Uncertainty quantification and reduction for the monotonicity properties of expensive-to-evaluate computer models

Julien Bect, Nicolas Bousquet, Bertrand Iooss, Shijie Liu, Alice Mabille, Anne-Laure Popelin, Thibault Rivière, Rémi Stroh, Roman Sueur, Emmanuel Vazquez

To cite this version:
Julien Bect, Nicolas Bousquet, Bertrand Iooss, Shijie Liu, Alice Mabille, et al.. Uncertainty quantification and reduction for the monotonicity properties of expensive-to-evaluate computer models. Uncertainty in Computer Models 2014 Conference, Jul 2014, Sheffield, United Kingdom. hal-01103724
Abstract. We consider the problem of estimating monotonicity properties of a scalar-valued numerical model—e.g., a finite element model combined with some post-processing. Several quantitative monotonicity indicators are introduced. Since the evaluation of the numerical model is usually time-consuming, these indicators have to be estimated with a small budget of evaluations.

We adopt a Bayesian approach, where the numerical model itself is modeled as a Gaussian process. We estimate the monotonicity indicators, and quantify the uncertainty surrounding them, through conditional simulations of the Gaussian process partial derivatives. The approach is illustrated with a numerical model of a passive component in a power plant.

Future work will leverage this framework together with the Stepwise Uncertainty Reduction principle to create sequential design strategies, in order get an improved knowledge of the monotonicity properties of the model.

Understanding the structure of numerical models

Consider a deterministic numerical model:

\[ f : \mathbb{X} \rightarrow \mathbb{R}^d, \]

with the following assumptions:

- \( x \in \mathbb{X} \subseteq \mathbb{R}^d \)
- \( f(x) \in \mathbb{R} \) (scalar output),
- \( f \) is expensive to evaluate.

Computer experiments, when properly designed, are a useful tool to discover (or confirm) structural properties of such a numerical model:

- active/inactive variables (screening)
- additive responses, low-order interactions (S.A.),
- and, in this work: monotonicity properties

using a limited number of evaluations (runs of the code).

Partial monotonicity properties

We assume that

- \( \mathbb{X} \) is an hyper-rectangle: \( \mathbb{X} = \prod_{j=1}^{d} [a_j, b_j] \)
- \( f \) admits (at least first-order) partial derivatives, but \( \mathcal{O} \) is not available.

Definition (increasing case). \( f \) is said to be increasing with respect to its \( j \)th variable if

\[ \forall x^{(j)} \in \prod_{j \neq j}^{d} [a_j, b_j], \quad x^{(j)} \rightarrow f(x) \text{ is increasing}, \]

Proposition. \( f \) is increasing with respect to its \( j \)th variable if, and only if,

\[ \frac{\partial f}{\partial x^j}(x) \geq 0 \quad \forall x \in \mathbb{X}. \]

Quantitative monotonicity indicators

Several quantities of interest (nonlinear functionals of \( f \)):

- extrema of the partial derivatives

\[ M_j^f(f) = \max_{x \in \mathbb{X}} \frac{\partial f}{\partial x^j}(x), \text{ and } \tilde{M}_j^f(f) = \max_{x \in \mathbb{X}} \frac{\partial f}{\partial x^j}(x), \]

- “positivity rate” of the partial derivatives

\[ \alpha_j(f) = \mu(\Gamma_j^f(f)), \text{ where } \Gamma_j^f(f) = \left\{ \frac{\partial f}{\partial x^j}(x) \geq 0 \right\}, \]

where \( \mu \) is a given probability measure on \( \mathbb{X} \).

Proposed (Bayesian) approach:

- Endow \( f \) with a Gaussian Process prior,
- Estimate the monotonicity indicators, and quantify the uncertainty surrounding them, using conditional simulations of the partial derivatives.

Prediction and simulation of derivatives: cokriging

Consider a classical GP model:

\[ f(\beta, \theta) \sim \mathcal{GP}\left( \sum_{j=1}^{d} \beta_j h_j(\cdot), k_0(\cdot, \cdot) \right) \]

- \( h_1, \ldots, h_d \) known functions, which admit first-order partial derivatives (typically, polynomial functions)
- \( k_0 \) a stationary covariance function (for simplicity).

Theorem (see, e.g., Stein, 1999). Let \( k_0(h) = k_0(x, x + h) \). The random process \( f \) is mean-square differentiable if, and only if, \( k_0 \) is twice differentiable at \( h = 0 \).

Theorem (Scheuerer, 2010). The random process \( f \) is mean-square differentiable if, and only if, the partial derivatives exist almost surely in Sobolev’s weak sense.

Useful fact. Cokriging is just a special case of kriging with an auxiliary discrete variable:

\[ \tilde{f}(x, j) = \begin{cases} f(x) & \text{if } j = 0, \\ \frac{\partial f}{\partial x^j}(x) & \text{if } j > 0, \end{cases} \]

- No need to replace your favorite kriging software

(if it is flexible enough . . .)

Industrial test case \((d = 7)\)

The methodology has been applied to an industrial test case proposed by EDF R&D

- Goal: assess the performance of a passive component in a power plant
- Thermomechanical numerical code, \( d = 7 \) quantitative input factors.

Implementation \implies discretizations

- Intractable posterior distributions for the quantitative monotonicity indicators: we rely on conditional simulations,
- Approximate computation of the indicators themselves using a Monte Carlo sample \( x_1, \ldots, x_m (m = 200) \)
- Simulation scheme

\[ R = 100 \text { iid draws of the spatial MC run-sample,} \]

\[ S = 500 \text { iid conditional simulations on each of them,} \]

\[ RS = 50 000 \text { (dependent) conditional samplepaths.} \]

Cokriging example: prediction and 95%-CI

The true function \( f(x) \) (black line) is observed at six locations (red vertical dotted lines).

Cokriging example: conditional simulations

The kriging prediction (red line) and CI (black dashed lines) are performed using a zero-mean GP model, with a Matlab 5/2 covariance function.

References


Future work

Work is in progress on

- Advanced simulation techniques for excursion sets (see Ginsbourger et al., 2014, for a preview): will replace the crude MC-based technique used here,
- Stepwise uncertainty reduction (using ideas from Bect et al., 2012): will be used to enrich a given initial design to learn more precisely the value of the indicators \( \alpha_j, M_j^f \) and \( M_j^f \), or the set \( \Gamma_j^f \) itself!