Reliability Engineering and System Safety 94 (2009) 1183–1193

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Hierarchical adaptive experimental design for Gaussian process emulators

Daniel Busby *

IFP (Institut Français du Pétrole), Rueil-Malmaison 92500, France

ARTICLE INFO

Article history:
Received 30 October 2007
Accepted 4 July 2008
Available online 18 July 2008

Keywords:
Sequential experimental design
Gaussian process regression
Data-adaptive modeling
Reservoir forecasting
Sensitivity analysis

ABSTRACT

Large computer simulators have usually complex and nonlinear input output functions. This complicated input output relation can be analyzed by global sensitivity analysis; however, this usually requires massive Monte Carlo simulations. To effectively reduce the number of simulations, statistical techniques such as Gaussian process emulators can be adopted. The accuracy and reliability of these emulators strongly depend on the experimental design where suitable evaluation points are selected. In this paper a new sequential design strategy called hierarchical adaptive design is proposed to obtain an accurate emulator using the least possible number of simulations. The hierarchical design proposed in this paper is tested on various standard analytic functions and on a challenging reservoir forecasting application. Comparisons with standard one-stage designs such as maximin latin hypercube designs show that the hierarchical adaptive design produces a more accurate emulator with the same number of computer experiments. Moreover a stopping criterion is proposed that enables to perform the number of simulations necessary to obtain required approximation accuracy.

1. Introduction

In many industrial applications computer models, denoted as simulators, are used to predict the behavior of complex physical systems. Such simulators are used for instance in reservoir engineering applications to predict and improve oil recovery or in the automobile industry to test engine performance. Inputs of such simulators are estimated by experts and can be highly uncertain. Moreover, some of these input parameters need to be calibrated using observations of the physical system and others need to be optimized. All these operations usually require performing a very high number of simulations, and because simulators usually take long time to run, this problem can become unpractical.

In mathematical terms, a simulator output can be represented as a function $\mathbf{y} = f(\mathbf{x})$ with $\mathbf{x} \in \Omega \subset \mathbb{R}^d$, and a simulator run is defined as evaluating the function f for a particular input configuration $\mathbf{x}$.

To propagate uncertainty from input $\mathbf{x}$ through output $\mathbf{y}$, or to study the sensitivity of $\mathbf{y}$ to variations in $\mathbf{x}$, many evaluations of f at some scattered dataset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \Omega$ are usually performed. In particular, for global sensitivity analysis [1], the number of required runs can be of the order of several thousands (see [2]).

To reduce this very high number of direct simulations a possible approach is to construct a statistical approximation of the function f from some initial training dataset $X_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. A statistical approximation of f, usually referred to as an emulator [3] or metamodel [4], provides for an input $\mathbf{x}^*$, not belonging to the training dataset X_1, a complete probability distribution of $f(\mathbf{x}^*)$. The most probable value of the distribution of $f(\mathbf{x}^*)$ is then the best estimate of $f(\mathbf{x}^*)$ given the simulation results $\{f(\mathbf{x}) : \mathbf{x} \in X_1\}$, obtained at the training dataset X_1.

In this paper we consider the situation in which there are no particular engineering restrictions to select the training dataset. Also the dataset can be freely enlarged in order to produce a better analysis of the input output relationship.

In experimental design one studies how to choose the best training dataset to perform a given statistical analysis on the data. Traditionally experimental designs are a one-stage process [5], and presuppose a given parametric model of the input output relation. For nonparametric models the number of necessary points to obtain a reliable approximation depends on the complexity of the function under investigation; then multistage processes referred to as sequential designs, are preferred [4,6–8].

Our final goal is to provide an efficient and reliable method to compute global sensitivity analysis indices and to propagate uncertainty from input to output [9]. To this end, following the lines of our previous work [7], we propose a hierarchical sequential design where the objective is to obtain a best approximation of the function f using the least possible number of simulation runs. Approximations of the function f are obtained by Gaussian process (GP) regression. Similarly to [9] the obtained

* Tel.: +33 147 527 406; fax: +33 014 752 5617.
E-mail address: Daniel.BUSBY@ifp.fr

0951-8320/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2008.07.007

approximation is then used instead of the simulator to compute sensitivity indices and to propagate uncertainty.

Note that GP regression is a statistical approximation method that allows to estimate the full distribution of f at untried input $\mathbf{x}$. However, information about the full distribution is usually unreliable, therefore in this work we follow a common practice to use only the best estimate of f to compute sensitivity indices and for uncertainty propagation.

The problem of finding the best estimate is then equivalent to scattered data approximation [10]. However, statistical approximations methods such as GP regression are generally more flexible and provide more valuable physical interpretation than their scattered data approximation equivalent such as radial basis functions or splines [11].

We remark that GP regression methods rely on the choice of the covariance function (kernel) model. The kernel models are usually parameterized and obtaining information about these *hyperparameters* is usually a difficult and computationally expensive task. A common method for computing hyperparameters is maximum likelihood estimation (MLE), alternatively they can be integrated out using Markov Chain Monte Carlo (MCMC) methods in Bayesian frameworks [12]. Note that obtaining good estimates of hyperparameters can be crucial to provide accurate inferences.

In this work a multistage experimental design is proposed for GP regression to provide a better estimate of the kernel hyperparameters and to finally improve the approximation quality at minimum cost in terms of simulations. Several improvements of the hierarchical nonlinear approximation scheme presented in a previous work [7] are proposed.

In a hierarchical approximation scheme a sequence $s_1, s_2, \ldots, s_L$ of approximations to an unknown function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, is constructed from samples of f taken at scattered locations $X = \{x_1, \ldots, x_m\}$. The construction of the approximation relies on a data hierarchy

$$X_1 \subset X_2 \subset \cdots \subset X_L \subset X \tag{1}$$

of nested subsets of X, where L denotes the number of levels. Then, the functions s_ℓ approximate f at the subsets X_ℓ of the level $\ell, 1 \leqslant \ell \leqslant L$, according to some specific approximation scheme. Note that the sequence $s_1, s_2, \ldots, s_L$ of approximations to f is from coarse to fine. Indeed, the coarsest approximation s_1 is computed on the basis of the initial design X_1, which includes only very few data. In contrast, the approximations s_ℓ at finer levels ℓ contain gradually more information from their corresponding designs X_ℓ.

As discussed in O'Hagan [3], we call the statistical representation of f (the GP regression model) an *emulator*, whether the emulator mean is our best estimate or approximation of the function f.

At each step of the sequential design we use information about the emulator to select the next inputs to sample in order to increase the approximation accuracy.

Following the lines of our previous work [7], the experiment selection at each iteration is performed by adaptive domain decomposition of the input space followed by local design in each subdomain. The adaptive domain decomposition called *adaptive gridding* is hierarchical from coarse to fine and its computation is based on the following observation: the more the function wiggles the more points are needed to approximate the function. This information is embedded in the parameters of the covariance function of the Gaussian emulator (in the so-called correlation lengths or roughness parameters), which are estimated by maximum likelihood.

Adaptivity is also used to select in which subdomain points are to be added. The adaptive criterion is based on a preselected accuracy level of the approximation. The chosen accuracy level also affects the stopping criterion of the sequential design. To estimate the approximation accuracy a measure of the prediction error is computed in each subdomain using cross validation. This prediction error is then compared to the selected accuracy level, to decide whether or not adding a point in a given subdomain. The stopping criterion is reached when the global prediction error (average prediction error of all subdomains) is below the accuracy threshold.

A formal description of the proposed hierarchical approximation is given in the following algorithm scheme:

Algorithm 1 (*Hierarchical GP regression*). *Input*: *Initial Design* X_1 *with corresponding simulator output values* $\{f(\mathbf{x}) : \mathbf{x} \in X_1\}$ *and a selected relative accuracy target* α.

```
(1)  Construct the initial design $X_1$ and run the simulator at $X_1$ to
     obtain the initial data $\{f(\mathbf{x}) : \mathbf{x} \in X_1\}$.
(2)  FOR $\ell = 1, 2, 3, \ldots$ DO
(2a)   Build the emulator $s_l$ and estimate the approximation accuracy
       $\alpha_l$.
(2b)   IF $\alpha_l < \alpha$, EXIT
(2c)   Construct the new sites $X_{\ell+1} = X_\ell \cup X_{\mathrm{add}}$ by a two-stage
       experimental design: apply global adaptive gridding, followed
       by local maximin design.
(2d)   Run the simulator at the new design sites $X_{\mathrm{add}}$. Obtain design
       responses $\{f(\mathbf{x}) : \mathbf{x} \in X_{\ell+1}\}$.
```

Output: *Design sites* X_L *and responses* $\{f(\mathbf{x}) : \mathbf{x} \in X_L\}$, *emulator* s_L *with an estimated approximation accuracy* $\alpha_L < \alpha$.

This hierarchical approximation scheme results in a multiresolution representation of the emulator, from coarse to fine. That is, with increasing accuracy at each iteration, to obtain a sufficiently accurate approximation at the finest level after merely a few iterations. This refinement strategy allows control of the gradually increasing computational costs for building the hierarchical emulator sequence, and their increasing approximation accuracy.

A typical application of our method is reservoir forecasting. In this application usually 10–20 uncertain inputs are considered and the number of affordable simulations is a few hundreds.

The outline of the paper is as follows. In Section 2, the GP regression is reviewed. Section 3 describes in detail the new hierarchical adaptive experimental design (HAED). Finally, Section 4 presents numerical results obtained by applying our method to two synthetic functions, being regarded as a simulator, and to a synthetic oil reservoir test case using a commercial fluid flow simulator.

It is shown that our hierarchical scheme effectively manages to increase the approximation accuracy. Moreover, for the analyzed test cases, the sequential design outperforms state of the art maximin latin hypercube design (LHD) [13,14].

2. GP regression

The construction of the individual approximations $s_\ell, \ell = 1, \ldots, L$, in Algorithm 1 is based on GP regression. In this section the frequentist approach of GP regression, Kriging is reviewed. The Kriging method was introduced by Matheron [15] in geostatistics, and then used by Sacks et al. [6] for the design and analysis of computer experiments. For a Bayesian formulation of GP regression see Kennedy and O'Hagan [16] or for a machine learning perspective the recent book of Rasmussen and Williams [12]. Note also that the type of computer simulators considered in this work are deterministic, i.e., rerunning the code with the same input $\mathbf{x}$ produces exactly the same outputs $\mathbf{y}$.

Consider the output of a simulator as an unknown deterministic function, say $f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$. Our objective is to predict the value of f at some point x, given a design $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, and