_i}, i \leq m\}$ and $S = \{\mathbf{x_i}, i > 0\}$. Finally, $\forall m > n_0$ we note κ_m the UP-EI of $\hat{s}_{\mathbf{Z}_m}$. We are going to prove that $\mathbf{x}^\star$ is adherent to the sequence S generated by the UP-EGO($\hat{s}$) algorithm.

Lemma 1. $\exists \theta > 0$, $\forall m \geq n_0$, $\forall \mathbf{x} \in \mathbb{X}$, $\forall i \in 1, \ldots, m$, $\forall n > m$, $w_{i,n}(\mathbf{x}) \leq \theta d(\mathbf{x}, \mathbf{x_i})^2$.

Definition 10. *A surrogate model $\hat{s}$ is called an interpolating surrogate model if for all $n \in \mathbb{N}^\star$ and for all $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n}) \in \mathbb{X}^n \times \mathbb{R}^n$, $\hat{s}_{\mathbf{Z_n}}(\mathbf{x}) = s(\mathbf{x})$ if $\mathbf{x} \in \mathbf{X_n}$.*

Definition 11. *A surrogate model $\hat{s}$ is called bounded on $\mathbb{X}$ if for all s a continuous function on $\mathbb{X}$, $\exists L, U$, such that for all $n > 1$ and for all $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n} = s(\mathbf{X_n})) \in \mathbb{X}^n \times \mathbb{R}^n$, $\forall \mathbf{x} \in \mathbb{X}$, $L \leq \hat{s}_{\mathbf{Z_n}}(\mathbf{x}) \leq U$*

Definition 12. *A surrogate model $\hat{s}$ is called continuous if $\forall n_0 > 1$ $\forall \mathbf{x} \in \mathbb{X}$ $\forall \varepsilon > 0$, $\exists \delta > 0$, $\forall n \geq n_0$, $\forall \mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n}) \in \mathbb{X}^n \times \mathbb{R}^n$, $\forall \mathbf{x}' \in \mathbb{X}$, $d(\mathbf{x}, \mathbf{x}') < \delta \implies |\hat{s}_{\mathbf{Z_n}}(\mathbf{x}) - \hat{s}_{\mathbf{Z_n}}(\mathbf{x}')| < \varepsilon$*

Theorem 2. *Let s be a real function defined on $\mathbb{X}$ and let $\mathbf{x}^\star \in \arg\min\{s(\mathbf{x}), \mathbf{x} \in \mathbb{X}\}$. If $\hat{s}$ is an interpolating continuous surrogate model bounded on $\mathbb{X}$, then $\mathbf{x}^\star$ is adherent to the sequence of points S generated by UP-EGO($\hat{s}$).*

The proofs (Section 4.9) show that the exploration parameter is important for this theoretical result. In our implementation, we scale the input spaces to be the hypercube $\big[-1, 1]$ and we set δ to 0.005% of the output variation. Hence, the exploratory effect only slightly impacts the UP-EI criterion in practical cases.

4.5.4 Numerical examples

Let us consider the set of test functions (Table 4.2).

Table 4.2: Optimization test functions

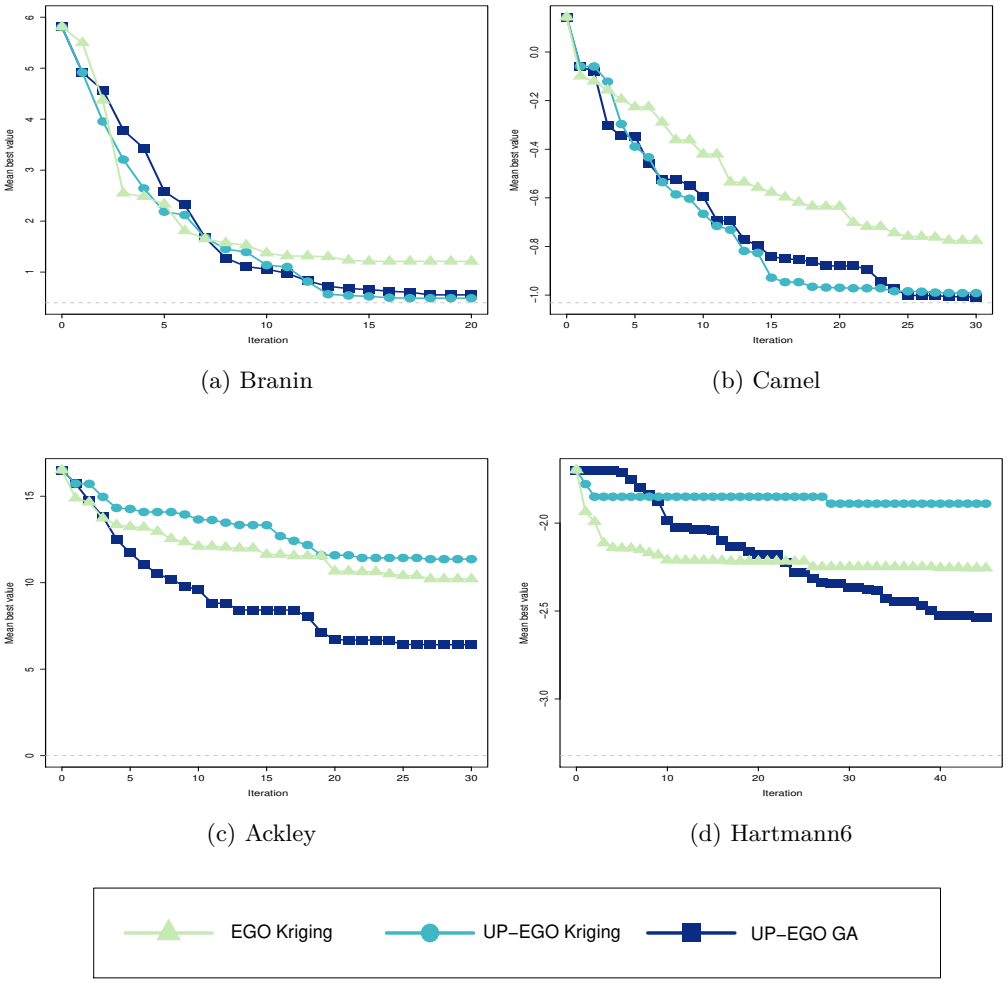
function $f^{(i)}$	Dimension $d^{(i)}$	Number of initial points $n_0^{(i)}$	Number of iterations $N_{max}^{(i)}$
Branin	2	5	40
Ackley	2	10	30
Six-hump Camel	2	10	30
Hartmann6	6	20	40

We launched the optimization process for these functions with three different optimization algorithms:

- EGO [JSW98]: Implementation of the R package DiceOptim [RGD12] using the default parameters.
- UP-EGO algorithm applied to a universal kriging surrogate model $\hat{s_k}$ that uses Matérn 5/2 covariance function and a constant trend function. We denote this algorithm UP-EGO($\hat{s_k}$)
- UP-EGO algorithm applied to the genetic aggregation $\hat{s_a}$. It is then denoted UP-EGO($\hat{s_a}$).

For each function $f^{(i)}$, we launched each optimization process for $N_{max}^{(i)}$ iterations starting with $N_{seed} = 20$ different initial design of experiments of size $n_0^{(i)}$ generated by an optimal space-filling sampling. The results are given using boxplots in Appendix 4.11. We also display the mean best value evolution in Figure 4.8.

The results shows that the UP-EGO algorithms give better results than the EGO algorithm for Branin and Camel functions. These cases illustrate the efficiency of the method. Moreover, for Ackley and Harmtann6 functions the best results are given by UP-EGO using the genetic aggregation. Even if this is related to the nature of the surrogate model, it underlines the efficient contribution of the universality of UP-EGO. Further, let us focus on the boxplots of the last iterations of Figures 4.11 and 4.14 (Appendix 4.11). It is important to notice that UP-EGO results for Branin function depend slightly on the



(a) Branin (b) Camel

(c) Ackley (d) Hartmann6

Figure 4.8: Comparison of 3 surrogate-based optimization strategies. Mean over N_{seed} of the best value as a function of the number of iterations.

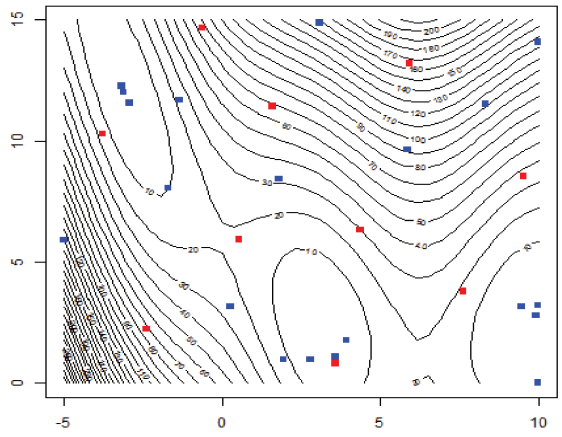


Figure 4.9: Example of sequence generated by the UP-EGO(kriging) algorithm on Branin function. Initial design points are in red, added points are in blue.

initial design points. On the other hand, let us focus on the Hartmann function case. The results of UP-EGO using the genetic aggregation depend on the initial design points. In fact, more optimization iterations are required for a full convergence. However, compared to the EGO algorithm, UP-EGO algorithm has good performances for both cases:

- Full convergence
- Limited-budget optimization.

Otherwise, the Branin function has multiple solutions. We are interested in checking whether the UP-EGO algorithm would focus on one local optimum or on the three possible regions. We present in Figure 4.9 the spatial distribution of the generated points by the UP-EGO (kriging) algorithm for the Branin function. We can notice that UP-EGO generated points around the three local minima.

4.6 Fluid Simulation Application: Mixing Tank

The problem addressed here concerns a static mixer where hot and cold fluid enter at variable velocities. The objective of this analysis is generally to find inlet velocities that minimize pressure loss from the cold inlet to the outlet and minimize the temperature spread at the outlet. In our study, we are interested in a better exploration of the design

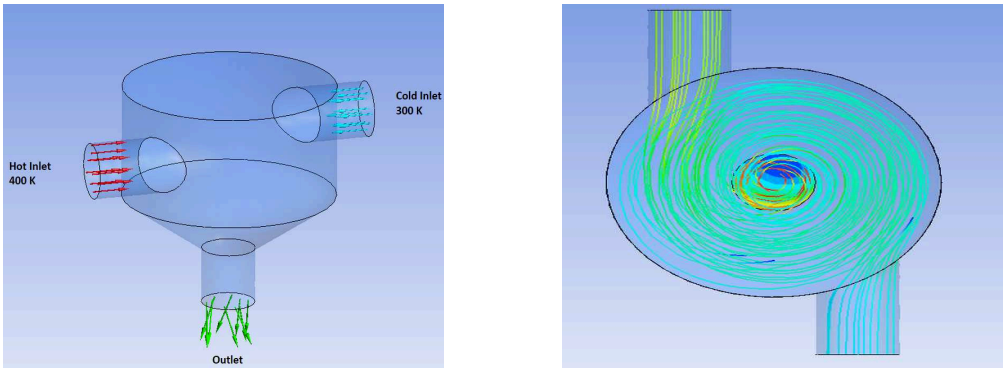


Figure 4.10: Mixing tank

using an accurate cheap-to-evaluate surrogate model.

The simulations are computed within ANSYS Workbench environment and we used DesignXplorer to perform surrogate-modeling. We started the study using 9 design points generated by a central composite design. We produced also a set of $N_t = 80$ test points $Z_t = (X_t = (x_1^{(t)}), \ldots, \mathbf{x}_{N_t}^{(t)}), Y_t = (y_1^{(t)}), \ldots, y_{N_t}^{(t)}))$. We launched UP-SMART applied to the genetic aggregation response surface (GARS) in order to generate 10 suitable design points and a kriging-based refinement strategy. The genetic aggregation response surface (GARS) developed by DesignXplorer creates a mixture of surrogate models including support vector machine regression, Gaussian process regression, moving Least Squares and polynomial regression. We computed the root mean square error (Equation (4.17)), the relative root mean square error (Equation (4.18)) and the relative average absolute error (Equation (4.19)) before and after launching the refinement processes.

$$RMSE_{Z^{(t)}}(\hat{s}) = \frac{1}{N^t}\sum_{i=1}^{N_t}(y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)}))^2 \tag{4.17}$$

$$RRMSE_{Z^{(t)}}(\hat{s}) = \frac{1}{N^t}\sum_{i=1}^{N_t}\left(\frac{y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)})}{y_i^{(t)}}\right)^2 \tag{4.18}$$

$$RAAE_{Z^{(t)}}(\hat{s}) = \frac{1}{N^t}\sum_{i=1}^{N_t}\frac{\mid y_i^{(t)} - \hat{s}(\mathbf{x_i}^{(t)}) \mid}{\sigma_Y} \tag{4.19}$$

Table 4.3: Quality measures of different response surfaces of static mixer simulations

Surrogate model	RRMSE	RMSE	RAAE
GARS Initial	0.16	0.10	0.50
GARS Final	0.10	0.07	0.31
Kriging Initial	0.16	0.11	0.48
Kriging Final	0.16	0.11	0.50

We give in Table 4.3 the obtained quality measures for the temperature spread output. In fact, the pressure loss is nearly linear and every method gives a good approximation.

The results show that UP-SMART gives a better approximation. Here, it is used with a genetic aggregation of several response surface. Even if the good quality may be due to the response surface itself, it highlights the fact that UP-SMART made the use of such surrogate model-based refinement strategy possible.

4.7 Empirical Inversion

4.7.1 Empirical inversion criteria adaptation

Inversion approaches consist in the estimation of contour lines, excursion sets or probability of failure. These techniques are specially used in constrained optimization and reliability analysis.

Several iterative sampling strategies have been proposed to handle these problems. The empirical distribution $\mu_{n,\mathbf{x}}$ can be used for inversion problems. In fact, we can compute most of the well-known criteria such as the Bichon's criterion [BES+08] or the Ranjan's criterion [RBM08] using the *UP distribution*. In this section, we discuss some of these criteria: the targeted mean square error $TMSE$ [PGR+10], Bichon [BES+08] and the Ranjan criteria [RBM08]. The reader can refer to Chevalier, Picheny, and Ginsbourger [CPG14] for an overview.

Let us consider the contour line estimation problem : let T be a fixed threshold. We are interested in enhancing the surrogate model accuracy in $\{\mathbf{x} \in \mathbb{X}, s(\mathbf{x}) = T\}$ and in its neighborhood.

Targeted MSE (TMSE) The Targeted Mean Square Error (TMSE) [PGR+10] aims at decreasing the mean square error where the kriging prediction is close to T.

It is the probability that the response lies inside the interval $\left[T-\varepsilon, T+\varepsilon\right]$ where the parameter $\varepsilon > 0$ tunes the size of the window around the threshold T. High values make the criterion more exploratory while low values concentrate the evaluation around the contour line.

We can compute an estimation of the value of this criterion using the *UP distribution* (Equation (4.20)).

$$\begin{aligned} TMSE_{T,n}(\mathbf{x}) &= \hat{\sigma}_n^2(\mathbf{x}) \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) 1_{\left[T-\varepsilon, T+\varepsilon\right]}\left(\hat{s}_{n,-i}(\mathbf{x})\right) \\ &= \hat{\sigma}_n^2(\mathbf{x}) \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) 1_{\left[-\varepsilon, \varepsilon\right]}\left(\hat{s}_{n,-i}(\mathbf{x}) - T\right) \end{aligned} \tag{4.20}$$

Notice that the last criterion takes into account neither the variability of the predictions at $\mathbf{x}$ nor the magnitude of the distance between the predictions and T.

Bichon criterion The expected feasibility defined in [BES+08] aims at indicating how well the true value of the response is expected to be close to the threshold T.

The bounds are defined by $\varepsilon_{\mathbf{x}}$ which is proportional to the kriging standard deviation $\hat{\sigma}(\mathbf{x})$. Bichon proposes using $\varepsilon_{\mathbf{x}} = 2\hat{\sigma}(\mathbf{x})$ [BES+08].

This criterion can be extended to the case of the *UP distribution*. We define in Equation (4.21) EF_n the empirical Bichon's criterion where $\varepsilon_{\mathbf{x}}$ is proportional to the empirical standard deviation $\hat{\sigma}_n^2(\mathbf{x})$ (Equation (4.5)).

$$EF_n(\mathbf{x}) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})(\varepsilon_{\mathbf{x}} - |T - \hat{s}_{n,-i}(\mathbf{x})|) 1_{\left[-\varepsilon_{\mathbf{x}}, \varepsilon_{\mathbf{x}}\right]}(\hat{s}_{n,-i}(\mathbf{x}) - T) \tag{4.21}$$

Ranjan criterion Ranjan et al. [RBM08] proposed a criterion that quantifies the improvement $I_{Ranjan}(\mathbf{x})$ defined in Equation (4.22)

$$I_{Ranjan}(\mathbf{x}) = \left(\varepsilon_{\mathbf{x}}^2 - (y(\mathbf{x}) - T)^2\right) 1_{[-\varepsilon_{\mathbf{x}}, \varepsilon_{\mathbf{x}}]}(y(\mathbf{x}) - T) \tag{4.22}$$

where $\varepsilon_{\mathbf{x}} = \alpha\hat{\sigma}(\mathbf{x})$, and $\alpha > 0$. $\varepsilon_{\mathbf{x}}$ defines the size of the neighborhood around the contour T.

It is possible to compute the *UP distribution*-based Ranjan's criterion (Equation (4.23)). Note that we set $\varepsilon_{\mathbf{x}} = \alpha\hat{\sigma}_n^2(\mathbf{x})$.

$$\mathbb{E}\Big[I_{Ranjan}(\mathbf{x})\Big] = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\Big(\varepsilon_{\mathbf{x}}^2 - (\hat{s}_{n,-i}(\mathbf{x}) - T)^2\Big)1_{[-\varepsilon_{\mathbf{x}},\varepsilon_{\mathbf{x}}]}(\hat{s}_{n,-i}(\mathbf{x}) - T) \tag{4.23}$$

4.7.2 Discussion

The use of the pointwise criteria (Equations (4.20), (4.21), (4.23)) might face problems when the region of interest is relatively small to the prediction jumps. In fact, as the cumulative distribution function of the *UP distribution* is a step function, the probability of the prediction being inside an interval can vanish even if it is around the mean value. For instance $\mu_{n,\mathbf{x}}\big(y(\mathbf{x}) \in [T-\varepsilon, T+\varepsilon]\big)$ can be zero. This is one of the drawbacks of the empirical distribution. Some regularization techniques are possible to overcome this problem. For instance, the technique that consists in defining the region of interest by a Gaussian density $\mathcal{N}(0, \sigma_\varepsilon^2)$ [PGR+10]. Let g_ε be this Gaussian probability distribution function.

The new $TMSE$ denoted $TMSE_{T,n}^{(2)}(\mathbf{x})$ criterion is then as in Equation (4.24).

$$TMSE_{T,n}^{(2)}(\mathbf{x}) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) g_\varepsilon\big(\hat{s}_{n,-i}(\mathbf{x}) - T\big) \tag{4.24}$$

The use of the Gaussian density to define the targeted region seems more relevant when using the UP local variance. Similarly, we can apply the same method to the Ranjan's and Bichon's criteria.

4.8 Conclusion

To perform surrogate model-based sequential sampling, several relevant techniques require to quantify the prediction uncertainty associated to the model. Gaussian process regression provides directly this uncertainty quantification. This is the reason why Gaussian modeling is quite popular in sequential sampling. In this work, we defined a universal approach for uncertainty quantification that could be applied for any surrogate model. It is based on a weighted empirical probability measure supported by cross-validation

sub-models predictions.

Hence, one could use this distribution to compute most of the classical sequential sampling criteria. As examples, we discussed sampling strategies for refinement, optimization and inversion. Further, we showed that, under some assumptions, the optimum is adherent to the sequence of points generated by the optimization algorithm UP-EGO. Moreover, the optimization and the refinement algorithms were successfully implemented and tested both on single and multiple surrogate models. We also discussed the adaptation of some inversion criteria. The main drawback of *UP distribution* is that it is supported by a finite number of points. To avoid this, we propose to regularize this probability measure. In a future work, we will study and implement such regularization scheme and extend its applications to other application such as: multi-objective constrained optimization and reliability based design optimization.

4.9 Proofs

We present in this section the proofs of Proposition 2, Lemma 1 and Theorem 2. Here, we use the notations of Section 4.5.3.

Proof of Proposition 2. Let $n > 1$, $\mathbf{Z_n} = (\mathbf{X_n} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^\top, \mathbf{Y_n} = s(\mathbf{X_n}))$, and $\hat{s}$ a model that interpolates the data i.e $\forall i \in 1, \ldots, n$, $\hat{s}_{\mathbf{Z_n}}(\mathbf{x_i}) = s(\mathbf{x_i}) = y_i$.

First, we have $\xi_n(\mathbf{x_i}) = \delta \underline{d}_{\mathbf{X_n}}(\mathbf{x_i})$. Since $\mathbf{x_i} \in \mathbf{X_n}$ then $\xi_n(\mathbf{x_i}) = 0$. Further, $EEI_n(\mathbf{x_i}) = w_{i,n}(\mathbf{x_i}) \max(y_n^\star - \hat{s}_{n,-i}(\mathbf{x_i}), 0) + \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{j,n}(\mathbf{x_i}) \max(y_n^\star - y_i, 0)$. Notice that

- $w_{i,n}(\mathbf{x_i}) = 0$
- $\max(y_n^\star - y_i, 0) = 0$

Then $EEI_n(\mathbf{x_i}) = 0$. Finally, $\kappa_n(\mathbf{x_i}) = EEI_n(\mathbf{x_i}) + \xi_n(\mathbf{x_i}) = 0$. □

Proof of Lemma 1. Let us note :

- $\phi_\rho(\mathbf{x}, \mathbf{x}') = 1 - e^{-\frac{d((\mathbf{x},\mathbf{x}'))^2}{\rho^2}}$.
- $w_{i,n}(\mathbf{x}) = \frac{\phi_\rho(\mathbf{x},\mathbf{x_i})}{\sum_{k=1}^{n} \phi_\rho(\mathbf{x},\mathbf{x_k})}$.

Convex inequality gives $\forall a \in \mathbb{R},\ 1 - e^{-a} < a$ then $\phi_\rho(\mathbf{x}, \mathbf{x_k}) \leq \frac{d((\mathbf{x},\mathbf{x_k}))^2}{\rho^2}$. Further, let $\mathbf{x_{k_1}}, \mathbf{x_{k_2}}$ be two different design points of X_{n_0}, $\forall \mathbf{x} \in \mathbb{X}$, $\max\limits_{i\in\{1,2\}}\{d(\mathbf{x}, \mathbf{x_{k_i}})\} \geq \frac{d(\mathbf{x_{k_1}},\mathbf{x_{k_2}})}{2}$ otherwise the triangular inequality would be violated. Consequently,

$$\forall n > n_0,\ \sum_{k=1}^{n} \phi_\rho(\mathbf{x}, \mathbf{x_k}) \geq \phi_\rho(\mathbf{x}, \mathbf{x_{k_1}}) + \phi_\rho(\mathbf{x}, \mathbf{x_{k_2}}) \geq \phi_{2\rho}(\mathbf{x_{k_1}}, \mathbf{x_{k_2}}) > 0$$

$$\forall n > n_0,\ \forall \mathbf{x} \in \mathbb{X} : w_{i,n}(\mathbf{x}) = \frac{\phi_{i,n}(\mathbf{x})}{\sum\limits_{k=1}^{n} \phi_{k,n}(\mathbf{x})} \leq \frac{\phi_{i,n}(\mathbf{x})}{\phi_{2\rho}(\mathbf{x_{k_1}}, \mathbf{x_{k_2}})} \leq \frac{d((\mathbf{x}, \mathbf{x_i}))^2}{\rho^2 \phi_{2\rho}(\mathbf{x_{k_1}}, \mathbf{x_{k_2}})}$$

Considering $\theta = \frac{1}{\rho^2 \phi_{2\rho}(\mathbf{x_{k_1}},\mathbf{x_{k_2}})}$ ends the proof. □

Proof of Theorem 2. $\mathbb{X}$ is compact so S has a convergent sub-sequence in $\mathbb{X}^{\mathbb{N}}$ (Bolzano-Weierstrass theorem). Let $(x_{\psi(n)})$ denote that sub-sequence and $\mathbf{x}_\infty \in \mathbb{X}$ its limit. We can assume by considering a sub-sequence of ψ and using the continuity of the surrogate model $\hat{s}$ that:

- $d(\mathbf{x}_\infty, \mathbf{x}_{\psi(n)}) \leq \frac{1}{n}$ for all $n > 0$
- $\exists \nu_n \geq d(\mathbf{x}_\infty, \mathbf{x}_{\psi(n)})$ such that $\forall \mathbf{x}' \in \mathbb{X}$, $d(\mathbf{x}', \mathbf{x}_\infty) \leq \nu_n \implies |\hat{s}_{m,-i}(\mathbf{x}_\infty) - \hat{s}_{m,-i}(\mathbf{x}')| \leq \frac{1}{n}$, $\forall i \in 1, \ldots, m$, where $m > n_0$.

For all $k > 1$, we note $v_k = \psi(k+1) - 1$, the step at which UP-EGO algorithm selects the point $\mathbf{x}_{\psi(k+1)}$. So, $\kappa_{v_k}(\mathbf{x}_{\psi(k+1)}) = \max\limits_{\mathbf{x}\in\mathbb{X}}\{\kappa_{v_k}(\mathbf{x})\}$.

Notice first that for all $n > 0$, $\mathbf{x}_{\psi(n)}, \mathbf{x}_{\psi(n+1)} \in \mathcal{B}(\mathbf{x}_\infty, \frac{1}{n})$ where $\mathcal{B}(\mathbf{x}_\infty, \frac{1}{n})$ is the closed ball of center $\mathbf{x}_\infty$ and radius $\frac{1}{n}$. So:

$$\xi_{v_n}(\mathbf{x}_{\psi(n+1)}) = \delta \underline{d}_{X_{v_n}}(\mathbf{x}_{\psi(n+1)}) \leq \delta d(\mathbf{x}_{\psi(\mathbf{n})}, \mathbf{x}_{\psi(n+1)}) \leq \frac{2\delta}{n} \tag{i}$$

According to Lemma 1, $w_{\psi(n),v_n} \leq \theta \left(d(\mathbf{x}_{\psi(n+1)}, \mathbf{x}_{\psi(n)})\right)^2$ so $w_{\psi(n),v_n} \leq \frac{4\theta}{n^2}$. Consequently:

$$w_{\psi(n),v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-\psi(n)}(\mathbf{x}_{\psi(n+1)}), 0) \leq \frac{4\theta(U-L)}{n^2} \tag{ii}$$

Further, $\forall i \in 1, \ldots, v_n$, $i \neq \psi(n)$, $\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) = y_{\psi(n)}$ since the surrogate model is an interpolating one. hence $\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) \geq y^\star_{v_n}$ and so $\max(y^\star_{v_n} - \hat{s}_{v_n,-i}, 0) \leq \max(\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) -$

$\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)}),0) \leq \left|\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)})\right|$. Triangular inequality gives: $\max(y^\star_{v_n} - \hat{s}_{v_n,-i},0) \leq \left|\hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n)}) - \hat{s}_{v_n,-i}(\mathbf{x}_\infty)\right| + \left|\hat{s}_{v_n,-i}(\mathbf{x}_\infty) - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)})\right|$ and finally:

$$\max(y^\star_{v_n} - \hat{s}_{v_n,-i},0) \leq \frac{2}{n} \tag{iii}$$

We have:

$$\begin{aligned}
\left|\kappa_{v_n}(\mathbf{x}_{\psi(n+1)})\right| &= \xi_{v_n}(\mathbf{x}_{\psi(n+1)}) + \sum_{i=1}^{v_n} w_{i,v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)}),0) \\
&= \xi_{v_n}(\mathbf{x}_{\psi(n+1)}) + w_{\psi(n),v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-\psi(n)}(\mathbf{x}_{\psi(n+1)}),0) \\
&+ \sum_{\substack{i=1 \\ i\neq\psi(n)}}^{v_n} w_{i,v_n}(\mathbf{x}_{\psi(n+1)}) \max(y^\star_{v_n} - \hat{s}_{v_n,-i}(\mathbf{x}_{\psi(n+1)}),0) \\
&\leq \frac{2\delta}{n} + \frac{4\theta(U-L)}{n^2} + \frac{2}{n}
\end{aligned}$$

Considering (i),(ii) and (iii) :

$$\left|\kappa_{v_n}(\mathbf{x}_{\psi(n+1)})\right| \leq \frac{2\delta}{n} + \frac{4\theta(U-L)}{n^2} + \frac{2}{n}$$

Notice that:
$\kappa_{v_n}(\mathbf{x}_{\psi(n+1)}) = \max_{\mathbf{x}\in\mathbb{X}}\{\kappa_{v_n}(\mathbf{x})\}$ and $\delta \underline{d}_{\mathbf{S}_{v_n}}(\mathbf{x}^\star) = \xi_{v_n}(\mathbf{x}^\star) \leq \kappa_{v_n}(\mathbf{x}^\star) \leq \kappa_{v_n}(\mathbf{x}_{\psi(n)})$. Since $\lim_{n\to\infty} \left|\kappa_{v_n}(\mathbf{x}_{\psi(n+1)})\right| = 0$ so $\lim_{n\to\infty} \underline{d}_{\mathbf{S}_{v_n}}(\mathbf{x}^\star) \to 0$. □

4.10 Software and acknowledgments

An R package containing UP distribution tools is available on `https://github.com/malekbs/UP`. We gratefully acknowledge the French National Association for Research and Technology (ANRT, CIFRE grant number 2015/1349). We warmly thank Andom Iyassu for his help on editing. We also thank two anonymous reviewers and the associate editor for their valuable comments. Part of this research was presented at the Chair in Applied Mathematics OQUAIDO, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, Ecole Centrale de Lyon, Mines Saint-Etienne, University of Grenoble, University of Nice, University of Toulouse) around advanced methods for Computer Experiments. We thank the participants for their feedback.

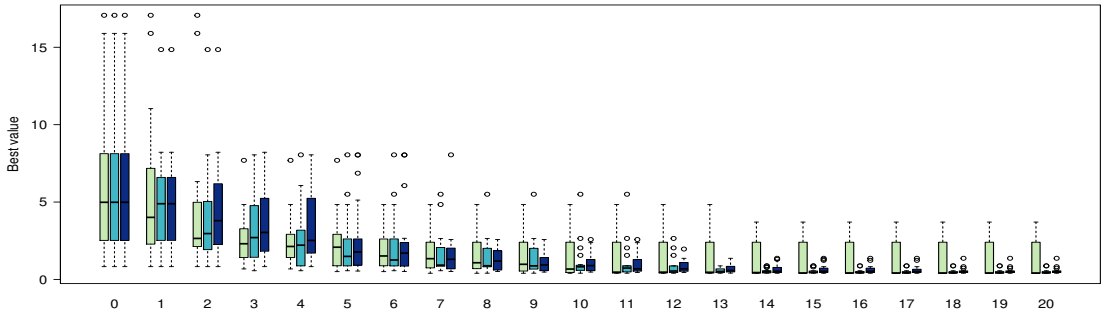


Figure 4.11: Branin: Box plots convergence

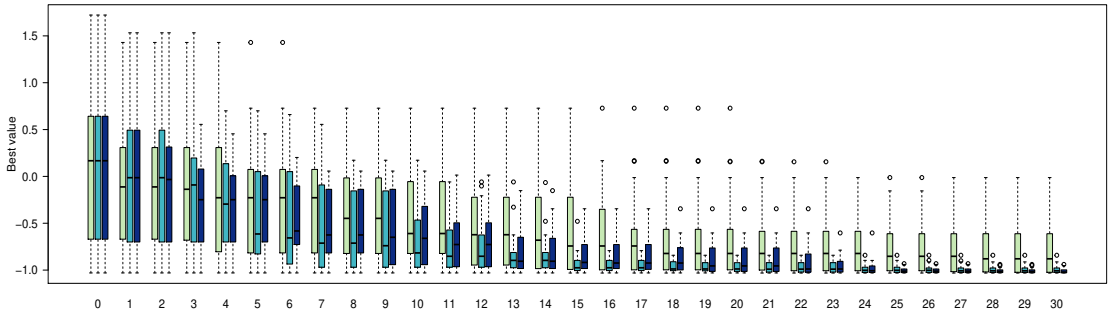


Figure 4.12: Six-hump camel: Box plots convergence

4.11 Appendix A: Optimization test results

In this section, we use boxplots to display the evolution of the best value of the optimization test bench. For each iteration, we display: left: EGO in light green, middle UP-EGO using kriging in light blue, right: UP-EGO using genetic aggregation in dark blue.

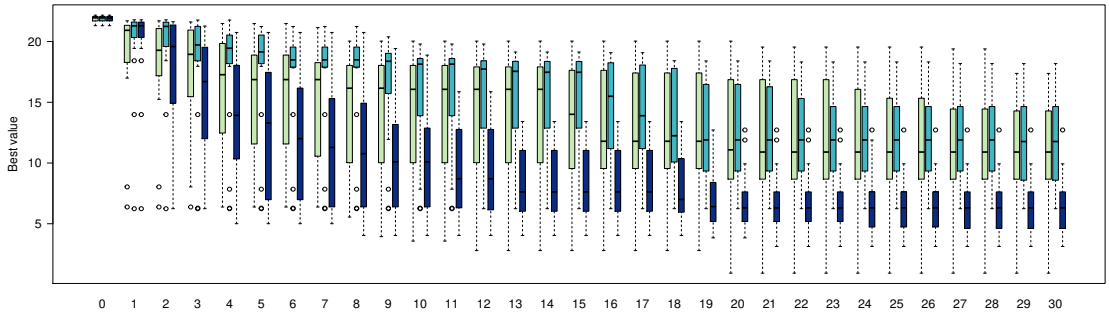


Figure 4.13: Ackley: Box plots convergence

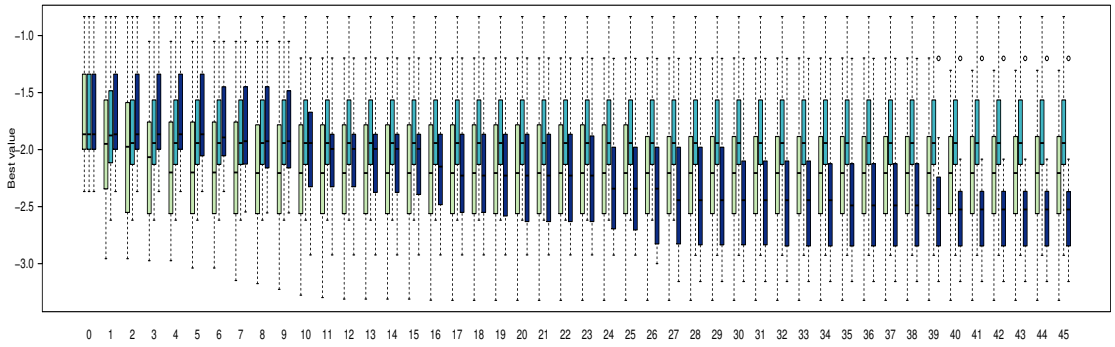


Figure 4.14: Hartmann6: Box plots convergence

Part III

Adaptive feature learning with dimension reduction

Chapter 5

Sequential dimension reduction for learning features of expensive black-box functions

Abstract Learning a feature of an expensive black-box function (optimum, contour line,...) is a difficult task when the dimension increases. A classical approach is two-stage. First, sensitivity analysis is performed to reduce the dimension of the input variables. Second, the feature is estimated by considering only the selected influential variables. This approach can be computationally expensive and may lack flexibility since dimension reduction is done once and for all.

In this paper, we propose a so-called *Split-and-Doubt* algorithm that performs sequentially both dimension reduction and feature oriented sampling. The 'split' step identifies influential variables. This selection relies on new theoretical results on Gaussian process regression. We prove that large correlation lengths of covariance functions correspond to inactive variables. Then, in the 'doubt' step, a doubt function is used to update the subset of influential variables. Numerical tests show the efficiency of the *Split-and-Doubt* algorithm.

5.1 Introduction

In design problems, the goal may be the estimation of a feature of an expensive black-box function (optimum, probability of failure, level set, ...). Several methods have been proposed to achieve this goal. Nevertheless, they generally suffer from the curse of dimensionality. Thus, their usage is limited to functions depending on a moderate number of variables. Meanwhile, most of real life problems are complex and may involve a large number of variables.

Let us focus first on high-dimensional optimization problems. In this context, we look for a good approximation of a global minimum of an expensive-to-evaluate black-box function $f : \Omega = [0, 1]^D \to \mathbb{R}$ using a limited number of evaluations of f. That is, we aim at approximating $x^\star \in \Omega$ such that:

$$x^\star \in \arg\min_{x \in \Omega} f(x) \tag{5.1}$$

Bayesian optimization (BO) techniques have been successfully used in various problems [Moč82,JSW98,Jon01,Sas02,SV99]. These methods give interesting results when the number of evaluations of the function f is relatively low [HHLB11]. They are generally limited to problems of moderate dimension, typically up to about 10 [WHZ+16]. Here, we are particularly interested in the case where the dimension D is large and the number of influential variables d, also called *effective dimension*, is much smaller: $d << D$. In this case, there are different approaches to tackle the dimensionality problem.

A direct approach consists in first performing global sensitivity analysis. Then, the most influential variables are selected and used in the parametric study. Chen et al. [CKC12] stated that "*Variables selection and optimization have both been extensively studied separately from each other*". Most of these methods are two-stage: First, the influential variables are selected and then optimization is performed on these influential variables. These strategies are generally computationally expensive. Furthermore, the set of selected variables does not take into account the new data. However, this new information may modify the results of the sensitivity analysis study. For an overview of global sensitivity analysis methods, one may refer to [IL15].

Some Bayesian optimization techniques are designed to handle the dimensionality problem. For instance, the method called Random EMbedding Bayesian Optimization

(REMBO) selects randomly the subspace of influential variables [WHZ+16, BGR15b]. The main strengths of REMBO are that the selected variables are linear combinations of the input variables and that it works for huge values of D. However, the effective dimension d must be specified.

In this paper, we propose a versatile sequential dimension reduction method. At each iteration, the effective dimension d is estimated and so should not be specified. Moreover, the design is sequentially generated in order to achieve jointly two goals. The first goal is the learning of the influential variables. The second one is the estimation of the optimum (in the optimization case). The algorithm selects the set of influential variable based on the values of the correlation lengths of Automatic Relevance Determination (ARD) covariance. We show theoretical results that support the intuition that large correlation lengths correspond to inactive variables.

The paper is organized as follows. Section 5.2 presents the background and the notations. Section 5.3 introduces the so-called *Split-and-Doubt*. The algorithm is based on theoretical results stated in Section 5.4. Finally, Section 5.5 illustrates the performance of the algorithm on various test functions. For readability, proofs are postponed to Section 5.6.

5.2 General notations and background

5.2.1 Gaussian Process Regression (GPR)

Kriging or Gaussian process regression (GPR) models predict the outputs of a function $f\colon \Omega = [0,1]^D \to \mathbb{R}$, based on a set of n observations [Ste12, RW06]. It is a widely used surrogate modeling technique. Its popularity is mainly due to its statistical nature and properties. Indeed, it is a Bayesian inference technique that provides an estimate of the prediction error distribution. This uncertainty is an efficient tool to construct strategies for various problems such as prediction refinement, optimization or inversion.

The GPR framework uses a centered real-valued Gaussian Process (GP) Y over Ω as a prior distribution for f. The predictions are given by the conditional distribution of Y given the observations $y = (y_1, \ldots, y_n)^\top$ where $y_i = f(x^{(i)})$ for $1 \leq i \leq n$. We denote by $k_\theta : \Omega \times \Omega \to \mathbb{R}$ the covariance function (or kernel) of Y: $k_\theta(x, x') = \mathrm{Cov}[Y(x), Y(x')]$

$((x, x') \in \Omega^2)$, by $X = \begin{pmatrix} x^{(1)}, & \dots, & x^{(n)} \end{pmatrix}^\top \in \Omega^n$ the matrix of observation locations and by $Z = \begin{pmatrix} X & y \end{pmatrix}$ the matrix of observation locations and values where $x^{(i)} = (x_1^{(i)}, \dots, x_D^{(i)})$ for $1 \leq i \leq n$. Without loss of generality, we consider the simple kriging framework. The *a posteriori* conditional mean $\widehat{m}_{\theta,Z}$ and the *a posteriori* conditional variance $\widehat{\sigma}^2_{\theta,Z}$ are given by:

$$\widehat{m}_{\theta,Z}(x) = k_\theta(x, X)^\top K_\theta^{-1} y \tag{5.2}$$

$$\widehat{\sigma}^2_{\theta,Z}(x) = k_\theta(x, x) - k_\theta(x, X)^\top K_\theta^{-1} k_\theta(x, X) \tag{5.3}$$

Here, $k_\theta(x, X)$ is the vector $(k_\theta(x, x^{(1)}), \dots, k_\theta(x, x^{(n)}))^\top$ and where $K_\theta = k_\theta(X, X)$ is the invertible matrix with entries $\Big(k_\theta(X, X)\Big)_{ij} = k_\theta(x^{(i)}, x^{(j)})$, for $1 \leq i, j \leq n$.

Several methods are used to select the covariance function. A common approach consists in assuming that the covariance function belongs to a parametric family. In this paper, we consider the Automatic Relevance Determination (ARD) kernels defined in (5.4). A review of covariance functions is given in [Abr97].

$$k_\theta(x, y) = \sigma^2 \prod_{p=1}^{D} k\Big(\frac{d(x_p, y_p)}{\theta_p}\Big), \text{for } x, y \in \Omega. \tag{5.4}$$

Here, $d(,)$ is a distance on $\Omega \times \Omega$ and $k : \mathbb{R} \to \mathbb{R}$ is a fixed stationary covariance function. The hyper-parameters σ and $\theta_1, \dots, \theta_D$ for $i \in 1, \dots, n$ have to be estimated. The ARD kernels include most popular kernels such as the exponential kernel, the Matérn $5/2$ kernel and the double exponential kernel.

The hyper-parameters of these parametric families can be estimated by maximum Likelihood (ML) or cross validation (CV). Both methods have interesting asymptotic properties [Bac13a, Bac14, BLN17]. Nevertheless, when the number of observations is relatively low, the estimation can be misleading. These methods are also computationally demanding when the number of observations is large.

On one hand, estimating the correlation lengths by the maximum likelihood estimator gives the estimator $\widehat{\theta}^\star_{MLE} \in \arg\max_\theta l_Z(\theta)$ where the likelihood $l_Z(\theta)$ is given in (5.5).

$$l_Z(\theta) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|k_\theta(X, X)|}} \exp\Big(-y^\top k_\theta(X, X)^{-1} y\Big). \tag{5.5}$$

On the other hand, the idea behind Cross-validation (CV) is to estimate the prediction errors by splitting the observations once or several times. One part is used as a test set while the remaining parts are used to construct the model. The Leave-One-Out Cross-Validation (LOO-CV) consists in dividing the n point into n subset of one point each. Then, each subset plays the role of test set while the remaining points are used together as the training set. Using Dubrule's formula [Dub83], the LOO-CV estimator is given in (5.6).

$$\widehat{\theta}^{\star}_{CV} \in \arg\min_{\theta} \frac{1}{n} y^{\top} K_{\theta}^{-1} \operatorname{diag}(K_{\theta}^{-1})^{-1} K_{\theta}^{-1} y \tag{5.6}$$

For more insight on these estimators, one can refer to [Bac13b].

5.2.2 Derivative based global sensitivity measures: DGSM

Sobol' and Kucherenko [SG95,SK09] proposed the so-called Derivative-based Global Sensitivity Measures (DGSM) to estimate the influence of an input variable of a function $f : \Omega = [0,1]^D \to \mathbb{R}$. For each variable χ_i, the index ϑ_i is the global energy of the corresponding partial derivatives.

$$\vartheta_i(f) = \int_{\Omega} \Big(\frac{\partial f(x)}{\partial \chi_i}\Big)^2 dx, i = 1, \ldots, D. \tag{5.7}$$

DGSM provides a quantification of the influence of a single input on f. Indeed, assuming that f is of class C^1, then χ_i is not influential iff $\frac{\partial f}{\partial \chi_i}(x) = 0, \forall\, x \in \Omega$ iff $\vartheta_i = 0$.

DGSM has recently shown its efficiency for the identification of non-influential inputs [RFIK14]. We further define the normalized DGSM $\widehat{\vartheta}$ in (5.8). $\widehat{\vartheta}_i$ measures the influence of χ_i with regard to the total energy.

$$\widehat{\vartheta}_i(f) = \frac{\vartheta_i(f)}{\sum_{p=1}^{D} \vartheta_p(f)}, i = 1, \ldots, D. \tag{5.8}$$

5.3 The *Split-and-Doubt* Algorithm

5.3.1 Definitions

Variable splitting Let us consider the framework of a GPR using a stationary ARD kernel. Intuitively, large correlation lengths values correspond to inactive variables in the

function. We prove this intuition in Proposition 3. The influential variables are selected in our algorithm according to the estimated value of their corresponding correlation lengths. We show also that the ML (and CV) estimator is able to assign asymptotically large correlation length value to the inactive variables (Propositions 5 and 6).

Let $\widehat{\theta^\star} = (\widehat{\theta^\star}_1, .., \widehat{\theta^\star}_D)$ be the ML estimation of the correlation lengths:

$$\widehat{\theta^\star} \in \arg\max_\theta l_Z(\theta).$$

The influential variables are then selected according to the estimated value of their corresponding correlation lengths. We split the indices into a set of influential variables I_M and a set of minor variables I_m as follows:

- $I_M = \{i; \widehat{\theta^\star}_i < T\}$
- $I_m = \{i; \widehat{\theta^\star}_i \geq T\}$

where $T \in \mathbb{R}$ is a suitable threshold. Let d_M (resp. d_m) be the size of I_M (resp. I_m). We further call $\Omega_m := [0,1]^{d_m}$ the minor subspace, that is the space of minor variables and $\Omega_M := [0,1]^{d_M}$ the major subspace, that is the subspace of major variables. We will use the set notation: for a set I of $\{1, \ldots, D\}$, x_I will denote the vector extracted from x with coordinates x_i, $i \in I$. Hence, x_{I_M} (resp. x_{I_m}) denotes the sub-vector of x whose coordinates are in the major (resp. minor) subspace. For simplicity, we will also write $x = (x_{I_M}, x_{I_m})$, without specifying the re-ordering used to obtain x by gathering x_{I_M} and x_{I_m}.

Doubt The doubt measures the influence of the variables from the minor subspace Ω_m. It is a decreasing function of the correlation lengths. We will use it to question the variable splitting.

Definition 13 (Doubt)**.** *Let δ be a function associated with a variable splitting (I_m, I_M) such that for all vector $\theta = (\theta_1, \ldots, \theta_D) \in \mathbb{R}^D$:*

$$\delta(\theta) = \sum_{i \in I_m} \max(\theta_i^{-1} - T^{-1}, 0).$$

Contrast Given two different correlation lengths $\theta^{(1)}$ and $\theta^{(2)}$ and a location x, the contrast measures the discrepancy between the corresponding predictions at x. It will be used to build a sequential design in the minor subspace.

Definition 14 (Prediction contrast)**.** *For a point x and two correlation lengths $\theta^{(1)}$ and $\theta^{(2)}$, the prediction contrast $PC(x, \theta^{(1)}, \theta^{(2)})$ is*

$$PC(x, \theta^{(1)}, \theta^{(2)}) = \left| \widehat{m}_{\theta^{(1)},Z}(x) - \widehat{m}_{\theta^{(2)},Z}(x) \right|.$$

5.3.2 The algorithm

The *Split-and-Doubt* algorithm performs a new variable selection at each iteration. It samples a point in two steps: a goal-oriented sampling in the major subspace and a sampling of the minor variables to question the variable selection done at the previous step. The *Split-and-Doubt* corresponding to the optimization goal, with the expected improvement (EI) criterion [JSW98] is described below:

Here, the algorithm is applied for optimization (Step 3). We used the Expected Improvement criterion (5.9).

$$EI_Z(x) = \mathbb{E}\Big[\max(\min_i y_i - Y(x), 0) | Z \Big] \tag{5.9}$$

It is important to recall that it is here possible to use any other optimization criterion to sample $x_M^\star$. We can use other criteria for other purposes such as contour estimation [PGR$^+$10,RBM08,BES$^+$08], probability of failure estimation [BGL$^+$12] or surrogate model refinement [BFI07].

The settings of the algorithm are mainly the kernel k, the limit ℓ and the threshold T. Another hidden setting is the search space for the ML estimator. We use a Matérn 5/2 kernel and we set $\ell = \text{erf}(\frac{1}{\sqrt{2}})$ and an adaptive threshold $T = 20 \min_{i \in [1,D]} (\widehat{\theta^\star}_i)$.

5.3.3 Remarks on the steps of the *Split-and-Doubt* algorithm

Remark on the doubt When the observations do not carry enough information, it is hard to estimate accurately the correlation lengths. The use of such values can lead to unsatisfactory results [FJ08,BBV11]. In our algorithm, the estimated correlation lengths are used to select the major variables. So, it is important to always question the estimation. If this is done once and for all, poor estimation can lead to considering a major variable

Algorithm 4: *Split-and-Doubt*-EGO (f) algorithm

Data: Design Points $Z = (X, y)$

Algorithm parameters: ℓ, kernel k, threshold T;

repeat

1. Estimate the correlation lengths: $\widehat{\theta^\star} \in \underset{\theta \in (\mathbb{R}_+^\star)^D}{\arg\max}\, l_Z(\theta)$ (Eq. (5.5));

2. Split the variables: Define the major set $I_M = \{i; \widehat{\theta}_i^\star < T\}$ and the minor set $I_m = \{i; \widehat{\theta}_i^\star \geq T\}$, $d_m = |I_m|$;

3. Design in the major subspace: Compute $x_M^\star$ according to the objective function in the major subspace (by EI for instance): We compute a new GPR considering only the major variables to compute the EI. Let $Z_M = (X_{I_M}, y)$

$$x_M^\star \in \underset{x_M \in \Omega_M}{\arg\max}\, \mathrm{EI}_{Z_M}(x_M)$$

4. Doubt the variable splitting: Compute a challenger θ' for correlation lengths.

$$\theta' \in \underset{\theta \in (\mathbb{R}_+^\star)^D}{\arg\max}\, \delta(\theta) \quad \text{subject to } 2\left|\ln\left(\frac{l_Z(\theta)}{l_Z(\widehat{\theta^\star})}\right)\right| < \chi^2(\ell, d_m)$$

5. Design in the minor subspace: Compute $x_m^\star$ by maximum contrast with the challenger θ'

$$x_m^\star \in \underset{x_m \in \Omega_m}{\arg\max}\, \mathrm{PC}\Big(x = (x_M^\star, x_m), \widehat{\theta^\star}, \theta'\Big)$$

6. Update: Evaluate the new point output $y_{n+1} = f(x^{(n+1)})$ with $x_{I_m}^{(n+1)} = x_M^\star$ and $x_{I_m}^{(n+1)} = x_m^\star$ and add the new point to the design:

$$X^\top \leftarrow \begin{pmatrix} X^\top & x^{(n+1)} \end{pmatrix}, y^\top \leftarrow (y^\top, y_{n+1})$$

until *this end condition*;

Result: $Z = (X, y)$

inactive. Therefore, we look for a "challenger kernel" at each iteration. Specifically, we are looking for correlation lengths that maximize the doubt and that are accepted by a likelihood ratio test. Indeed, this is why we limit the search space by a likelihood ratio deviation from the estimated correlation lengths $\widehat{\theta^\star}$: $\Theta_l = \{\theta; 2\Big|\ln\Big(\frac{l_Z(\theta)}{l_Z(\widehat{\theta^\star})}\Big)\Big| < l\}$. Note that we used $l = \chi^2(\ell, d_m)$. Following [FJ08, CKC12], the likelihood ratio test is compared to the χ^2 distribution to decide whether the correlation lengths are allowable or not.

Remark on the contrast Sampling the coordinates in the non-influential variables subspace $x_M^\star + \Omega_m$ aims at revealing the contrast between the maximum likelihood correlation lengths $\widehat{\theta^\star}$ and the challenger correlation lengths θ'. The main idea is to sample the point that helps either correcting the first estimation or reducing the allowable doubt space Θ in order to strengthen the belief in the estimated kernel.

We could have used an alternative direct approach. It consists in maximizing the likelihood ratio between two estimations of the correlation lengths in the future iterations.

Definition 15 (likelihood contrast)**.** *For a point x and two correlation lengths $\theta^{(1)}$ and $\theta^{(2)}$, the likelihood contrast LC is:*

$$LC(x, \theta^{(1)}, \theta^{(2)}) = \mathbb{E}\Big[\Big|\ln\Big(\frac{L(\theta^{(1)}, Z \cup (x, \widehat{Y}(x)))}{L(\theta^{(2)}, Z \cup (x, \widehat{Y}(x)))}\Big)\Big|\Big]$$

where $\widehat{Y}(x) \sim \mathcal{N}\Big(\widehat{m}_{\theta_2,Z}(x), (\widehat{\sigma}_{\theta_2,Z}(x))^2\Big)$.

However, this approach is computationally more expensive. Therefore, we prefer to use the prediction contrast (Definition 14).

5.3.4 Example: Illustration of the contrast effect

We illustrate here how the Doubt/ Contrast strategy can help correcting an inaccurate variable splitting. To do so, let us consider the following example. Let $f(x_1, x_2) = \cos(2\pi x_2)$. We assume that we have at hands four design points $x^{(1)} = (0, \frac{2}{3})$, $x^{(2)} = (\frac{1}{3}, 0)$, $x^{(3)} = (\frac{2}{3}, 1)$ and $x^{(4)} = (1, \frac{1}{3})$ and their corresponding responses $y_1 = y_4 = f(x^{(1)}) = f(x^{(4)}) = -0.5$ and $y_2 = y_3 = f(x^{(2)}) = f(x^{(3)}) = 1$. Here, the search space for the correlation lengths is $[0.5, 10]^2$.

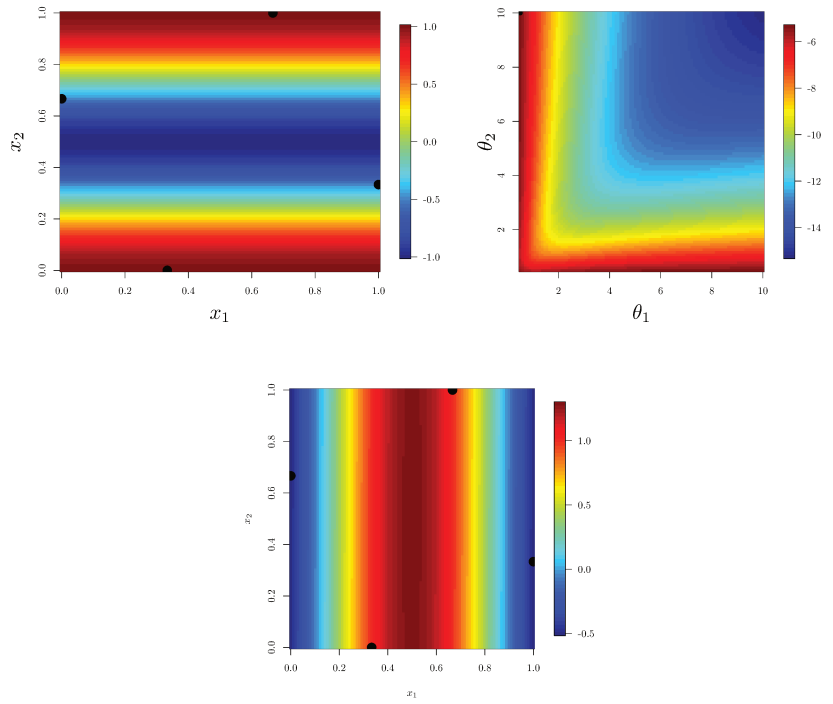


Figure 5.1: Upper left: $f(x_1, x_2) = \cos(2\pi x_2)$, the color code indicates the values of the f and solid black circles indicate the design points. Upper right: log-likelihood of the correlation lengths, solid black circle: $\widehat{\theta^\star}$. Bottom: The predictions given by the GPR using $k_{\widehat{\theta^\star}}$.

Misleading estimation The log-likelihood of the correlation lengths in the search space $[0.5, 10]^2$ for the Matérn 5/2 kernel is displayed in Figure 5.1. Notice that the likelihood is maximized for different values of θ and that $\widehat{\theta^\star} = (0.5, 10)$ is among these values:

$$(0.5, 10) \in \underset{\theta \in (\mathbb{R}_+^\star)^D}{\arg\max}\, l_Z(\theta).$$

We also display in Figure 5.1 the function f, the design points, and the predictions using $k_{\widehat{\theta^\star}}$. This example shows that a limited number of observations may lead to inaccurate correlation lengths and consequently inaccurate predictions.

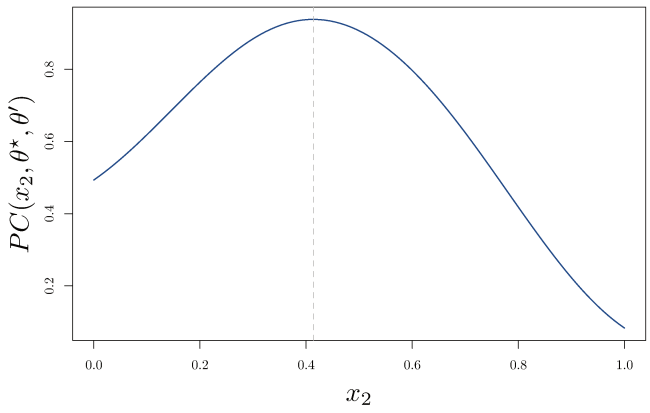


Figure 5.2: The prediction contrast in function of x_2.

Doubt/Contrast strategy Adding more points will arguably improve the quality of the correlation lengths estimation. Here, we want to bold that the improvement due to the Doubt/Contrast strategy is not only related to the fact that more points are sampled. To do so, we set $T = 10$. So, $I_M = \{1\}$ and $I_m = \{2\}$. Notice that the challenger correlation lengths is $\theta' = (0.5, 0.5)$. It gives the maximum doubt $\delta(\theta') = \frac{1}{0.5} - \frac{1}{10} = 1.9$. In this example, we are sampling the fifth point $x^{(5)}$. The value sampled by the EI in the major space is $x_M^\star = 0.64$. We display in Figure 5.2 the prediction contrast $PC((x_M^\star, x_2), \widehat{\theta^\star}, \theta')$ as a function of x_2.

Let us now consider two cases: a) we add the point sampled by the maximum contrast $(x_M^\star, x_m^\star)$ and b) we add the point with the minimum contrast $(x_M^\star, 1)$. For both cases, we display the updated likelihood function in Figure 5.3.

Notice that:

a) For $x_m = x_m^\star$ (maximum contrast), the log-likelihood has larger value for small values of θ_2. Thus, the same inaccurate variable splitting is prevented.

b) For $x_m = 1$ (small contrast value), we may still use the same misleading variable splitting.

Even if this 2-dimensional toy example is pathological, it illustrates the interests of the Doubt/Contrast strategy. This strategy can be valuable in high dimension when the number of observations is relatively small to estimate accurately the correlation lengths.

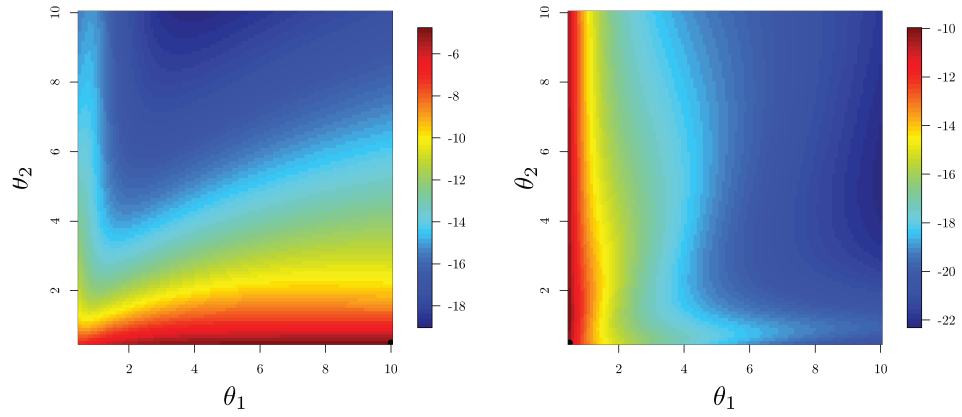


Figure 5.3: Left: The log likelihood of the correlation lengths if we add $\big((x_M^\star, x_m^\star), f(x_M^\star, x_m^\star)\big)$. Right: The log likelihood of the correlation lengths if we add $\big((x_M^\star, 1), f(x_M^\star, 1)\big)$.

5.4 Links between correlation lengths and variable importance

In this section, we consider a deterministic function $f : \mathbb{R}^D \to \mathbb{R}$ to be modeled as a GP path. We consider the centered stationary GP with a covariance function k_θ defined by

$$k_\theta(\mathbf{h}) = \prod_{i=1}^{D} k(h_i/\theta_i),$$

where $k : \mathbb{R} \to \mathbb{R}$ is a fixed covariance function satisfying $k(0) = 1$ and $\theta \in (0, \infty)^D$ is the vector of correlation lengths. As an example, the k may be the function $k(h) = e^{-h^2}$.

Intuitively, a small correlation length θ_i for the GP should correspond to an input variable χ_i that has an important impact on the function value $f(x)$. Conversely, if the function f does not depend on χ_i, then the length θ_i should ideally be infinite. This intuition is precisely the motivation for the *Split-and-Doubt* algorithm suggested in Section 5.3.

In this section, we show several theoretical results that confirm this intuition. First, we show that if the correlation length θ_i goes to zero (respectively infinity) then the derivative-based global sensitivity measure, obtained from the GP predictor for the input

x_i, tends to its maximum value 1 (resp. its minimum value 0). Then, we show that an infinite correlation length θ_i can provide an infinite likelihood or a zero LOO mean square error, for the GP model, when the function f does not depend on x_i.

We use the additional following notations throughout the section. For $D, p, q \in \mathbb{N}^\star$, for a covariance function g on $\mathbb{R}^D$, for two $p \times D$ and $q \times D$ matrices X and Z, we denote by $g(X, Z)$ be the $p \times q$ matrix defined $[g(X, Z)]_{i,j} = g(X_i, Z_j)$ where M_l is the line l of a matrix M. When $d = 1$, $p = 1$ or $q = 1$, we identify the corresponding matrices with vectors. We assume that for any $p, d \in \mathbb{N}$, for any $\theta \in (0, \infty)^D$, for any $p \times d$ matrix X with two-by-two distinct lines, the matrix $k_\theta(X, X)$ is invertible. Further, for any vector u, u_{-1} is obtained from u by removing the i^{th} component of u.

5.4.1 Correlation lengths and derivative-based global sensitivity measures

Consider the function f to be observed at the locations $x^{(1)}, ..., x^{(n)} \in \Omega$, with $n \in \mathbb{N}$ and for a bounded domain $\Omega \subset \mathbb{R}^D$. Let X be the $n \times D$ matrix with lines given by $x^{(1)}, ..., x^{(n)}$, y be the vector of responses $y = (f(x^{(1)}), ..., f(x^{(n)}))^\top$ and Z the $(n+1) \times D$ matrix $Z = \begin{pmatrix} X & y \end{pmatrix}$

Recall that the prediction of f at any line vector $x \in \Omega$, from the GP model, is given by $\widehat{m}_{\theta,Z}(x) = r_\theta(x)^\top K_\theta^{-1} y$, with $r_\theta(x) = k(x, X)$, $K_\theta = k_\theta(X, X)$. Then, we use the notation $\vartheta_i(\theta)$ for the DGSM index of the variable χ_i on the predictor function $\widehat{m}_{\theta,Z}(x)$:

$$\vartheta_i(\theta) = \vartheta_i(\widehat{m}_{\theta,Z}) = \int_\Omega \left(\frac{\partial \widehat{m}_{\theta,Z}(x)}{\partial \chi_i} \right)^2 dx.$$

We also use the following notation for the normalized DGSM index the variable χ_i:

$$\widehat{\vartheta}_i(\theta) = \widehat{\vartheta}_i(\widehat{m}_{\theta,Z}) = \frac{\vartheta_i(\theta)}{\sum_{r=1}^{D} \vartheta_r(\theta)}.$$

The normalized DGSM indices $\widehat{\vartheta}_i(\theta)$ satisfies $0 \leq \widehat{\vartheta}_i(\theta) \leq 1$. The larger this indice is, the more important the variable χ_i is for $\widehat{m}_{\theta,Z}(x)$. In the two next propositions, we show that, under relatively minimal conditions, we have $\widehat{\vartheta}_i(\theta) \to 1$ as $\theta_i \to 0$ and $\widehat{\vartheta}_i(\theta) \to 0$ as $\theta_i \to \infty$. Hence, we give a theoretical support to the intuition that small correlation lengths correspond to important input variables.

Proposition 3. *Assume that the components of y are not all equal. Assume that k is continuously differentiable on $\mathbb{R}$. Let $i \in \{1, ..., D\}$ be fixed. For $j = 1, ..., n$ let $v^{(j)} = x^{(j)}_{-i}$. Assume that $v^{(1)}, ..., v^{(n)}$ are two by two distinct. Then, for fixed $\theta_{-i} \in (0, \infty)^D$*

$$\widehat{\vartheta}_i(\theta) \underset{\theta_i \to \infty}{\longrightarrow} 0.$$

Proposition 4. *Assume that the components of y are not all equal. Consider the same notation as in Proposition 3. Assume that k is continuously differentiable on $\mathbb{R}$, that $k(t) \to 0$ as $|t| \to \infty$ and that Ω is an open set. Assume also that $x^{(1)}, ..., x^{(n)}$ are two-by-two distinct. Let $i \in \{1, ..., d\}$ be fixed. Then for fixed $\theta_{-i} \in (0, \infty)^{d-1}$*

$$\widehat{\vartheta}_i(\theta) \underset{\theta_i \to 0}{\longrightarrow} 1.$$

In Propositions 3 and 4, the regularity conditions on k are mild, and the conditions on $x^{(1)}, ..., x^{(n)}$ hold in many cases, for instance when $x^{(1)}, ..., x^{(n)}$ are selected randomly and independently or from a latin hypercube procedure (see e.g. [SWN13]).

5.4.2 Estimated correlation lengths and inactive variables

We first recall the likelihood function:

$$l_Z(\theta) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|k_\theta(X, X)|}} \exp\left(-y^\top k_\theta(X, X)^{-1} y\right).$$

In the next proposition, we show that, if the function f does not depend on the variable χ_i, then the likelihood $l_Z(\theta)$ goes to infinity when θ_i goes to infinity. This is a theoretical confirmation that maximum likelihood can detect inactive input variables and assign them large correlation lengths.

Proposition 5. *Assume that k is continuous. Assume that for any $\theta \in (0, \infty)^D$, the reproducing kernel Hilbert space (RKHS) of the covariance function k_θ contains all infinitely differentiable functions with compact supports on $\mathbb{R}^D$.*

Let $i \in \{1, ..., D\}$ be fixed. For $j = 1, ..., n$ let $v^{(j)} = x^{(j)}_{-i}$. Assume that

i) $x^{(1)}, ..., x^{(n)}$ are two-by-two distinct;

ii) $y_r = y_s$ if $v^{(r)} = v^{(s)}$;

iii) there exist $a, b \in \{1, ..., n\}$ with $a \neq b$ so that $v_a = v_b$.

Then, for fixed $\theta_{-i} \in (0, \infty)^D$

$$l_Z(\theta) \underset{\theta_i \to \infty}{\longrightarrow} \infty$$

In Proposition 5, the conditions i), ii) and iii) are quite minimal. The condition i) ensures that the likelihood is well-defined, as the covariance matrix is invertible for all $\theta \in (0,\infty)^D$. The condition ii) holds when $f(x)$ does not depend on x_i. The condition iii) is necessary to have $l(\theta)$ going to infinity, since if $v^{(1)},...,v^{(n)}$ are two by two distinct, the determinant of $k_\theta(X,X)$ remains bounded from below as $\theta_i \to \infty$ (see also the proof of Proposition 3). Note that the conditions ii) and iii) together imply that there is a pair of input points x_a, x_b for which only the value of the i-th component changes and the value of f does not change, which means that the data set presents an indication that the input variable χ_i is inactive.

We refer to, e.g., [Wen04] for a reference to the RKHS notions that are used in this section. There are many examples of stationary covariance functions k satisfying the RKHS condition in Proposition 5. In particular, let $\widehat{k_\theta}$ be the Fourier transform of k_θ defined by $\widehat{k_\theta}(w) = \int_{\mathbb{R}^D} k_\theta(x) e^{-\mathrm{i}w^\top x} dx$ with $\mathrm{i}^2 = -1$. Then, if there exists $\tau < \infty$ so that $\widehat{k_\theta}(w)||w||^\tau \to \infty$ as $||w|| \to \infty$, then the RKHS condition of Proposition 5 holds. [This follows from Theorem 10.12 in [Wen04] and from the fact that an infinitely differentiable function with compact support ϕ has a Fourier transform $\widehat{\phi}$ satisfying $\widehat{\phi}(w)||w||^\gamma \to 0$ as $||w|| \to \infty$ for any $\gamma < \infty$.

Hence, Lemma 5 holds in particular when k is the exponential covariance function with $k(t) = e^{-|t|}$. Lemma 5 also holds when k is the Matérn covariance function with

$$k\left(t\right) = \frac{1}{\Gamma\left(\nu\right) 2^{\nu-1}} \left(2\sqrt{\nu}|t|\right)^{\nu} K_\nu \left(2\sqrt{\nu}|t|\right),$$

where $0 < \nu < \infty$ is the smoothness parameter (see e.g. [Ste12]). It should however be noted that the double exponential covariance function k (defined by $k(t) = \exp(-t^2)$ with $t \in \mathbb{R}$) does not satisfy the condition of Lemma 5. [Notice that [XS17] study specifically the asymptotic behavior of the maximum likelihood estimation of a variance parameter for the Gaussian covariance function, when the number of observations of a smooth function goes to infinity.]

In the next proposition, we study the LOO mean square prediction error

$$CV_Z(\theta) = \sum_{j=1}^{n} (y_j - \widehat{y}_{\theta,j})^2,$$

with $\widehat{y}_{\theta,j} = k_\theta(x^{(j)}, X_{-j})k_\theta(X_{-j}, X_{-j})^{-1}y_{-j}$, where X_{-j} and y_{-j} are obtained, respectively, by striking off the line j of X and the component j of y. We show that, similarly as for the likelihood, inactive variables can be detected by this LOO criterion, since we can have $CV_Z(\theta) \to 0$ as $\theta_i \to \infty$ if the function f does not depend on χ_i

For $j = 1, ..., n$ let $v^{(j)} = x^{(j)}_{-i}$.

Proposition 6. *Let k satisfy the same conditions as in Lemma 5. Let $i \in \{1, ..., d\}$ be fixed.*

For $j = 1, ..., n$ let $v^{(j)} = x^{(j)}_{-i}$. Assume that

i) $x^{(1)}, ..., x^{(n)}$ are two-by-two distinct;

ii) $y_r = y_s$ if $v^{(r)} = v^{(s)}$;

iii) for all $r \in \{1, ..., n\}$ there exists $s \in \{1, ..., n\}$, $r \neq s$, so that $v^{(r)} = v^{(s)}$.

Let θ_{-i} be obtained from θ by removing its component i. Then, for any fixed $\theta_{-i} \in (0, \infty)^{d-1}$, we have

$$CV_Z(\theta) \underset{\theta_i \to \infty}{\longrightarrow} 0.$$

In Proposition 6, the conditions i) and ii) are interpreted similarly as for Proposition 5. The condition iii) however provides more restrictions than for the likelihood in Proposition 5. This condition states that for any observation point in the data set, there exists another observation point for which only the inactive input i is changed. This condition is arguably necessary to have $CV_Z(\theta) \to 0$.

5.5 Numerical examples

5.5.1 Tests set

We illustrate the *Split-and-Doubt* on five benchmark optimization problems. The first four are classical synthetic functions, the two-dimensional Branin function, the general Ackley function in six dimension, the six-dimensional Hartmann function and the general Rosenbrock function in five dimensions. The fifth one is the Borehole function [MMY93]. It models the water-flow in a borehole. For each function, we added inactive input variables in order to embed them in a higher dimensional $D^{(i)}$. The settings are summarized in (Table 5.1).

Table 5.1: Optimization test functions.

$f^{(i)}$	$d^{(i)}$	$D^{(i)}$	Number of design points $n_0^{(i)}$	Number of iterations $N_{max}^{(i)}$
Hartmann 6-dim	6	15	30	30
Rosnebrock	5	20	40	60
Ackley	6	20	45	40
Borhole	6	25	30	25
Branin	2	25	30	50

We launched the optimization process for these functions with three different optimization algorithms:

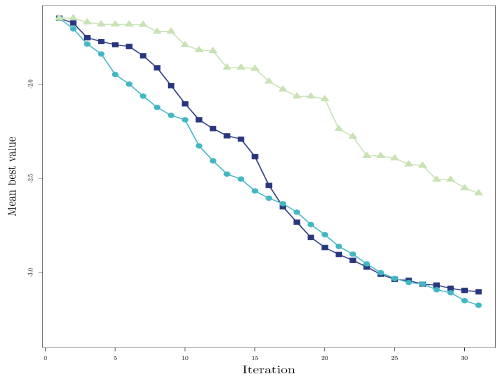
- EGO [JSW98]: Implementation of the R package DiceOptim [RGD12] using the default parameters.
- *Split-and-Doubt* algorithm with Matérn 5/2 covariance function.
- *Split-Without-Doubt* algorithm: It uses the same variable splitting as *Split-and-Doubt* and generates the minor variables by uniform random sampling.

For each function $f^{(i)}$, we launched each optimization process for $N_{max}^{(i)}$ iterations starting with $N_{seed} = 20$ different initial design of experiments of size $n_0^{(i)}$ generated by an maximin space-filling sampling.

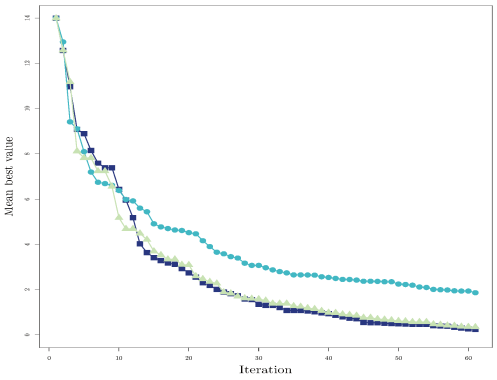
5.5.2 Results

The results are represented by box plots in Appendix 5.8. We also display the mean best value evolution in Figure 5.4.

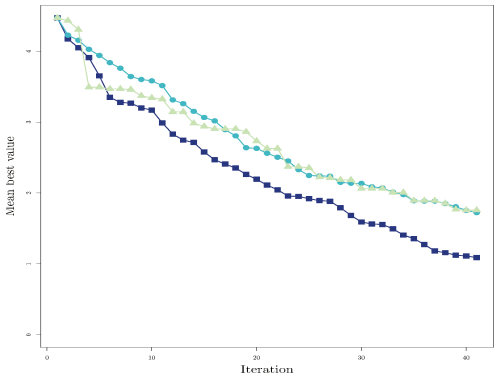
We can see that *Split-and-Doubt* gives better results than EGO for Rosenbrock, Ackley and Borehole function. EGO does not converge for the first two functions and used more iterations for the last one. These cases illustrate the efficiency of the dimension reduction for limited budget optimization. For Branin function the convergence is relatively fast for all the three algorithms. This is due to the fact that the effective dimension is 2 and that the first design of experiments covers well these dimensions.



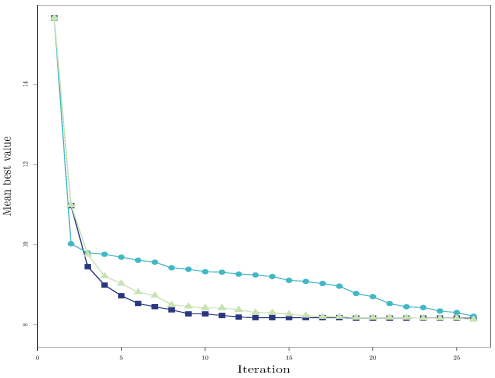
(a) Hartmann 6-dim



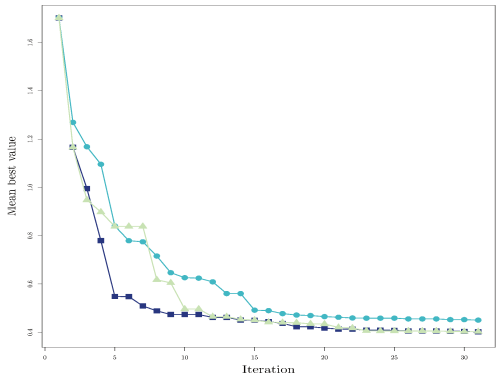
(b) Rosenbrock



(c) Ackley



(d) Borehole



(e) Branin

Figure 5.4: Comparison of 3 optimization strategies. Mean over N_{seed} of the best current value as a function of the number of iterations.

On one hand, sampling the minor variables at random or using the Doubt/Contrast strategy gives close results when the influential variables are easily determined. On the other hand, the efficiency of the Doubt/Contrast approach is visible on Hartmann and Ackley functions, by comparing the results of *Split-and-Doubt* and *Split-without-Doubt*. Notice that we start in general with a relatively small amount of design points. Thus, the initial estimation of the correlation lengths can be inaccurate. In these cases, the Doubt/Contrast approach is valuable to improve the estimation. To further highlight this idea, we display in Figure 5.5 the percentage of undetected influential variables and the miss-classification rate of all the variables for both *Split-and-Doubt* and *Split-without-Doubt* for the Rosenbrock function.

Among the 20 DOE, the *Split-and-Doubt* detects all the major variables for 19 cases starting from iteration 15 and for all the DOE starting from iteration 37. However, the *Split-without-Doubt* struggles to select properly all the influential variables even in the last iterations. Considering all the variables, the miss-classification rate decrease rapidly for the *Split-and-Doubt*. However, for one test a minor variable is considered influential

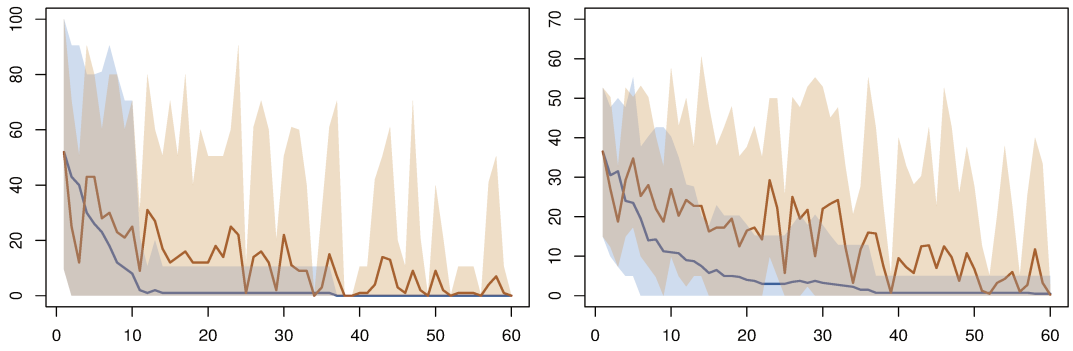


Figure 5.5: Solid line: mean value over 20 repetitions, colored area: 95% confidence interval. Blue: *Split-and-Doubt*, Red: *Split-without-Doubt*. x-axis, iteration number. Left: Miss-classification rate of major variables. Right: Miss-classification rate of all the variables

until the end. This can be explained by the nature of the doubt function that aims at correcting only a miss-classification of an influential variable.

Finally, as we can see in Figure 5.6, *Split-and-Doubt* is faster than EGO in terms of computing time. The fact that we perform two optimization procedures in smaller spaces makes the algorithm faster than optimizing the EI in dimension D.

5.6 Proofs

For the proofs of Propositions 3 and 4, we let $k'(t) = \partial k(t)/\partial t$.

Proof of Proposition 3 . Without loss of generality, we consider $i = 1$ in the proof. Let $\theta_{-1} \in (0, \infty)^{D-1}$ be fixed. We have

$$\frac{\partial}{\partial \chi_j} \widehat{m}_{\theta,Z}(x) = \left(\frac{\partial r_\theta(x}{\partial \chi_j} \right)^T K_\theta^{-1} y.$$

When $\theta_1 \to \infty$, K_θ converges to the $n \times n$ matrix $L_{\theta_{-1}}$ with $(L_{\theta_{-1}})_{pq} = \prod_{r=2}^{D} k([x_r^{(p)} - x_r^{(q)}]/\theta_r)$, by continuity of k. This matrix is invertible by assumption on k and $v^{(1)}, ..., v^{(n)}$.

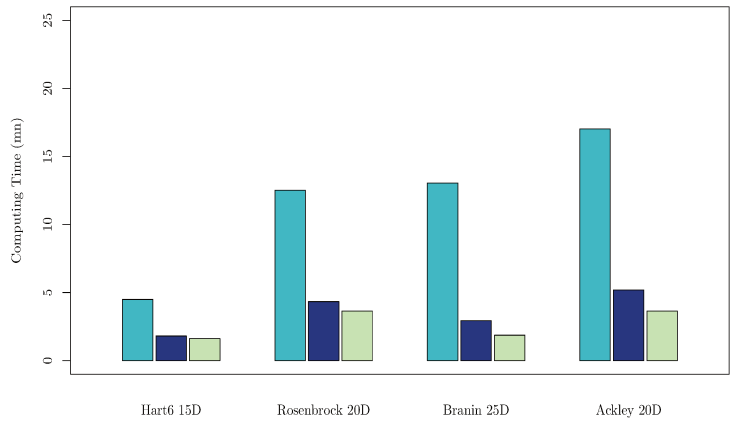


Figure 5.6: Mean computing time: Left: EGO, Middle: *Split-and-Doubt*, Right: *Split-Without Doubt* in minutes.

Hence $||K_\theta^{-1}y||$ is bounded as $\theta_1 \to \infty$. We have for $j = 1, ..., n$

$$\left(\frac{\partial r_\theta(x)}{\partial \chi_1}\right)_j = \frac{1}{\theta_1} k'([x_1 - x_1^{(j)}]/\theta_1) \prod_{p=2}^{D} k([x_p - x_p^{(j)}]/\theta_p).$$

k is differentiable and Ω is bounded. Hence by uniform continuity as $\theta_1 \to \infty$

$$\sup_{x\in\Omega} \left|\left| \frac{\partial r_\theta(x)}{\partial \chi_1} \right|\right| \to 0.$$

Hence, $\vartheta_1(\theta) \to 0$ as $\theta_1 \to \infty$. Let now for $x = (u, v)$ with $u \in \mathbb{R}$, $l_{\theta_{-1}}(x)$ be the $n \times 1$ vector defined by $[l_{\theta_{-1}}(x)]_j = \prod_{r=1}^{d-1} k([v_r - v_r^{(j)}]/\theta_{r+1})$ (we recall that for $j = 1, ..., n$, $v^{(j)} = x_{-1}^{(j)}$). Let $\widehat{g}_{\theta_{-1}}(x) = l_{\theta_{-1}}(v) L_{\theta_{-1}}^{-1} y$. Then for $m = 2, ..., D$, by the triangle and Cauchy-Schwarz inequalities

$$\begin{aligned}
&\left| \frac{\partial \widehat{m}_{\theta,Z}(x)}{\partial \chi_m} - \frac{\partial \widehat{g}_{\theta_{-1}}(x)}{\partial \chi_m} \right| \\
&\leq \left|\left| \frac{\partial r_\theta(x)}{\partial \chi_m} - \frac{\partial l_{\theta_{-1}}(x)}{\partial \chi_m} \right|\right| . \left|\left| K_\theta^{-1} y \right|\right| + \left|\left| \frac{\partial l_{\theta_{-1}}(x)}{\partial \chi_m} \right|\right| . \left|\left| K_\theta^{-1} y - L_{\theta_{-1}}^{-1} y \right|\right| . \qquad (5.10)
\end{aligned}$$

In (5.10), the vector in the first norm has component $r \in \{1, ..., n\}$ equal to

$$(k((u - u_r)/\theta_1)) - 1) \frac{1}{\theta_m} k'([v_{m-1} - v_{m-1}^{(r)}]/\theta_m) \prod_{\substack{p=2,...,D \\ p\neq m}} k([v_{p-1} - v_{p-1}^{(r)}]/\theta_p)$$

which goes to 0 as $\theta_1 \to \infty$, uniformly over $x \in \Omega$, by uniform continuity. The second norm in (5.10) is bounded as discussed above. The third norm in (5.10) does not depend on θ_1 and is thus bounded uniformly over $x \in \Omega$ as $\theta_1 \to \infty$. The fourth norm in (5.10) goes to 0 as $\theta_1 \to \infty$ as discussed above.

Hence, uniformly over $x \in \Omega$,

$$\left| \frac{\partial \widehat{m}_{\theta,Z}(x)}{\partial \chi_m} - \frac{\partial \widehat{g}_{\theta_{-1}}(x)}{\partial \chi_m} \right| \underset{\theta_1 \to \infty}{\longrightarrow} 0.$$

Furthermore, the function $\widehat{g}_{\theta_{-1}}$ is continuously differentiable and non-constant on Ω because $\widehat{g}_{\theta_{-1}}(x^{(r)}) = y_r$ for $r = 1, ..., n$ and because the components of y are not all equal. This implies that

$$\liminf_{\theta_1 \to \infty} \sum_{m=2}^{D} \vartheta_m(\theta) > 0,$$

which concludes the proof. □

Proof of Proposition 4 . As before, we consider $i = 1$ in the proof. We have for $m = 2, ..., D$ and $r = 1, ..., n$

$$\left(\frac{\partial r_\theta(x)}{\partial \chi_m} \right)_r = k([x_1 - x_1^{(r)}]/\theta_1) \frac{1}{\theta_m} k'([x_m - x_m^{(r)}]/\theta_m) \prod_{\substack{j=2,...,D \\ j \neq m}} k([x_j - x_j^{(r)}]/\theta_j).$$

Hence, $||\partial r_\theta(x)/\partial \chi_m||$ is bounded as $\theta_1 \to 0^+$ uniformly in $x \in \Omega$ from the assumptions on k.

For $j = 1, \ldots, n$, let u_j be the first component of $x^{(j)}$ and let $v^{(j)} = x_{-1}^{(j)}$. As $\theta_1 \to 0^+$, the matrix K_θ converges to the $n \times n$ matrix $N_{\theta_{-1}} = \left[\mathbf{1}_{u_p = u_q} (L_{\theta_{-1}})_{pq} \right]_{p,q=1,...,n}$ with the notation of the proof of Proposition 3. The matrix $N_{\theta_{-1}}$ is invertible because its submatrices are invertible. This is so because for any $p = 1, ..., n$ the subset $\{v^{(q)}; q = 1, ..., n, u_q = u_p\}$ is composed of two-by-two distinct elements since $x^{(1)}, ..., x^{(n)}$ are two-by-two distinct.

Hence, $||K_\theta^{-1} y||$ is bounded as $\theta_1 \to 0^+$ and so $\sum_{m=2}^{D} \vartheta_m(\theta)$ is bounded as $\theta_1 \to 0^+$.

Let now $j \in \{1, ..., n\}$ for which $y_j \neq 0$. Let $\delta > 0$, not depending on θ_1, be small enough so that $\prod_{r=1}^{D} [x_r^{(j)} - \delta, x_r^{(j)} + \delta] \in \Omega$. Then we have

$$\sup_{s \in [-\delta, \delta]^D; |s_1| = \sqrt{\theta_1}} \left| \widehat{m}_{\theta,Z}(x^{(j)} + s) \right| \underset{\theta_1 \to 0^+}{\longrightarrow} 0. \tag{5.11}$$

Indeed, we have

$$(r_\theta(x_j+s))_p = k\left(\frac{u_p-u_j-s_1}{\theta_1}\right)\prod_{r=2}^{D} k\left(\frac{(x_p)_r - x_r^{(j)} - s_r}{\theta_r}\right).$$

The product above is bounded uniformly over $s \in [-\delta,\delta]^D$ by uniform continuity of k. Also, whether $u_p - u_j = 0$ or $u_p - u_j \neq 0$, we have

$$\sup_{|s_1|=\sqrt{\theta_1}} k\left(\frac{u_p-u_j-s_1}{\theta_1}\right) \underset{\theta_1\to 0^+}{\longrightarrow} 0.$$

Finally, $||K_\theta^{-1}y||$ is bounded as $\theta_1 \to 0^+$ as discussed above. Hence (5.11) is proved. Also, let $E = \{u_j\} \times \prod_{r=2}^{D}[x_r^{(j)}-\delta, x_r^{(j)}+\delta]$. Then as $\theta_1 \to 0^+$, uniformly over $x \in E$, for $p = 1, ..., n$, we have

$$(r_\theta(x))_p \underset{\theta_1\to 0^+}{\longrightarrow} \mathbf{1}_{\{u_p=u_j\}}\prod_{r=2}^{D} k\left(\frac{x_r-(x_p)_r}{\theta_r}\right).$$

Also $K_\theta^{-1}y \underset{\theta_1\to 0^+}{\longrightarrow} N_{\theta_{-1}}y$ as discussed above. Hence, as $\theta_1 \to 0^+$, $\widehat{m}_{\theta,Z}(x)$ converges uniformly over $x \in E$ to a function value $\widehat{g}_{\theta_{-1}}(x)$, with $\widehat{g}_{\theta_{-1}}(x)$ continuous with respect to $x \in E$. Since $\widehat{m}_{\theta,Z}(x_j) = y_j$, we can choose the $\delta > 0$ (still independently of θ_1) so that it also satisfies

$$\liminf_{\theta_1\to 0^+}\inf_{x\in E}|\widehat{m}_{\theta,Z}(x)| \geq \frac{|y_j|}{2}. \tag{5.12}$$

We have

$$\begin{aligned}
\int_\Omega\left(\frac{\partial\widehat{m}_{\theta,Z}(x)}{\partial\chi_1}\right)^2 dx &\geq \int_{\prod_{r=1}^{D}[x_r^{(j)}-\delta,x_r^{(j)}+\delta]}\left(\frac{\partial\widehat{m}_{\theta,Z}(x)}{\partial\chi_1}\right)^2 dx \\
&= \int_{\prod_{r=2}^{D}[x_r^{(j)}-\delta,x_r^{(j)}+\delta]} dx_{-1}\int_{x_1^{(j)}-\delta}^{x_1^{(j)}+\delta} dx_1\left(\frac{\partial\widehat{m}_{\theta,Z}(x)}{\partial\chi_1}\right)^2 \\
&\geq \int_{\prod_{r=2}^{D}[x_r^{(j)}-\delta,x_r^{(j)}+\delta]} dx_{-1}\int_{x_1^{(j)}-\sqrt{\theta_1}}^{x_1^{(j)}} dx_1\left(\frac{\partial\widehat{m}_{\theta,Z}(x)}{\partial\chi_1}\right)^2 \\
\text{(Jensen:)} &\geq \int_{\prod_{r=2}^{D}[x_r^{(j)}-\delta,x_r^{(j)}+\delta]} dx_{-1}\sqrt{\theta_1}\left(\frac{1}{\sqrt{\theta_1}}\int_{x_1^{(j)}-\sqrt{\theta_1}}^{x_1^{(j)}} dx_1\frac{\partial\widehat{m}_{\theta,Z}(x)}{\partial\chi_1}\right)^2 \\
&\geq (2\delta)^{d-1}\frac{1}{\sqrt{\theta_1}}\left(\inf_{x\in E}|\widehat{m}_{\theta,Z}(x)| - \sup_{s\in[-\delta,\delta]^D;|s_1|=\sqrt{\theta_1}}\left|\widehat{m}_{\theta,Z}(x^{(j)}+s)\right|\right)^2 \\
&\underset{\theta_1\to 0^+}{\longrightarrow} \infty,
\end{aligned}$$

from (5.11) and (5.12). This concludes the proof. □

Proof of Proposition 5. Without loss of generality, we consider $i = 1$ in the proof. Let us consider the 2×2 submatrix of $k_\theta(X, X)$ obtained by extracting the lines and columns a, b, with a, b as in the condition iii) of the lemma. Then as $\theta_1 \to \infty$ this submatrix converges to the singular matrix $((1,1)^\top, (1,1)^\top)$. Hence, we have, as $\theta_1 \to \infty$, $|k_\theta(X, X)| \to 0$ (since $k_\theta(X, X)$ has components bounded in absolute value by 1). Hence, it is sufficient to show that $y^\top k_\theta(X, X)^{-1} y$ is bounded in order to conclude the proof.

Let X_{θ_1} be obtained from X by dividing its first column by θ_1 and by leaving the other columns unchanged. Let $x_{\theta_1,j}$ be the transpose of the line j of X_{θ_1}, for $j = 1, ..., n$. Let $\bar{\theta} = (1, \theta_{-1})$. Then, $y^t k_\theta(X, X)^{-1} y = y^t k_{\bar{\theta}}(X_{\theta_1}, X_{\theta_1})^{-1} y$.

We now use tools from the theory of RKHSs and refer to, e.g., [Wen04] for the definitions and properties of RKHSs used in the rest of the proof. Let $\mathcal{H}$ be the RKHS of $k_{\bar{\theta}}$. Let $\alpha_{\theta_1} = k_{\bar{\theta}}(X_{\theta_1}, X_{\theta_1})^{-1} y$. Then, $f_{\theta_1} : \mathbb{R}^D \to \mathbb{R}$ defined by $f_{\theta_1}(x) = \sum_{j=1}^n [\alpha_{\theta_1}]_j k_{\bar{\theta}}(x - x_{\theta_1,j})$ is the function of $\mathcal{H}$ with minimal RKHS norm $||.||_{\mathcal{H}}$ satisfying $f_{\bar{\theta_1}}(x_{\theta_1,j}) = y_j$ for $j = 1, ..., n$.

As $\theta_1 \to \infty$, the points $x_{\theta_1,1}, ..., x_{\theta_1,n}$ converge to the points $w_1, ..., w_n$ with $w_i = (0, v_i^\top)^\top$. We observe that, by assumption, $y_r = y_s$ for $w_r = w_s$. Hence, there exists $\epsilon > 0$ small enough and p column vectors $c_1, ..., c_p$ in $\mathbb{R}^D$ with the following properties: (i) each Euclidean ball with center c_m, $m = 1, ..., p$, and radius 2ϵ does not contain two w_r, w_s with $y_r \neq y_s$, $r, s \in \{1, ..., n\}$; (ii) each w_j, $j = 1, ..., n$, is contained in a ball with center c_m with $m \in \{1, ..., p\}$ and radius ϵ; (iii) the p balls with centers $c_1, ..., c_p$ and radii 2ϵ are two-by-two non-intersecting. We can also assume that each ball with center c_m, $m = 1, ..., p$ and radius ϵ contains at least one $w_{j(m)}$ with $j(m) \in \{1, ..., n\}$ and we write $z_m = y_{j(m)}$.

Then, from Lemma 3, there exists an infinitely differentiable function g with compact support on $\mathbb{R}^d$ so that for $m = 1, ..., p$, $g(x) = z_m$ for $||x - c_m|| \leq 2\epsilon$. Hence, for θ_1 large enough, the function g satisfies $g(x_{\theta_1,j}) = y_j$ for $j = 1, ..., n$.

Hence, $||f_{\theta_1}||_{\mathcal{H}} \leq ||g||_{\mathcal{H}}$ for θ_1 large enough, where $||g||_{\mathcal{H}}$ does not depend on θ_1. Finally, a simple manipulation of $||.||_{\mathcal{H}}$ (see again [Wen04] for the definitions), provides

$$\begin{aligned} ||f_{\theta_1}||_{\mathcal{H}} &= \sum_{r,s=1}^{n} \alpha_{\theta_1,r}\alpha_{\theta_1,s}k_{\bar{\theta}}(x_{\theta_1,r} - x_{\theta_1,s}) \\ &= y^\top k_\theta(X,X)^{-1}k_\theta(X,X)k_\theta(X,X)^{-1}y \\ &= y^\top k_\theta(X,X)^{-1}y. \end{aligned}$$

This concludes the proof. □

Proof of Proposition 6. Without loss of generality, we consider $i = 1$ in the proof. Also, up to renumbering the lines of X and components of y, it is sufficient to show that, for fixed $\theta_{-1} \in (0,\infty)^D$, as $\theta_1 \to \infty$, $\widehat{y}_{\theta,n} \to y_n$. We use the same notation $\bar{\theta}$, $\mathcal{H}$ and $x_{\theta_1,j}$ as in the proof of Proposition 5. Then, we have $\widehat{y}_{\theta,n} = f_{\theta_1}(x_{\theta_1,n})$, where $f_{\theta_1} \in \mathcal{H}$ is the function with minimal norm $||.||_{\mathcal{H}}$ satisfying $f_{\theta_1}(x_{\theta_1,j}) = y_j$ for $j = 1, ..., n-1$.

Furthermore, from the proof of Proposition 5, there exists a function $g \in \mathcal{H}$, not depending on θ_1 satisfying, for θ_1 large enough, $g(x_{\theta_1,j}) = y_j$ for $j = 1, .., n$. This shows that $||f_{\theta_1}||_{\mathcal{H}}$ is bounded as $\theta_1 \to \infty$. Let $m \in \{1, ..., n-1\}$ be so that $v_m = v^{(n)}$ (the existence is assumed in the condition iii)). Let also, for $x \in \mathbb{R}^D$, $k_{\bar{\theta},x} \in \mathcal{H}$ be the function $k_{\bar{\theta}}(x - .)$. Then we have (see again [Wen04]), with $(.,.)_{\mathcal{H}}$ the inner product in $\mathcal{H}$

$$\begin{aligned} |\widehat{y}_n - y_n| &= |f_{\theta_1}(x_{\theta_1,n}) - f_{\theta_1}(x_{\theta_1,m})| \\ &= \left|(f_{\theta_1}|k_{\bar{\theta},x_{\theta_1,n}})_{\mathcal{H}} - (f_{\theta_1}|k_{\bar{\theta},x_{\theta_1,m}})_{\mathcal{H}}\right| \\ &\leq ||f_{\theta_1}||_{\mathcal{H}}||k_{\bar{\theta},x_{\theta_1,n}} - k_{\bar{\theta},x_{\theta_1,m}}||_{\mathcal{H}} \\ &= ||f_{\theta_1}||_{\mathcal{H}}\sqrt{k_{\bar{\theta}}(x_{\theta_1,n} - x_{\theta_1,n}) + k_{\bar{\theta}}(x_{\theta_1,m} - x_{\theta_1,m}) - 2k_{\bar{\theta}}(x_{\theta_1,n} - x_{\theta_1,m})}. \end{aligned}$$

In the above display, the square root goes to zero as $\theta_1 \to \infty$ because $x_{\theta_1,n} - x_{\theta_1,m}$ goes to zero and $k_{\bar{\theta}}$ is continuous. This concludes the proof. □

Lemma 2. *For any $0 < \epsilon_1 < \epsilon_2 < \infty$, there exists an infinitely differentiable function $g : \mathbb{R} \to \mathbb{R}$ satisfying $g(u) = 1$ for $|u| \leq \epsilon_1$ and $g(u) = 0$ for $|u| \geq \epsilon_2$.*

Proof. Let $h : \mathbb{R} \to \mathbb{R}$ be defined by $h(t) = \exp(-1/(1-t^2))\mathbf{1}\{t \in [-1,1]\}$. Then h is

infinitely differentiable. Hence, g can be chosen of the form

$$g(t) = \begin{cases} A \int_{-\infty}^{t} h\left(B\left[u + \frac{\epsilon_1+\epsilon_2}{2}\right]\right) du & \text{if } t \leq 0 \\ A \int_{t}^{\infty} h\left(B\left[u - \frac{\epsilon_1+\epsilon_2}{2}\right]\right) du & \text{if } t \geq 0 \end{cases},$$

with $2/(\epsilon_2 - \epsilon_1) < B < \infty$ and $A = B/(\int_{-\infty}^{\infty} h(u)du)$. It can be checked that g is infinitely differentiable and satisfies the conditions of the lemma. □

Lemma 3. *Let $d, p \in \mathbb{N}$. Let $x^{(1)}, ..., x^{(p)}$ be two-by-two distinct points in $\mathbb{R}^D$ and $\epsilon > 0$ be so that the p closed Euclidean balls with centers x_i and radii ϵ are disjoint. Let $y_1, ..., y_p \in \mathbb{R}$ be arbitrary. Then there exists an infinitely differentiable function $r : \mathbb{R}^D \to \mathbb{R}$, with compact support, satisfying for $i = 1, ..., p$, $g(u) = y_i$ when $||u - x_i|| \leq \epsilon$.*

Proof. Let $l = \min_{i \neq j} ||x^{(i)} - x^{(j)}||$ and observe that $\epsilon < 2l$. Let g satisfies Lemma 2 with $\epsilon_1 = \epsilon^2$ and $\epsilon_2 = l^2/4$. Then the function r defined by $r(u) = \sum_{i=1}^{p} y_i g(||u - x_i||^2)$ satisfies the conditions of the lemma. □

5.7 Conclusion

Performing Bayesian optimization in high dimension is a difficult task. In several real-life problems, some variables are not influential. Therefore, we propose the so-called *Split-and-Doubt* algorithm that performs sequentially both dimension reduction and feature oriented sampling. The "split" step (model reduction) is based on a property of stationary ARD kernel of Gaussian process regression. We proved that large correlation lengths correspond to inactive variables. We also showed that classical estimators such ML and CV assign large correlation lengths to inactive variables.

The "doubt" step question the "split" step and helps correcting the estimation of the correlation lengths. It is possible to use this strategy for different feature learning purposes such as refinement, optimization and inversion. The optimization *Split-and-Doubt* algorithm has been evaluated on classical benchmark functions embedded in larger dimensional spaces by adding useless input variables. The results show that *Split-and-Doubt* is faster than classical EGO in the whole design space and outperforms it for most of the discussed tests.

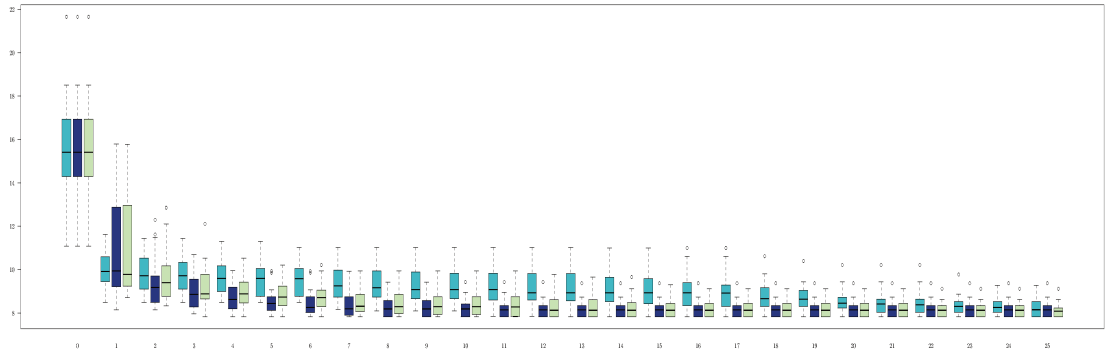


Figure 5.7: Borehole: Box plots convergence.

The main limitation of *Split-and-Doubt* is that we perform correlation length estimation in the whole design space. This computation is expensive. To overcome this problem, one can use fast maximum likelihood approximation techniques [DBDB13]. Future research may investigate such methods and extend *Split-and-Doubt* to constrained optimization.

Acknowledgments Malek Ben Salem is funded by a CIFRE grant from the ANSYS company, subsidized by the French National Association for Research and Technology (ANRT, CIFRE grant number 2014/1349). Part of this research was presented at the Chair in Applied Mathematics OQUAIDO, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, Ecole Centrale de Lyon, Mines Saint-Etienne, University of Grenoble, University of Nice, University of Toulouse) around advanced methods for Computer Experiments. We thank the participants for their feedback.

5.8 Appendix A: Optimization test results

In this section, we use box plots to display the evolution of the best value of the optimization test bench. For each iteration, we display: Left: EGO in light blue, Middle: *Split-and-Doubt* in dark blue, Right: *Split-Without-Doubt* in light green.

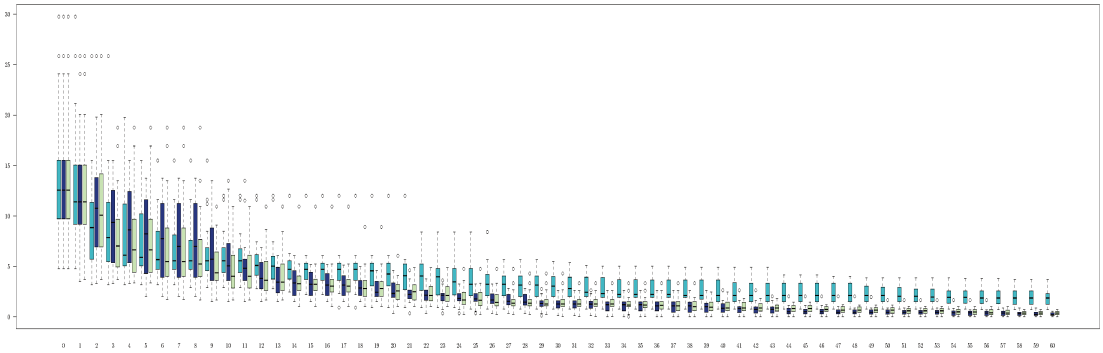


Figure 5.8: Rosenbrock: Box plots convergence.

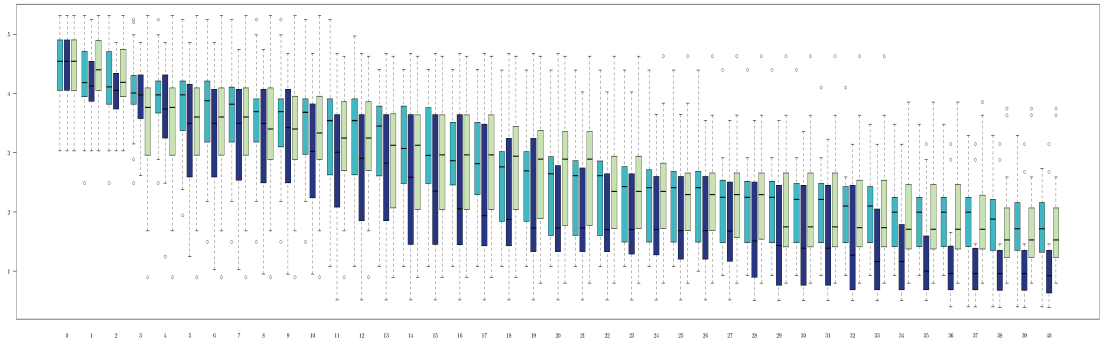


Figure 5.9: Ackley: Box plots convergence.

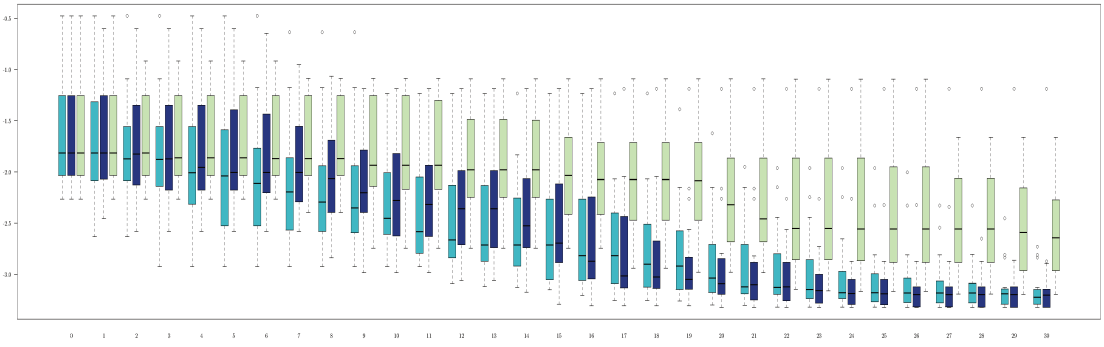


Figure 5.10: Hartmann 6-dim: Box plots convergence.

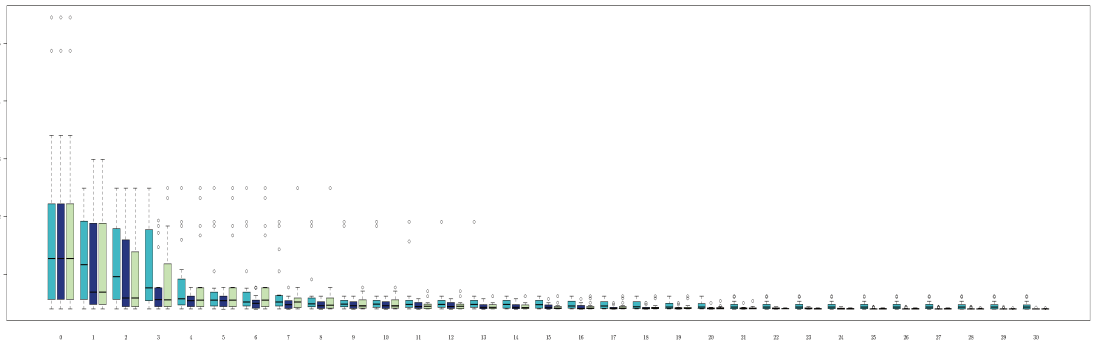


Figure 5.11: Branin: Box plots convergence.

Conclusion and future works

Chapter 6

Conclusion and future works

6.1 Conclusion

The need to efficiently explore the space of simulation-based designs has motivated our thesis work. Essentially, we deal with surrogate models based strategies. With regards to current limitations and motivated by industrial needs, we tackled three aspects of surrogate-modeling. A summary of the main contributions and some ideas for enhancement and future research are given below.

- **Automatic selection:** In this work, we proposed and studied the so called penalized predictive score (PPS). It favours the selection of surrogates with regularity and smoothness properties. Further, it is suitable for the construction of ensemble of surrogates. Exploiting these properties, we presented two surrogate model selection algorithms. The first one constructs the optimal ensemble with regard to the PPS. The second one further explores the set of surrogate modeling techniques using a genetic algorithm. These algorithms were evaluated on a benchmark of 15 test functions. The results highlight the efficiency of both approaches.

  To improve this framework, we may first look for a flexible weighting method (instead of using constant weights) of the components of the PPS by incorporating expert-based physical characteristics of a given function. Second, the use of local weights in ensemble remains both challenging and promising. In this context, adding this additional degree of freedom may help enhancing prediction accuracy

but would require a relevant regularization in order to avoid over-fitting.

- **Universal Prediction distribution:** The popularity of Gaussian process regression is mainly due to the provided prediction distribution. Meanwhile, there is a multitude of surrogate modeling techniques and they do not all provide an uncertainty quantification tool. In this work, we gave a universal method for uncertainty quantification that could be applied for any surrogate model.

  It is based on a weighted empirical probability measure supported by cross-validation sub-models predictions. Consequently, one may use this distribution to compute most of the classical sequential sampling criteria. We also discussed sequential design strategies for prediction refinement, optimization and inversion. Further, we showed that, under some assumptions, the optimum is adherent to the sequence of points generated by the optimization algorithm UP-EGO. Moreover, the optimization and the refinement algorithms were successfully implemented and experienced both on single and multiple surrogate models. Software development containing UP tools have been implemented and an R package is available.

  As a perspective, the UP-distribution can be extended to compute empirical spatial covariances between two locations. More generally, the UP distribution might be enhanced by studying the links with empirical Bayesian methods: Can the UP distribution be seen as an a posteriori distribution of some process? It is also interesting to study the asymptotic properties of the UP distribution for specific surrogate models, when the number of observations tends to infinity.

- **Sequential design in high dimensions:** In this work, we proposed the so-called *Split-and-Doubt* algorithm that performs sequentially both dimension reduction and feature oriented sampling. The "split" step (model reduction) is based on a property of stationary ARD kernel of Gaussian process regression. Indeed, we proved that large correlation lengths correspond to inactive variables. We also showed that classical estimators such as ML and CV assign large correlation lengths to inactive variables.

  In the "doubt" step, we question the "split" step in order to help correcting the estimation of the correlation lengths. The two-step approach aims at performing

both feature learning and dimension reduction. Note that we can use this strategy for different features such as prediction refinement, optimization and inversion. An optimization version of *Split-and-Doubt* algorithm has been evaluated on classical benchmark functions embedded in larger dimensional spaces. The results show that *Split-and-Doubt* is faster than classical EGO in the whole design space and outperforms it for most of the discussed tests.

A relevant generalization of the *Split-and-Doubt* for multi-objective or constrained optimization remains challenging. Eventually, it would be better to assess the influence of each variable on each output function separately. That is, one variable can be influential for a given constraint and inactive for a given output and vice versa. The main challenge is to adapt the sampling criteria to this context.

Appendix A

Résumés des chapitres en français

A.1 Introduction

En recherche comme en ingénierie de conception, les simulations numériques sont devenues populaires. En effet, elles offrent plusieurs avantages en les comparant à la réalisation d'une expérience notamment en termes de rapidité et de coût. Dans certaines études, il est impossible de réaliser une expérience (étude du climat, tremblement de terre, conception d'un profile -aéronautique-) d'où l'indispensabilité du recours au simulations numériques.

La concurrence et les normes de plus en plus pointues stimulent le besoin des nouveaux modèles plus efficaces, plus robustes et plus optimisées. Par conséquent, les simulations ne sont plus seulement utilisées pour valider une conception. Mais, elles sont également utilisées pour explorer l'espace de conception à la recherche de nouveaux modèles avec des performances optimales. L'exploration et l'optimisation nécessitent en général de nombreuses évaluations du simulateur. Cependant, les simulations haute-fidélités de modèles complexes restent coûteuses en termes de calcul malgré l'évolution du calcul haute performance.

Pour surmonter ce coût, des modèles de substitution, également appelés méta-modèles ou surfaces de réponse, sont utilisés pour accélérer l'exploration de l'espace de conception. Ces fonctions visent à émuler la véritable fonction, ici le simulateur à calcul intensif, tout en étant moins coûteux en calcul. Les modèles de substitution sont couramment utilisés

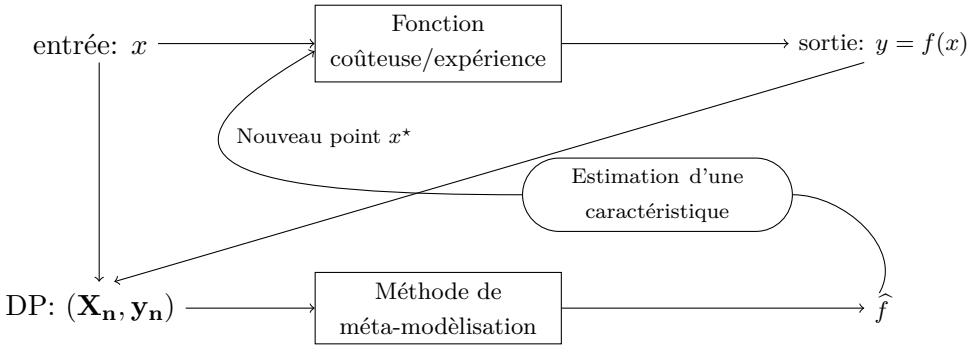


Figure A.1: Illustration de la méta-modélisation

dans la conception technique [Kle08, SWN13] et il existe de nombreuses méthodes de construction de telles approximations [Mat69, LS81, SS04, PG89].

Les méta-modèles sont basés sur n observations $Z_n = (z_1, \ldots, z_n)$, où $z_j = (x_j, y_j)$ pour $1 \leq j \leq n$ et $y_j = f(x_j)$, aussi appelé plan d'expérience. L'objectif principal de la méta-modélisation est de remplacer une fonction coûteuse f par une surface de réponse $\widehat{f}_{Z_n}$. Parfois, cette approximation $\widehat{f}_{Z_n}$ est utilisée pour accélérer l'estimation d'une caractéristique de la fonction f. La précision des modèles de remplacement s'appuie, entre autres, sur la pertinence du plan d'expérience. Par conséquent, l'échantillonnage des $(x_j)_{1 \leq j \leq n}$ est crucial. On présente dans la figure A.1, un schéma illustrant la méta-modélisation.

Les contributions de cette thèse traitent principalement trois aspects de la méta-modélisation : la sélection du méta-modèle, l'échantillonnage séquentiel pour un méta-modèle quelconque et l'échantillonnage séquentiel en grande dimension. Pour présenter ces contributions le document est présenté comme suit :

- Dans la Partie I, on introduit le contexte générale de ce travail dans (Chapitre 1). Le chapitre 2, quant à lui, présente brièvement les notions et l'état de l'art nécessaires pour bien situer nos contributions.

- Nos mettons deux articles sur le thème de la méta-modélisation dans La Partie II. Le premier présente deux algorithmes de sélection de méta-modèles (Chapitre 3).

Le deuxième présente une méthode universelle qui permet d'associer une incertitude à la prédiction de tout méta-modèle.

- La Partie III contient une contribution sur l'échantillonnage séquentielle qui permet de faire conjointement l'estimation d'une caractéristique d'une fonction et la réduction de dimension.

Les Chapitres 3, 4, 5 sont une reproduction des articles suivant:

- M. Ben Salem and L. Tomaso. Automatic selection for general surrogate models. *Structural and Multidisciplinary Optimization*, Feb 2018 (Chapitre 3).

- M. Ben Salem, O. Roustant, F. Gamboa, and L. Tomaso. Universal prediction distribution for surrogate models. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):1086–1109, 2017 (Chapitre 4).

- M. Ben Salem, F. Bachoc, O. Roustant, F. Gamboa, and L. Tomaso. Sequential dimension reduction for learning features of expensive black-box functions. *Preprint available at hal-01688329*, 2017 (Chapitre 5).

A.2 État de l'art

Le chapitre 2 donne le contexte nécessaire pour définir nos contributions. Nous présentons brièvement plusieurs techniques classiques de méta-modélisation ainsi que différentes techniques d'évaluation de la qualité de ces méta-modèles.

Une attention particulière est dédiée à la régression par processus gaussien (GP): Nous discutons de l'estimation des paramètres des noyaux et comment l'incertitude de prédiction permet de définir des critères pertinents d'échantillonnage séquentielle. La dernière partie du chapitre est consacrée aux techniques d'échantillonnage des plans d'expérience. On intègre dans cette définition les techniques adaptatifs de la planification séquentielle, y inclut, celle qui vise à l'estimation d'une fonctionnalité, telle que les schémas d'optimisation.

A.3 Sélection de modèles de remplacement

Dans le chapitre 3, nous traitons le problème de sélection de surface de réponse. En effet, il existe de nombreux types de méta-modèle et pour chaque type, il existe différents réglages possibles. D'abord, il n'existe pas un méta-modèle optimal pour tous les problèmes. Ensuite, il est difficile de choisir les modèles de substitution les plus appropriés pour un plan d'expériences donné. De surcroît, il est difficile d'évaluer la qualité d'un méta-modèle sans des données supplémentaires de vérification.

Nous proposons un critère de sélection qui évalue la qualité des modèles de substitution. Nous l'appelons le score prédictif pénalisé (PPS). On peut calculer le PPS pour tous les méta-modèles. Par construction, le PPS convient particulièrement aux fonctions de réponse qui ont des caractéristiques de régularité et de douceur. En général, ces caractéristiques sont implicitement attendues dans le cadre de la méta-modélisation. Nous montrons que PPS permet la construction d'une agrégation pertinente de méta-modèles. Les poids PPS-optimaux de ces agrégations permettent d'éviter le sur-ajustement. Ils sont, par ailleurs, faciles à optimiser, parce qu'on présente une formule directe pour le calcul de ces poids optimaux. Nous présentons également deux schémas de sélection de méta-modèle basés sur le score. Le premier calcule l'ensemble PPS-optimal plutôt que de sélectionner un modèle de substitution. Le second est basé sur un cadre évolutif qui permet l'exploration de l'espace des modèles de substitution.

A.4 Prédiction universelle de l'erreur

Le chapitre 4 donne un nouvel outil pour associer une distribution de prédiction à n'importe quel modèle de substitution et, par conséquent, pour étendre les méthodes de conception séquentielle basées sur les processus gaussiens (PG) à n'importe quel modèle de substitution. Rappelons que le principal avantage de l'approche basée sur les PG est qu'elle fournit partout une mesure de l'incertitude associée à la prédiction du modèle de substitution. Cette incertitude est un outil efficace pour construire des stratégies pour divers problèmes tels que l'amélioration de la prédiction, l'optimisation ou l'inversion.

Dans ce chapitre, nous proposons une méthode universelle pour définir une mesure d'incertitude adaptée à tout modèle de substitution. Elle s'appuie sur des prédictions

de sous-modèles de validation croisée (CV) et conduit à une mesure empirique locale quantifiant localement l'incertitude de la prédiction. Cette distribution, appelée, la distribution de prédiction universelle (*UP distribution*), permet la définition de nombreux critères d'échantillonnage. Nous donnons et étudions des techniques d'échantillonnage adaptatif pour améliorer la précision de la prédiction et une extension de l'algorithme EGO (Efficient Global Optimization). Nous discutons aussi de l'utilisation de *UP distribution* pour les problèmes d'inversion.

A.5 Réduction de dimension et estimation de caractéristiques

De nos jours, de nombreux problèmes de conception sont complexes et peuvent impliquer un grand nombre de variables. L'exploration de l'espace de conception en grande dimension est une tâche difficile. Dans plusieurs cas industriels, certaines variables ne sont presque pas influentes. Le chapitre 5 présente un algorithme pour l'apprentissage d'une caractéristique de la fonction étudiée et la réduction de dimension. La méthode est basée sur la régression du processus gaussien. Notre méthode est appelée l'algorithme *split-and-doubt*. L'étape «split» (réduction du modèle) est basée sur une propriété des noyaux de détermination automatique de la pertinence stationnaire de la régression du processus gaussien. Nous montrons que les grandes longueurs de corrélation correspondent à des variables inactives. Nous montrons également que les estimateurs classiques tels que le maximum de vraisemblance et la validation croisée assignent des longueurs de corrélation importantes aux variables inactives.

L'étape «doute» remet en question l'étape «split» et aide à corriger une estimation erronée des longueurs de corrélation. Il est possible d'utiliser cette stratégie pour différents objectifs d'apprentissage, tels que le raffinement, l'optimisation ou l'inversion. L'algorithme d'optimisation *Split-and-Doubt* a été évalué sur des fonctions classiques plongées dans des espaces de plus grande dimension en ajoutant des variables d'entrée inutiles. Les résultats montrent que *Split-and-Doubt* est plus rapide que l'EGO classique dans l'ensemble de l'espace de conception et le surpasse pour la plupart des cas de test considérés.

A.6 Conclusion

Notre travail a été motivé par la nécessité d'une exploration plus efficace de l'espace de conception. Essentiellement, nous traitons des stratégies fondées sur des modèles de substitution. En considérant les limitations actuelles et motivés par les besoins industriels, nous avons abordé trois aspects de la méta-modélisation. Un résumé des principales contributions et quelques idées pour l'amélioration et la recherche future sont donnés ci-dessous.

- Dans ce travail, nous avons proposé et étudié ce qu'on appelle le score prédictif pénalisé (PPS). Il favorise la sélection de méta-modèles qui ont des propriétés de régularité et de douceur. En outre, il convient à la construction d'une agrégation de méta-modèles. En se basant sur ces propriétés, nous avons présenté deux algorithmes de sélection de modèles de substitution. Le premier construit l'ensemble PPS-optimal pour un ensemble fini (et modéré) de méta-modèles. Le second explore, d'avantage, l'ensemble des techniques de modélisation par substitution utilisant un algorithme génétique. Ces algorithmes ont été évalués sur un benchmark de 15 fonctions de test. Les résultats mettent en évidence l'efficacité des deux approches.

  Pour améliorer ce cadre, nous pouvons d'abord chercher une méthode de pondération flexible des composantes du PPS en incorporant des caractéristiques physiques expertes d'une fonction donnée. Deuxièmement, l'utilisation des poids locaux dans l'ensemble reste à la fois difficile et prometteuse. Dans ce contexte, l'ajout de ce degré de liberté supplémentaire peut aider à améliorer la précision de la prévision, nécessiterait une régularisation appropriée afin d'éviter un sur-ajustement excessif.

- La popularité de la régression du processus gaussien est principalement due à la distribution de prédiction qu'elle fournit. Au même temps, plusieurs techniques de méta-modélisation ne fournissent pas toutes un outil de quantification de l'incertitude. Dans ce travail, nous avons donné une méthode universelle de quantification de l'incertitude qui pourrait être appliquée à tout modèle de substitution. Elle est basée sur une mesure de probabilité empirique pondérée, où le support est les prédictions de sous-modèles de validation croisée. On peut adapter la

plupart des critères d'échantillonnage séquentiels classiques. Nous avons également discuté des stratégies de conception séquentielle pour le raffinement de la prédiction, l'optimisation et l'inversion. De plus, nous avons montré que, sous certaines hypothèses, l'optimum est adhèrent à la séquence de points générés par l'algorithme d'optimisation UP-EGO. De plus, les algorithmes d'optimisation et de raffinement ont été implémentés et testés avec succès. Un package R contenant des outils UP est disponible.

En perspective, la distribution UP peut être étendue pour calculer des covariances spatiales empiriques entre deux localisations : La distribution UP peut-elle être vue comme une distribution a posteriori d'un processus ? Il est également intéressant d'étudier les propriétés asymptotiques de la distribution UP pour des modèles de substitution spécifiques, lorsque le nombre d'observations tend vers l'infini.

- Dans ce travail, nous avons proposé l'algorithme appelé *Split-and-Doubt* qui effectue conjointement la réduction de dimension et l'apprentissage d'une caractéristique de la fonction étudiée. L'étape "split" (réduction du modèle) est basée sur une propriété des noyau stationnaire ARD de la régression par processus gaussien. En effet, nous avons démontré que les grandes longueurs de corrélation correspondent à des variables inactives. Nous avons également montré que les estimateurs classiques tels que ML et CV assignent des grandes longueurs de corrélation aux variables inactives. Dans l'étape "doute'', nous remettons en cause l'étape "split'' afin de corriger une estimation éventuellement erronée des longueurs de corrélation. L'approche en deux étapes vise à effectuer à la fois l'apprentissage de la caractéristique et la réduction de dimension. On peut utiliser cette stratégie pour différents objectifs telles que le raffinement, l'optimisation et l'inversion. Une version d'optimisation de l'algorithme *Split-and-Doubt* a été évaluée sur des fonctions classiques plongées dans des espaces de plus grande dimension. Les résultats montrent que *Split-and-Doubt* est plus rapide que l'EGO classique dans l'ensemble de l'espace de conception et le surpasse pour la plupart des tests décrits.

Une généralisation pertinente du *Split-and-Doubt* pour l'optimisation multi-objectif ou contrainte reste difficile. Il serait préférable d'évaluer l'influence de chaque variable sur chaque fonction de sortie séparément. C'est-à-dire qu'une variable peut

être influente pour une contrainte donnée et inactive pour une sortie donnée et vice versa. Le principal défi consiste à adapter les critères d'échantillonnage à ce contexte.

Bibliography

[ABDJ+00] C. Audet, A. J. Booker, J. Dennis Jr, P. D. Frank, and D. W. Moore. A surrogate-model-based method for constrained optimization. In *8th Symposium on Multidisciplinary Analysis and Optimization*, volume 4891, 2000.

[Abr97] P. Abrahamsen. *A review of Gaussian random fields and correlation functions*. Norsk Regnesentral/Norwegian Computing Center, 1997.

[AC10] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[Aka74] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, December 1974.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[ARR09] E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization*, 37(3):279–294, 2009.

[ASA+13] V. Aute, K. Saleh, O. Abdelaziz, S. Azarm, and R. Radermacher. Cross-validation based single response adaptive design of experiments for kriging metamodeling of deterministic computer simulations. *Structural and Multidisciplinary Optimization*, 48(3):581–605, 2013.

[AT09] R. J. Adler and J. E. Taylor. *Random fields and geometry*. Springer Science & Business Media, 2009.

[Bac13a] F. Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66(Supplement C):55 – 69, 2013.

[Bac13b] F. Bachoc. *Estimation paramétrique de la fonction de covariance dans le modèle de Krigeage par processus Gaussiens: application à la quantification des incertitues en simulation numérique*. PhD thesis, Paris 7, 2013.

[Bac14] F. Bachoc. Asymptotic analysis of the role of spatial sampling for covariance parameter estimation of Gaussian processes. *Journal of Multivariate Analysis*, 125(Supplement C):1 – 35, 2014.

[BB60] G. E. Box and D. W. Behnken. Some new three level designs for the study of quantitative variables. *Technometrics*, 2(4):455–475, 1960.

[BBV11] R. Benassi, J. Bect, and E. Vazquez. Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion. *LION*, 5:176–190, 2011.

[BC64] G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[BD87] G. E. Box and N. R. Draper. *Empirical model-building and response surfaces.*, volume 424. John Wiley & Sons, 1987.

[BES+08] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, 46(10):2459–2468, 2008.

[BFI07] D. Busby, C. L. Farmer, and A. Iske. Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing*, 29(1):49–69, 2007.

[BGL+12] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

[BGR15a] M. Binois, D. Ginsbourger, and O. Roustant. Quantifying uncertainty on pareto fronts with Gaussian process conditional simulations. *European Journal of Operational Research*, 243(2):386–394, 2015.

[BGR15b] M. Binois, D. Ginsbourger, and O. Roustant. A warped kernel improving robustness in Bayesian optimization via random embeddings. In *9th International Conference on Learning and Intelligent Optimization*, pages 281–286, Cham, 2015. Springer International Publishing.

[BHH78] G. E. Box, W. G. Hunter, and J. S. Hunter. *Statistics for experimenters: an introduction to design, data analysis, and model building*, volume 1. JSTOR, 1978.

[BLN17] F. Bachoc, A. Lagnoux, and T. M. N. Nguyen. Cross-validation estimation of covariance parameters under fixed-domain asymptotics. *Journal of Multivariate Analysis*, 2017.

[BSBR+17] M. Ben Salem, F. Bachoc, O. Roustant, F. Gamboa, and L. Tomaso. Sequential dimension reduction for learning features of expensive black-box functions. *Preprint available at hal-01688329*, 2017.

[BSRGT17] M. Ben Salem, O. Roustant, F. Gamboa, and L. Tomaso. Universal prediction distribution for surrogate models. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):1086–1109, 2017.

[BST18] M. Ben Salem and L. Tomaso. Automatic selection for general surrogate models. *Structural and Multidisciplinary Optimization*, Feb 2018.

[BTA11] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[BW92] G. E. Box and K. B. Wilson. On the experimental attainment of optimum conditions. In *Breakthroughs in Statistics*, pages 270–310. Springer, 1992.

[CBG+14] C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.

[CGE14] C. Chevalier, D. Ginsbourger, and X. Emery. *Corrected Kriging Update Formulae for Batch-Sequential Data Assimilation*, pages 119–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[Che53] H. Chernoff. Locally optimal designs for estimating parameters. *The Annals of Mathematical Statistics*, pages 586–602, 1953.

[CKC12] B. Chen, A. Krause, and R. M. Castro. Joint optimization and variable selection of high-dimensional Gaussian processes. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1423–1430, New York, NY, USA, 2012. ACM.

[CPG14] C. Chevalier, V. Picheny, and D. Ginsbourger. Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational statistics & data analysis*, 71(0):1021–1034, 2014.

[Cre93] N. A. Cressie. *Statistics for spatial data*. John Wiley & Sons, 1993.

[CWL04] P.-W. Chen, J.-Y. Wang, and H.-M. Lee. Model selection of svms using ga approach. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 3, pages 2035–2040. IEEE, 2004.

[DBDB13] J. H. De Baar, R. P. Dwight, and H. Bijl. Speeding up kriging through fast estimation of the hyperparameters in the frequency-domain. *Computers & Geosciences*, 54:99–106, 2013.

[DHF15] D. Dupuy, C. Helbert, and J. Franco. DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38, 2015.

[Dri73] M. F. Driscoll. The reproducing kernel hilbert space structure of the sample paths of a Gaussian process. *Probability Theory and Related Fields*, 26(4):309–316, 1973.

[DS78] L. W. Dixon and G. P. Szegö. *Towards global optimisation 2*. North-Holland Amsterdam, 1978.

[DSB11] V. Dubourg, B. Sudret, and J.-M. Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673–690, 2011.

[Dub83] O. Dubrule. Cross validation of kriging in a unique neighborhood. *Mathematical Geology*, 15(6):687–699, 1983.

[Duc77] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[DW79] L. P. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *Information Theory, IEEE Transactions on*, 25(5):601–604, Sep 1979.

[EDK11] M. T. Emmerich, A. H. Deutz, and J. W. Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE, 2011.

[ET93] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

[Fed72] V. V. Fedorov. *Theory of optimal experiments*. Elsevier, 1972.

[FJ08] A. I. Forrester and D. R. Jones. Global optimization of deceptive functions with sparse sampling. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, volume 1012, 2008.

[FK09] A. I. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

[Fri91] J. H. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

[Gau09] C. F. Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. sumtibus Frid. Perthes et IH Besser, 1809.

[GDT09] D. Gorissen, T. Dhaene, and F. D. Turck. Evolutionary model type selection for global surrogate modeling. *Journal of Machine Learning Research*, 10(Sep):2039–2078, 2009.

[Gei75] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[GHQS06] T. Goel, R. T. Haftka, N. V. Queipo, and W. Shyy. Performance estimate and simultaneous application of multiple surrogates. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006.

[GHSQ07] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2007.

[GMDO08] S. Gazut, J.-M. Martinez, G. Dreyfus, and Y. Oussar. Towards the optimal design of numerical experiments. *IEEE transactions on neural networks*, 19(5):874–882, May 2008.

[GS03] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[GWE03] A. A. Giunta, S. F. Wojtkiewicz, and M. S. Eldred. Overview of modern design of experiments methods for computational simulations. In *Proceedings of the 41st AIAA aerospace sciences meeting and exhibit, AIAA-2003-0649*, 2003.

[HDC09] C. Helbert, D. Dupuy, and L. Carraro. Assessment of uncertainty in computer experiments from universal to Bayesian kriging. *Applied Stochastic Models in Business and Industry*, 25(2):99–113, 2009.

[HHB78] W. G. Hunter, J. S. Hunter, and G. E. Box. *Statistics for experimenters: an introduction to design, data analysis, and model building*. Wiley New York, 1978.

[HHLB11] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *LION*, 5:507–523, 2011.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.

[Hub64] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[IL15] B. Iooss and P. Lemaître. A review on global sensitivity analysis methods. In *Uncertainty Management in Simulation-Optimization of Complex Systems*, pages 101–122. Springer, 2015.

[JCS02] R. Jin, W. Chen, and A. Sudjianto. On sequential sampling for global metamodeling in engineering design. In *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 539–548. American Society of Mechanical Engineers, 2002.

[JMY90] M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.

[Jon01] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[JSW98] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[JWHT13] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 6. Springer, 2013.

[Kle08] J. P. Kleijnen. *Design and analysis of simulation experiments*, volume 20. Springer, 2008.

[Kle09] J. P. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143. Morgan Kaufmann Publishers Inc., 1995.

[Kri51] D. G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52:119–139, 1951.

[KvB04] J. P. Kleijnen and W. C. van Beers. Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society*, 55(8):876–883, 2004.

[KvBvN12] J. P. Kleijnen, W. van Beers, and I. van Nieuwenhuyse. Expected improvement in efficient global optimization through bootstrapped kriging. *Journal of Global Optimization*, 54(1):59–73, 2012.

[LA06] G. Li and S. Azarm. Maximum accumulative error sampling strategy for approximation of deterministic engineering simulations. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006.

[LAFMD06] G. Li, S. Azarm, A. Farhang-Mehr, and A. R. Diaz. Approximation of multiresponse deterministic engineering simulations: a dependent metamodeling approach. *Structural and Multidisciplinary Optimization*, 31(4):260–269, 2006.

[Leg05] A. M. Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.

[Lin56] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

[Lip87] R. Lippmann. An introduction to computing with neural nets. *IEEE Assp magazine*, 4(2):4–22, 1987.

[LMA+04] Y. Lin, F. Mistree, J. K. Allen, K.-L. Tsui, and V. Chen. Sequential metamodeling in engineering design. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[LS81] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155):141–158, 1981.

[LSC06] S. Lessmann, R. Stahlbock, and S. F. Crone. Genetic algorithms for support vector machine model selection. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 3063–3069. IEEE, 2006.

[LSW09] J. L. Loeppky, J. Sacks, and W. J. Welch. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4), 2009.

[Mat63] G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[Mat69] G. Matheron. Le krigeage universel. *Cahiers du Centre de Morphologie Mathématique*, 1, 1969.

[MBC79] M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[MM95] M. D. Morris and T. J. Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381–402, 1995.

[MMAC16] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook. *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016.

[MMY93] M. D. Morris, T. J. Mitchell, and D. Ylvisaker. Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, 35(3):243–255, 1993.

[Moč75] J. Močkus. *On Bayesian Methods for Seeking the Extremum*, pages 400–404. Springer Berlin Heidelberg, 1975.

[Moč82] J. Močkus. The Bayesian approach to global optimization. *System Modeling and Optimization*, pages 473–481, 1982.

[MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[MP69] M. L. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT, 1969.

[MP11] J. Müller and R. Piché. Mixture surrogate models based on dempster-shafer theory for global optimization problems. *Journal of Global Optimization*, 51(1):79–104, 2011.

[MR85] C. A. Micchelli and T. J. Rivlin. Lectures on optimal recovery. In P. R. Turner, editor, *Numerical Analysis Lancaster 1984: Proceedings of the*

SERC Summer School held in Lancaster, England, Jul. 15 – Aug. 3, 1984, pages 21–93, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[NCK+11] H. M. Nguyen, I. Couckuyt, L. Knockaert, T. Dhaene, D. Gorissen, and Y. Saeys. An alternative approach to avoid overfitting for surrogate models. In *Proceedings of the Winter Simulation Conference*, pages 2765–2776, Dec. 2011.

[NM81] Y. Nakagawa and S. Miyazaki. Surrogate constraints algorithm for reliability optimization problems with two constraints. *IEEE Transactions on Reliability*, 30(2):175–180, 1981.

[ONL06] Y.-S. Ong, P. B. Nair, and K. Y. Lum. Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation*, 10(4):392–404, 2006.

[Owe92] A. B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, pages 439–452, 1992.

[Owe94] A. B. Owen. Lattice sampling revisited: Monte carlo variance of means over randomized orthogonal arrays. *The Annals of Statistics*, pages 930–945, 1994.

[PG89] T. Poggio and F. Girosi. A theory of networks for approximation and learning. *Laboratory, Massachusetts Institute of Technology*, 1140, 1989.

[PGR+10] V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N.-H. Kim. Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7):071008, 2010.

[PH16] W. A. Pruett and R. L. Hester. The creation of surrogate models for fast estimation of complex model outcomes. *PloS one*, 11(6):e0156574, 2016.

[Pic14] V. Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial Intelligence and Statistics*, pages 787–795, 2014.

[Pic15] V. Picheny. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.

[PM12] L. Pronzato and W. G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[PW85] L. Pronzato and É. Walter. Robust experiment design via stochastic approximation. *Mathematical Biosciences*, 75(1):103–120, 1985.

[QHS+05] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005.

[RBM08] P. Ranjan, D. Bingham, and G. Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4), 2008.

[RFIK14] O. Roustant, J. Fruth, B. Iooss, and S. Kuhnt. Crossed-derivative based sensitivity measures for interaction screening. *Mathematics and Computers in Simulation*, 105:105–118, 2014.

[RGD12] O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software, Articles*, 51(1):1–55, 2012.

[Rip05] B. D. Ripley. *Spatial statistics*, volume 575. John Wiley & Sons, 2005.

[RW06] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[Sas98] M. J. Sasena. *Optimization of computer simulations via smoothing splines and kriging metamodels*. PhD thesis, University of Michigan, 1998.

[Sas02] M. J. Sasena. *Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations*. PhD thesis, University of Michigan Ann Arbor, MI, 2002.

[Sch78] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[SG95] I. M. Sobol' and A. Gershman. On an alternative global sensitivity estimator. In *Proceedings of SAMO 1995, Belgirate, Italy*, pages 40–42. SAMO, 1995.

[SK08] T. Shao and S. Krishnamurty. A clustering-based surrogate model updating approach to simulation-based engineering design. *Journal of Mechanical Design*, 130(4):041101, 2008.

[SK09] I. M. Sobol' and S. Kucherenko. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, 79(10):3009 – 3017, 2009.

[SPG00] M. J. Sasena, P. Y. Papalambros, and P. Goovaerts. Metamodeling sampling criteria in a global optimization framework. In *8th Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, 2000.

[SQMC10] L. V. Santana-Quintero, A. A. Montano, and C. A. Coello. A review of techniques for handling expensive functions in evolutionary multi-objective optimization. In *Computational intelligence in expensive optimization problems*, pages 29–59. Springer, 2010.

[SS04] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[SSWB00] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[Ste12] M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, New York, 2012.

[Sto74] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974.

[SV99] S. Streltsov and P. Vakili. A non-myopic utility function for statistical global optimization algorithms. *Journal of Global Optimization*, 14(3):283–298, 1999.

[SW87] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[SWN13] T. J. Santner, B. J. Williams, and W. I. Notz. *The design and analysis of computer experiments*. Springer Science & Business Media, 2013.

[SYZ12] L. Shi, R. Yang, and P. Zhu. A method for selecting surrogate models in crashworthiness optimization. *Structural and Multidisciplinary Optimization*, 46(2):159–170, 2012.

[TNE07] S. Tomioka, S. Nisiyama, and T. Enoto. Nonlinear least square regression by adaptive domain method with multiple genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 11(1):1–16, 2007.

[TSGB16] R. Troian, K. Shimoyama, F. Gillot, and S. Besset. Methodology for the design of the geometry of a cavity and its absorption coefficients as random design variables under vibroacoustic criteria. *Journal of Computational Acoustics*, 24(02):1650006, 2016.

[Tuk77] J. W. Tukey. *Exploratory data analysis*. Reading, Mass., 1977.

[Vap13] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[VB09] E. Vazquez and J. Bect. A sequential Bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes*, 42(10):546 – 550, 2009. 15th IFAC Symposium on System Identification.

[VB10] E. Vazquez and J. Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095, 2010.

[VGH10] F. A. Viana, C. Gogu, and R. T. Haftka. Making the most out of surrogate models: Tricks of the trade. In *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 587–598. American Society of Mechanical Engineers, 2010.

[VHS09] F. A. Viana, R. T. Haftka, and V. Steffen. Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Structural and Multidisciplinary Optimization*, 39(4):439–457, 2009.

[VHW13] F. A. Viana, R. T. Haftka, and L. T. Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization*, 56(2):669–689, 2013.

[VPM07] E. Vazquez and M. Piera-Martinez. Estimation du volume des ensembles d'excursion d'un processus gaussien par krigeage intrinsèque. In *Conférence Journée de Statistiques*, pages CD–ROM, 2007.

[VVB10] F. A. Viana, G. Venter, and V. Balabanov. An algorithm for fast optimal latin hypercube design of experiments. *International journal for numerical methods in engineering*, 82(2):135–156, 2010.

[VVW09] J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.

[Wat81] D. F. Watson. Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *The computer journal*, 24(2):167–172, 1981.

[Wen04] H. Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

[Wer74] P. J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.

[WHZ+16] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[WP90] É. Walter and L. Pronzato. Qualitative and quantitative experiment design for phenomenological models—a survey. *Automatica*, 26(2):195–213, 1990.

[WS07] G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, 2007.

[XLWJ14] S. Xu, H. Liu, X. Wang, and X. Jiang. A robust error-pursuing sequential sampling approach for global metamodeling based on voronoi diagram and cross validation. *AMSE. Journal of Mechanical Design*, 136(7):071009, 2014.

[XS17] W. Xu and M. L. Stein. Maximum likelihood estimation for a smooth Gaussian random field model. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):138–175, 2017.

[ZJ16] X. Zhou and T. Jiang. Metamodel selection based on stepwise regression. *Structural and Multidisciplinary Optimization*, 54(3):641–657, 2016.

[ZML11] X. J. Zhou, Y. Z. Ma, and X. F. Li. Ensemble of surrogates with recursive arithmetic average. *Structural and Multidisciplinary Optimization*, 44(5):651–671, 2011.

[ZQPS05] L. E. Zerpa, N. V. Queipo, S. Pintos, and J.-L. Salager. An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. *Journal of Petroleum Science and Engineering*, 47(3):197–208, 2005.

[ZSL00] C. Zhang, H. Shao, and Y. Li. Particle swarm optimisation for evolving artificial neural network. In *2000 IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 2487–2490. IEEE, 2000.