Environmental Modelling & Software 74 (2015) 140–155

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Simple approach to emulating complex computer models for global sensitivity analysis

Bryan Stanfill [a], [*], Henrike Mielenz [a], David Clifford [a], [b], Peter Thorburn [a]

[a] CSIRO, Dutton Park, QLD 4102, Australia
[b] The Climate Corporation, San Francisco, CA 94103, USA

ARTICLE INFO

Article history:
Received 7 May 2015
Received in revised form
17 September 2015
Accepted 17 September 2015
Available online 1 October 2015

Keywords:
Agricultural simulator
APSIM
Generalized additive model
Meta-model
Uncertainty quantification

ABSTRACT

Sensitivity analysis is an important step in understanding how uncertainty is propagated through complex computer models. Unfortunately, the most reliable sensitivity analysis techniques take a significant amount of time to execute due to the large number of computer model evaluations required. Emulators can be used to speed up the process by replacing the computer model with a statistical model that mimics the computer model and is computationally efficient. In this manuscript we propose two emulator-based sensitivity index estimators that require minimal set-up and are computationally inexpensive to compute. We demonstrate their accuracy with computer models that have known sensitivity index values and illustrate their application in practice with the agriculture systems simulator APSIM.

1. Introduction

The global sensitivity analysis of complex computer models requires hundreds to thousands of computer model evaluations before a reliable measure of sensitivity can be computed (Cukier et al., 1973; Jansen, 1999; Saltelli et al., 2010). Additionally, the number of computer model evaluations required increases rapidly as the number of inputs included in the analysis increases. Computationally expensive computer models are therefore difficult, if not impossible, to analyze accurately.

Statistical models called emulators, or meta-models, are a popular tool to reduce the number of computer model evaluations required for an accurate sensitivity analysis. Gaussian processes are a common basis for emulators though Kalman filters and machine learning methods have been shown to be effective as well (Oakley and O'Hagan, 2004; Ratto et al., 2007, 2009; Storlie and Helton, 2008; Storlie et al., 2009). In a parallel but distinct stream of literature, high dimensional model representation (HDMR) methods have been used to emulate computer models, most notably by Li et al. (2010) and Li and Rabitz (2012). Finally, expansion techniques, such as the polynomial chaos (PC) expansion, have also been used to build emulators for global sensitivity analysis (Sudret, 2008).

Though Gaussian processes, Kalman filters, HDMR and PC methods are widely applicable, they often require a substantial amount of time and care from the researcher to achieve reliable results. Gaussian processes require and are sensitive to initial value and prior distribution choices made by the researcher; choices often made with little prior information (Oakley and O'Hagan, 2004; Strong et al., 2014). To implement PC methods, the maximum degree of the polynomials used to represent the computer model must be specified, though recent PC methods have automated that process (Blatman and Sudret, 2011; Narayan and Xiu, 2012; Buzzard, 2013). Kalman filters and HDMR methods require the same care in initial set-up and additionally require a large amount of ad hoc programming (Ratto et al., 2007; Li et al., 2010).

If care is not taken in initializing these forms of emulators, then it is quite common to return qualitatively incorrect results. For example, if a Guassian process emulator is not initialized correctly, it could erroneously identify an input as important. As described in Bastos and O'Hagan (2009), incorrect specification of initial parameters could lead to biased estimates of the simulator outputs as well as confidence regions that are too wide or too narrow for both the emulator predictions for the model output as well as the

[*] Corresponding author.
E-mail addresses: bryan.stanfill@csiro.au (B. Stanfill), henrike.mielenz@csiro.au (H. Mielenz), david.clifford@climate.com (D. Clifford), peter.thorburn@csiro.au (P. Thorburn).

http://dx.doi.org/10.1016/j.envsoft.2015.09.011
1364-8152/© 2015 Elsevier Ltd. All rights reserved.

estimate of uncertainty about the true model outputs. Some difficulties related to the initialization and execution of emulators have been alleviated by Dakota, a program that can be used to implement several types of emulators including Gaussian processes and PC methods (Adams et al., 2014). Unfortunately, Dakota is not an easy program to master and can not be compared to programs such as R in terms of generality. Therefore, a new class of emulators with few initialization requirements that can be applied widely is of interest.

A recent development in emulator literature exploits the flexible and easy to implement structure of generalized additive models (GAMs) to emulate computer model output. Mara and Joseph (2008) proposed separate GAMs be fit for each input in order to estimate each input's first-order sensitivity index. More complex GAM-based emulators were proposed in Storlie and Helton (2008) and Storlie et al. (2009) that use a complex variable selection method to estimate first-order and total sensitivity indices. Strong et al. (2014) used GAMs based on subsets of the input chosen by the researcher to efficiently estimate the expected value of perfect information in the context of computer models for medical decision making.

While the currently available GAM-based emulators have several advantages over Gaussian process- and Kalman filter-based emulators, they can be made more efficient. The separate GAM method of Mara and Joseph (2008) is easy to implement, but it can only estimate first-order indices for each input. The variable selection methods of Storlie and Helton (2008) and Storlie et al. (2009), on the other hand, can be used to estimate higher-order sensitivity indices, but they are highly complex and difficult to implement. The generality and computation simplicity of the single GAM approach introduced in Strong et al. (2014) is promising, but the use of their method to estimate sensitivity measures is not immediate.

In this manuscript we propose two emulation methods that combine the HDMR approach of Li et al. (2010) and the GAM methodology of Strong et al. (2014) to estimate first-order and total sensitivity indices based on a single, low-dimensional GAM that can be easily implemented using standard data analysis software. The proposed methods require no initial parameter values be provided by the user and are therefore easier to implement correctly than other popular emulators. The efficiency and accuracy of our proposed emulators are compared to popular methods by applying them to computer models where the sensitivity indices are known analytically. We also demonstrate how our method can be applied in practice by applying it to the wheat module of the agriculture systems simulator APSIM. In the supplementary material we demonstrate the computational simplicity of our approach by illustrating its implementation with the freely-available statistical software R (R Core Team, 2014).

2. Global sensitivity analysis methods

In this section we briefly describe the current best practice for variance based sensitivity analysis methods. Note that we only consider univariate computer models in this manuscript though the extension to multivariate methods is possible a la Campbell et al. (2006).

2.1. Variance based sensitivity analysis

Variance based sensitivity analysis is based on the idea that computer models can be decomposed into pieces that are functions of the inputs. As a consequence, the uncertainty in the output can be decomposed into contributions made by each of the inputs and their interactions.

Formally, let Y represent the scalar output of the computer model, which is represented by the function $f(\bullet)$, and let $\boldsymbol{X}$ represent the p-dimensional input vector $\boldsymbol{X} = (X_1,\ldots,X_p)$. Then $Y = f(\boldsymbol{X})$ can be decomposed into 2^p pieces that are attributable to a subset of the inputs as

$$\begin{aligned} Y &= f(\boldsymbol{X}) \\ &= f_0 + \sum_{i=1}^{p} f_i(X_i) + \sum_{i=1}^{p-1}\sum_{j=i+1}^{p} f_{ij}(X_i, X_j) + \ldots + f_{1\ldots p}(X_1, \ldots, X_p) \end{aligned} \tag{1}$$

where f_0 is the overall mean of Y, $f_i(X_i)$ is the mean of Y given X_i after removing the overall mean, $f_{ij}(X_i,X_j)$ is the mean of Y given X_i and X_j after the mean of Y and the marginal means given X_i and X_j have been removed, and so on for the higher-order terms.

Put another way, let $\boldsymbol{X}_{-i}$ represent the vector of all inputs except X_i, $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ be the expected value of Y taken over all possible values of $\boldsymbol{X}_{-i}$ with X_i fixed and $\mathrm{Var}_{X_i}[E_{\boldsymbol{X}_{-i}}(Y|X_i)]$ denote the variance of $E_{\boldsymbol{X}_{-i}}(Y|X_i)$ taken over all possible values of X_i. In the same fashion as $\boldsymbol{X}_{-i}$, define $\boldsymbol{X}_{-ij}$, its conditional expected value $E_{\boldsymbol{X}_{-ij}}(Y|X_i, X_j)$ and conditional variance $\mathrm{Var}_{X_i,X_j}[E_{\boldsymbol{X}_{-ij}}(Y|X_i, X_j)]$. Then $f_0 = E(Y)$, $f_i(X_i) = E_{\boldsymbol{X}_{-i}}(Y|X_i) - f_0$, and $f_{ij}(X_i, X_j) = E_{\boldsymbol{X}_{-ij}}(Y|X_i, X_j) - E_{\boldsymbol{X}_{-i}}(Y|X_i) - E_{\boldsymbol{X}_{-j}}(Y|X_j) + E(Y)$.

The variance of Y can be decomposed into pieces that are attributed to the main effect for each input and higher-order interactions between inputs. That is,

$$\mathrm{Var}(Y) = \sum_{i=1}^{p} V_i + \sum_{i=1}^{p-1}\sum_{j=i+1}^{p} V_{ij} + \ldots + V_{1\ldots p}$$

where V_i is the variance of Y that can be attributed to X_i alone, V_{ij} is the variance of Y that can be attributed to the interaction between X_i and X_j after their respective main effects have been removed and similarly for the higher-order terms. Each input's contribution to the uncertainty in Y can be rewritten using quantities defined in (1) as

$$\begin{aligned} V_i &= \mathrm{Var}[f_i(X_i)] = \mathrm{Var}_{X_i}\left[E_{\boldsymbol{X}_{-i}}(Y|X_i)\right], \\ V_{ij} &= \mathrm{Var}\left[f_{ij}(X_i, X_j)\right] \\ &= \mathrm{Var}_{X_i,X_j}\left[E_{\boldsymbol{X}_{-ij}}(Y|X_i, X_j)\right] - \mathrm{Var}_{X_i}\left[E_{\boldsymbol{X}_{-i}}(Y|X_i)\right] \\ &\quad - \mathrm{Var}_{X_j}\left[E_{\boldsymbol{X}_{-j}}(Y|X_j)\right], \end{aligned} \tag{2}$$

and similarly for higher-order terms (Saltelli et al., 2010).

Dividing the component of the variance decomposition associated with a subset of inputs by the total variance in Y gives the proportion of variability in Y that can be attributed to that subset of inputs. This quantity is called the sensitivity index for that subset of inputs. The sensitivity indices for the main effect of X_i and the interaction between X_i and X_j are

$$S_i = \frac{V_i}{\mathrm{Var}(Y)} \text{ and } S_{ij} = \frac{V_{ij}}{\mathrm{Var}(Y)},$$

respectively. The first-order and total sensitivity indices for X_i are given by S_i and T_i, respectively. The total sensitivity index for X_i is the sum of S_i plus all higher-order terms involving X_i and is defined as