Journal of Statistical Planning and Inference 144 (2014) 160–172

Contents lists available at ScienceDirect

Journal of Statistical Planning and Inference

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

Estimating sensitivity indices based on Gaussian process metamodels with compactly supported correlation functions

Joshua Svenson [a], Thomas Santner [b,*], Angela Dean [b,c], Hyejung Moon [d]

[a] JPMorgan Chase & Co., 1111 Polaris Parkway, Columbus, OH 43240, USA
[b] Department of Statistics, The Ohio State University, 1958 Neil Avenue, Columbus, OH 43210, USA
[c] Department of Mathematics, University of Southampton, Southampton SO17 1BJ, UK
[d] The Bank of Korea, 110, 3-Ga, Namdaemun-Ro, Jung-Gu, Seoul 100-794, Republic of Korea

ARTICLE INFO

Available online 18 April 2013

Keywords:
Bayesian estimation
Computer experiments
Global sensitivity indices
Main-effect sensitivity indices
Process-based estimator
Quadrature-based estimator
Total sensitivity indices

ABSTRACT

Specific formulae are derived for quadrature-based estimators of global sensitivity indices when the unknown function can be modeled by a regression plus stationary Gaussian process using the Gaussian, Bohman, or cubic correlation functions. Estimation formulae are derived for the computation of process-based Bayesian and empirical Bayesian estimates of global sensitivity indices when the observed data are the function values corrupted by noise. It is shown how to restrict the parameter space for the compactly supported Bohman and cubic correlation functions so that (at least) a given proportion of the training data correlation entries are zero. This feature is important in the situation where the set of training data is large. The estimation methods are illustrated and compared via examples.

1. Introduction

A *computer experiment* uses a computer simulator based on a mathematical model of a physical process as an experimental tool to determine "responses" or "outputs" at a set of user-specified input sites. These input sites constitute the design for the computer experiment. Sophisticated computer codes may take hours or even days to produce an output and, therefore, a flexible and rapidly computable predictor, sometimes called a code *emulator* or *metamodel*, is often fitted to the inputs/outputs of the design, which are then called *training data*. An emulator allows the detailed, albeit approximate, exploration of the output over the entire experimental region (see, for example, Sacks et al., 1989b; Santner et al., 2003). A *sensitivity analysis*, based on the outputs of either the simulator or emulator, enables the researcher to assess the variation in the output due to changes in individual inputs or groups of inputs (see, for example Saltelli et al., 2000; Helton et al., 2006; Oakley and O'Hagan, 2004).

In this paper, we assume that the computer simulator has d continuous input variables denoted by the vector $\boldsymbol{x} = (x_1, \ldots, x_d)$ and that the (one-dimensional) output of the simulator, denoted by $y(\boldsymbol{x}) = y(x_1, \ldots, x_d)$, can be determined for $\boldsymbol{x}$ in the hyper-rectangle $\mathcal{X} = \prod_{j=1}^{d} [l_j, u_j]$, but is computationally expensive. The sensitivity of $y(\boldsymbol{x})$ to the input values $\boldsymbol{x}$ can be measured locally or globally. A *local sensitivity index* is based on the change in $y(\cdot)$ at a specified $\boldsymbol{x}^0 = (x_1^0, \ldots, x_d^0)$ as the j^{th} input varies by a small amount parallel to the x_j-axis and this can be measured by the partial derivatives of $y(\cdot)$ with respect

* Corresponding author. Tel.: +1 614 292 2866; fax: +1 614 292 2096.
E-mail addresses: joshua.d.svenson@chase.com (J. Svenson), santner.1@osu.edu (T. Santner), dean.9@osu.edu (A. Dean), hjmoonoh@gmail.com (H. Moon).

0378-3758/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jspi.2013.04.003

to x_j. In contrast, a first (or higher) order *global sensitivity index* measures the change in $y(\cdot)$ as one (or more) inputs vary over their entire range, when the remaining inputs are fixed (see, for example Saltelli, 2002). Homma and Saltelli (1996) further defined the jth *total sensitivity index* as a measure of the change in $y(\cdot)$ due to the jth input, both through its main effect and its joint effect with other inputs. Chen et al. (2005, 2006) defined *subset sensitivity indices* based on non-overlapping partitions of the inputs. One popular definition of global sensitivity indices is in terms of the variability of the (weighted) average output $y(\boldsymbol{x})$ over $\boldsymbol{x}\in\mathcal{X}=\prod_{j=1}^{d}[l_j,u_j]$, as reviewed in Section 2.

As well as providing an understanding of the input/output relationship, sensitivity analysis provides a tool for "screening", that is for selecting the inputs that have major impacts on an input–output system, thereby allowing researchers to restrict attention to these important inputs while setting the others to nominal values in their computational simulator. For various discussions and applications of sensitivity analysis and screening, see for example, Welch et al. (1992), Linkletter et al. (2006), Moon et al. (2012), and the references cited therein.

For estimating local sensitivity indices, Morris (1991) proposed the use of "elementary effects" calculated directly from the simulator output, with inputs selected according to a "one-at-time" sampling design. This methodology was extended by Campolongo et al. (2007). Sampling designs for estimating global sensitivity indices were presented and discussed by, for example, Saltelli (2002), Morris et al. (2008), Da Viega et al. (2009), and Saltelli et al. (2010). In the case when the simulator is expensive to run, such estimation methods may require more simulator runs than is feasible in order to produce accurate global sensitivity index estimates. Chen et al. (2005), Oakley and O'Hagan (2004), Marrel et al. (2009), and Storlie et al. (2013) gave alternative estimation methods based on analytical and probabilistic methods using emulators.

In this paper, we use the popular $y(\boldsymbol{x})$ emulator based on a Gaussian process model as proposed, for example, by Sacks et al. (1989b), and which has the form

$$Y(\boldsymbol{x})=\boldsymbol{f}^{\top}(\boldsymbol{x})\boldsymbol{\beta}+Z(\boldsymbol{x}), \tag{1.1}$$

where $\boldsymbol{f}^{\top}(\boldsymbol{x})\boldsymbol{\beta}$ is a linear function of an unknown regression parameter vector $\boldsymbol{\beta}$, and $Z(\boldsymbol{x})$ is a zero-mean Gaussian process having variance σ^2. Assuming this type of model, Sacks et al. (1989a, 1989b) and Welch et al. (1992) used a $y(\boldsymbol{x})$ predictor derived from the classical theory of best linear unbiased prediction. Other authors, including Currin et al. (1991), O'Hagan (1992), Oakley and O'Hagan (2004), have viewed the random function $Y(\boldsymbol{x})$ as representing prior uncertainty about the true function and adopted a Bayesian approach to estimation.

The purpose of this paper is to give specific formulae for global sensitivity index estimates for a broad class of regression plus Gaussian process models (1.1) with independent inputs in the special case of stationary $Z(\boldsymbol{x})$ with compactly supported Bohman and cubic (separable) correlation functions. As compared with the often-used Gaussian correlation function, use of compactly supported correlation functions together with a suitably rich mean structure has the potential to provide sparse correlation matrices, thus allowing prediction to be performed with larger data sets within the Gaussian process framework (see Kaufman et al., 2011).

In Section 3, we give formulae for quadrature-based methods of estimation using Gaussian processes with polynomial mean and either Gaussian or Bohman correlation functions. In the on-line Supplementary Material, we provide the corresponding formulae for the cubic correlation function. In Section 4, together with the Supplementary Material, we derive the specific formulae required to compute both fully Bayesian and empirical (plug-in) Bayesian estimates of sensitivity indices. The formulae in these two sections extend the work of Chen et al. (2005), Oakley and O'Hagan (2004), Marrel et al. (2009), and others, who provide explicit formulae for global sensitivity estimators for Gaussian process emulators with constant mean and Gaussian correlation function.

In Section 6, it is shown via two examples that sensitivity indices estimated using output from a Gaussian process emulator under the compactly supported Bohman, and cubic correlation functions are similar to the estimates obtained using the Gaussian correlation function, but that the computational times are much shorter. Although the current examples are not extremely large, they illustrate the potential computational savings, described by Kaufman et al. (2011), that can be achieved when handling large data sets and/or large numbers of inputs. In line with previous studies, our examples also illustrate that calculation of sensitivity indices using a moment-based estimation method (based on "permuted column sampling" as described by Morris et al., 2008) is less accurate when using only a moderate number of simulator runs. Finally, Section 7 shows how to restrict the parameter space for the Bohman and cubic correlation functions so that (at least) a given proportion of the training data correlation entries are zero.

2. Calculation of main effect and total effect sensitivity indices

In this section, we review definitions of main effect and total effect global sensitivity indices, as described by Homma and Saltelli (1996), Saltelli (2002), Chen et al. (2005, 2006), for example. Throughout the paper, $Q=\{k_1,\ldots,k_s\}\subset\{1,2,\ldots,d\}$ denotes a non-empty subset of the input variables and $\boldsymbol{x}_Q$ denotes the vector of inputs $(x_{k_1},\ldots,x_{k_s})$ where, for definiteness, it is assumed $1\le k_1<k_2<\cdots<k_s\le d$. The vector of the remaining inputs will be denoted by $\boldsymbol{x}_{-Q}$ also arranged in lexicographical order of their input index. By rearranging the order of the entire set of input variables we write the input vector $\boldsymbol{x}$ as $\boldsymbol{x}=(\boldsymbol{x}_Q,\boldsymbol{x}_{-Q})$ in a slight abuse of notation.

Throughout the paper, we take $[l_j,u_j]=[0,1]$, for all inputs x_j, $j=1,\ldots,d$, so that $\mathcal{X}=[0,1]^d$. The formulae can be extended to the more general hyper-rectangle case. Also for simplicity of notation, it is assumed that the weight function can be