Advances in Engineering Software 100 (2016) 19–31

Contents lists available at ScienceDirect

Advances in Engineering Software

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

A software framework for probabilistic sensitivity analysis for computationally expensive models

N. Vu-Bac [c], T. Lahmer [c], X. Zhuang [e,f,*], T. Nguyen-Thoi [d,b], T. Rabczuk [a,b,*]

[a] Division of Computational Mechanics, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[b] Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[c] Institute of Structural Mechanics, Bauhaus-Universität Weimar, Marienstr. 15, D-99423 Weimar, Germany
[d] Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[e] State Key Laboratory for Disaster Reduction in Civil Engineering, College of Civil Engineering, Tongji University, Shanghai, China
[f] Institute of Continuum Mechanics, Leibniz University Hannover, Germany

ARTICLE INFO

Article history:
Received 22 August 2015
Revised 30 May 2016
Accepted 5 June 2016

Keywords:
Uncertainty quantification
Random sampling
Penalized spline regression
Sensitivity analysis
Matlab toolbox

ABSTRACT

We provide a sensitivity analysis toolbox consisting of a set of Matlab functions that offer utilities for quantifying the influence of uncertain input parameters on uncertain model outputs. It allows the determination of the key input parameters of an output of interest. The results are based on a probability density function (PDF) provided for the input parameters. The toolbox for uncertainty and sensitivity analysis methods consists of three ingredients: (1) sampling method, (2) surrogate models, (3) sensitivity analysis (SA) method. Numerical studies based on analytical functions associated with noise and industrial data are performed to prove the usefulness and effectiveness of this study.

1. Introduction

In many fields such as structural reliability [30–32], material modeling [30,31,33], finance etc., mathematical (numerical) models are used for predicting the response of a system. Due to the increasing computer power, the complexity of the model is growing. Generally, the more complex the models are, the larger becomes the uncertainty in the model outputs due to randomness in the input parameters. It is essential to determine how much the model output is changed by the variation in input parameters as well as calibrate and validate the mathematical models. Sensitivity analysis (SA) is a great help for these purposes [1]. Therefore, uncertainty and sensitivity analysis have recently received widespread interest of researcher in many fields such as material modeling and structural design. Numerous SA approaches have been developed to quantify the models with uncorrelated parameters [2]. However, engineering systems are complex and frequently contain correlated input parameters such that if one parameter varies, it results in variations in other parameters. The variation in the output of the models with correlated input parameters (e.g., composition constraints in material modeling [3]) is not only contributed by the variations in input parameter itself, but also contributed by the correlated variations in other parameters [4]. Hence, it is more realistic to estimate the effects of changing more than one parameters on the model outputs simultaneously. It is essential to understand the relations among the uncertain input parameters for designing a SA.

A few methods have been developed to quantitatively assess the effect of correlated input parameters on the model outputs. For instance, Xu and Gernert [4] improved the original Fourier amplitude sensitivity test (FAST) associated with Iman and Conover method [18] – used to generate correlated samples – to properly measure the sensitivity index for a model with correlated input parameters. Then, they developed another method to evaluate the sensitivity index for the uncorrelated and correlated contributions, see [5]. Nevertheless, those methods are limited in estimating the first-order sensitivity indices and the latter is based on a weak assumption that the model output linearly relates to input parameters. Later, SA methods, see [1,6,7], were proposed to quantitatively assess the total-effect sensitivity index that is essential for model simplification, however, most of them deal with analytical functions but not for experimental or simulation data.

Hence, a unified framework that links different steps from generating sample, constructing the surrogate model and implementing the sensitivity analysis method is needed. In this article, a review and computer implementation for uncertainty and sensitivity

* Corresponding authors.
E-mail addresses: bac.nam.vu@uni-weimar.de (N. Vu-Bac), zhuang@ikm.uni-hannover.de (X. Zhuang), timon.rabczuk@tdt.edu.vn (T. Rabczuk).

http://dx.doi.org/10.1016/j.advengsoft.2016.06.005
0965-9978/

Nomenclature

$\boldsymbol{\beta}$	Vector of regression coefficients
$\boldsymbol{\nu}$	Vector of regression coefficients for penalized spline regression model
λ	Smoothing parameter
$\boldsymbol{\epsilon}$	Vector of errors
$\mathbf{C}$	Truncated power basis of degree p
$\mathbf{R}^2$	Coefficient of determination (COD)
$\mathbf{X}$	Random variable vector $\mathbf{X} = [X_1, X_2, \ldots, X_N]^T$
$\mathbf{x}$	Vector of design variables
$\mathbf{Y}$	Vector of responses at sampled design points
μ	Mean vector of variables
ρ_{ij}	Coefficient of correlation
Σ	Covariance matrix
σ	Standard deviation (std. dev.) vector of variables
d	Number of variables (parameters)
$E(\cdot)$	Expected value of the quantity
$F(\cdot)$	Cumulative distribution function
$f(\cdot)$	True function to be modeled
N	Sample size
p	Degree of spline basis
$p(\cdot)$	Probability density function
$Prob$	Probability value
r_u	Uniformly distributed random number
RSS	Residual sum of squares
S_i	First-order sensitivity index of *ith* variable X_i
S_y	First-order sensitivity index of group y
S_{Ti}	Total-effect sensitivity index of *ith* variable X_i
S_{Ty}	Total-effect sensitivity index of group y
V	Total variance
$V(\cdot)$	Variance value of the quantity
V_i	Partial variance of *ith* variable X_i
X_i	Generic variable (parameter)

analysis and its application in engineering analysis has been carried out to provide a robust and powerful modeling tool to support for designing uncertainty and sensitivity analysis. We employ a sensitivity analysis (SA) method for the case of correlated parameters [6] whose formulas were derived similarly to Sobol' formulas for the case of uncorrelated parameters [8]. For the estimation, the sample data is generated from the joint and conditional probability distribution functions of input parameters which are required to account for the constraints in the inputs space. Gaussian copula is used to generate a joint cumulative distribution function (CDF) (multivariate normal distribution) that requires only marginal distributions and covariance matrix of input parameters.

Complex models are often very time-consuming and computationally expensive so that they cannot be used to compute sensitivity indices. Thus, the so-called surrogate-based approach is employed as an approximation of the real model for sensitivity analysis. In [10], the authors presented a penalized spline regression model for a single continuous predictor. Since predictor variables have nonlinear relationships with the model output, the regression models considering multiple smooth functions [11] are adopted in this article to approximate the observed data. Subsequently, the SA indices are computed based on penalized spline regression models.

The objective of this work is to provide a MATLAB toolbox consisting of a set of functions that can be used to randomly generate samples, construct the surrogate model and carry out the SA. The computer implementation has been presented in this article. The support MATLAB code can be found at the website (http://www.uni-weimar.de/Bauing/rabczuk/).

The article is outlined as follows. In the next section, we briefly describe the flow chart and structure of the framework. The sampling technique is shown in Section 3. Surrogate models containing polynomial and penalized spline regression models are presented in Section 4.2. The SA is described in Section 5. Application of the SA method for models with correlated input parameters are presented in Section 6 including two analytical models with additional noise and an industrial example. Finally, we close the manuscript with concluding remarks.

2. Matlab toolbox: A flowchart and structure of the code

A framework including sampling of correlated input parameters, construction of surrogate model, and implementation of sensitivity analysis are respectively described in Fig. 1. Also, the structure and purpose of the code are briefly depicted in Table 1. In the first step, sampling technique is used to randomly generate correlated input values which are then inserted into the computational model to obtain the model response in the second step. In the third step, the regression model is used to approximate the obtained training data from the previous step. Finally, SA approach is employed to estimate sensitivity indices based on the regression model. Accordingly, MATLAB subroutines for each step are given in Table 1.

3. Sampling method

Latin Hypercube Sampling (LHS) [14,15] is an improved sampling strategy that enables a reliable approximation of the stochastic properties even for a small number of samples N. LHS is used to provide the design points which are spread throughout the design space. The LHS can be summarized as:

- Divide the cumulative curve into N equal intervals on the cumulative distribution of each parameter;
- A probability value is then randomly selected from each interval of the parameter distribution

  $$Prob_i = (1/N) r_u + (i-1)/N, \tag{1}$$

  in the *ith* interval, where r_u is a uniformly distributed random number varying over the range [0, 1], see [16];
- Use the inverse cumulative distribution function (CDF) to map the probability value $Prob_i$ into the design space as:

  $$x = F^{-1}(Prob); \tag{2}$$

  where F^{-1} denotes the inverse CDF.

LHS reduces the computer effort due to the dense stratification across the range of each sampled variable [17]. Random number generators are widely used for sampling strategy of uncorrelated parameters. The MATLAB® program *lhsu.m* is improved based on the Kriging toolbox [13] and [14,18,19] is used to obtain the location of the design points.

```
function s = lhsu(N,k)
% Purpose
% LHS from uniformly uncorrelated distribution
% Inputs
% N : number of sample points
% k : number of variables
% Output
% s : the generated k dimensional N sample points chosen from uniform ...
    distributions on N subdivions of the interval (0.0, 1.0)
% authors: adapted from Budiman (2003)
%           modified by N. Vu-Bac et al. (2015)
ran=rand(N,k);
x=zeros(N,k);
for i=1:k
   idx=randperm(N);
   x(:,i)=(idx'+(ran(:,i)-1))/N;
end
```