AN EXTENDED USE OF THE SOBOL' ESTIMATOR

Andrea Pagano, **Marco Ratto**
Joint Research Centre, European Commission

SAMO 2007, 21 June, 2007

Aim

Define an extended, approximated use of the Sobol' estimator, as a first step in trying to bridge between 'sampling based' and 'smoothing' (variance-based) sensitivity approaches

Variance Based Methods

Quantify the relative importance of input variables (factors) in determining the value of an assigned output

$$y = f(x_1, \ldots, x_k)$$

Variance Based Methods

This is done computing the conditional variances V_I, defined as

$$V_i = V(E(y|x_i)), \quad \text{for} \quad I = i$$

$$V_{i_1 i_2} = V(E(y|x_{i_1}, x_{i_1})) - V_{i_1} - V_{i_2}, \quad \text{for} \quad I = \{i_1, i_2\}$$

V, E are the variance and expectation operators

Variance Based Methods

This leads to the well known variance decomposition formula (due to Sobol')

$$V(y) = \sum_i V_i + \sum_{i_1<i_2} V_{i_1 i_2} + \cdots + V_{1\ldots k}$$

$$1 = \sum_i S_i + \sum_{i_1<i_2} S_{i_1 i_2} + \cdots + S_{1\ldots k}$$

How to compute V_I

To compute $V_I = V(E(y|x_I))$ we need to calculate a double loop integral

$$V_I = \int E^2(y|x_I = \tilde{x}_I)p(\tilde{x}_I)\, d\tilde{x}_I - E^2(y)$$

$$E(y|x_I) = \int f(x_1, \ldots, x_k|x_I = \tilde{x}_I)p(x_{-I})\, dx_{-I},$$

How to compute V_I

- sampling based techniques (less efficient, rely on minimal regularity assumptions on f);
- smoothing/kriging metamodelling techniques (more efficient, rely on smoothness of f).

Sobol' Method

Let N be the 'base' sample size and k be the number of factors. To calculate V_I we need two matrices: M_a is the *sample* matrix and M_b is the *re-sample* matrix,

$$M_a = \begin{pmatrix} x^a_{11} & x^a_{12} & \dots & x^a_{1k} \\ x^a_{21} & x^a_{22} & \dots & x^a_{2k} \\ \vdots & \vdots & & \vdots \\ x^a_{N1} & x^a_{N2} & \dots & x^a_{Nk} \end{pmatrix} \qquad M_b = \begin{pmatrix} {x^b}_{11} & {x^b}_{12} & \dots & {x^b}_{1k} \\ {x^b}_{21} & {x^b}_{22} & \dots & {x^b}_{2k} \\ \vdots & \vdots & & \vdots \\ {x^b}_{N1} & {x^b}_{N2} & \dots & {x^b}_{Nk} \end{pmatrix}$$

Sobol' Estimator

As suggested by Sobol' one can use, as an estimator of V_I, the following

$$
\begin{aligned}
V_I &= V(E(y|x_I)) \\
&= \sum_{j=1}^{N} f(x^a_{j1}, \ldots, x^a_{jI}, \ldots, x^a_{jk}) f(x^b_{j1}, \ldots, x^a_{jI}, \ldots, x^b_{jk}) \\
&\quad -(\sum_{j=1}^{N} f(x^a_{j1}, \ldots, x^a_{jI}, \ldots, x^a_{jk}))^2
\end{aligned}
$$

Sobol' Estimator

The Sobol' method does not rely on ANY smoothness property of f.

The computational cost to estimate all first order sensitivity indices is equal to $N(k+1)$, ...

... whilst for computing the entire set we need $N(2^k - 1)$ model runs

Sobol' Estimator

Doubling the computational cost, i.e. using $N(2k+2)$, one can compute

- first order indices
- total-order indices

Saltelli's improvement of Sobol' ideas

Following the work of Saltelli (CPC 2002) we have that with $N(k+2)$ model runs one can compute

- One estimate for each of the k first order indices;
- One estimate for each of the k total-order indices;
- One estimate for each of the $\begin{pmatrix} k \\ 2 \end{pmatrix}$ V_{-ij} closed effect indices.

Saltelli's improvement of Sobol' ideas

with $N(2k+2)$ model runs one can compute (same as Classical Sobol')

- Double estimates for each of the k first order indices and for each of the k total-order indices;
- Double estimates for each of the $\left(\begin{array}{c} k \\ 2 \end{array} \right)$ V_{-ij} closed effect indices;
- Double estimates for each of the $\left(\begin{array}{c} k \\ 2 \end{array} \right)$ V_{ij} closed effect indices;

Comments on Sobol' and Saltelli approaches

Both for original Sobol' estimator and for the improvement proposed by Saltelli,

the computational cost still depends on the number of parameters

Replicated Latin Hypercube

As reviewed by Helton, Johnson, Sallaberry and Storlie (2006).

The r-LHS design of length $2N$ suffices to compute all first order indices

With extra Nk runs we are able to compute also the total order effects.

Replicated Latin Hypercube

We can summarize after Helton et al. review

- with $2N$ runs we compute first order indices;
- with $N(k+1)$ runs we compute total order indices;
- with $N(k+2)$ runs we compute both first and total order effects (i.e. same as Saltelli CPC 2002).

Lp_τ design

Generating a sample and a re-sample matrix using plain Sobol' Lp_τ sequences can improve convergence of MC estimates, while maintaining basic LHS properties:

- provide better space filling properties (low discrepancy);
- can be *easily* generated using their sequential properties.

Extended Sobol' estimator

We define a *new sensitivity* estimator which extends the basic Sobol' idea adding some minimal smoothness assumptions

$$\begin{aligned}\widetilde{V_I} &= V(E(y|x_I)) \\ &= \sum_{j=1}^{N} f(x^a_{j1},\ldots,x^a_{jI},\ldots,x^a_{jk}) f(x^b_{j'1},\ldots,x^b_{j'I},\ldots,x^b_{j'k}) \\ &\quad -E(y)^2\end{aligned}$$

Proximity assumption: first order

$$\begin{aligned}\widetilde{V_I} &= V(E(y|x_I)), I = \{i\} \\ &= \sum_{j=1}^{N} f(x^a_{j1}, \ldots, x^a_{jI}, \ldots, x^a_{jk}) f(x^b_{j'1}, \ldots, x^b_{j'I}, \ldots, x^b_{j'k})\end{aligned}$$

where $j' = j(a, b)$ is the row index of the *resample* matrix M_b whose entries in column i have the smallest distance from the jth row-index entries in column i of matrix M_a.

Proximity

The *re-sample* matrix is sorted according to minimum Euclidean distance of I co-ordinates to the *sample* matrix

$$M_a = \begin{pmatrix} x^a_{11} & \cdots & x^a_{1I} & \cdots & x^a_{1k} \\ x^a_{21} & \cdots & x^a_{2I} & \cdots & x^a_{2k} \\ \vdots & & \vdots & & \vdots \\ x^a_{N1} & \cdots & x^a_{NI} & \cdots & x^a_{Nk} \end{pmatrix},$$

$$M_b = \begin{pmatrix} x^b{}_{j'(I,1)1} & \cdots & x^b{}_{j'(I,1)I} & \cdots & x^b{}_{j'(I,1)k} \\ x^b{}_{j'(I,2)1} & \cdots & x^b{}_{j'(I,2)I} & \cdots & x^b{}_{j'(I,2)k} \\ \vdots & & \vdots & & \vdots \\ x^b{}_{j'(I,N)1} & \cdots & x^b{}_{j'(I,N)I} & \cdots & x^b{}_{j'(I,N)k} \end{pmatrix}$$

Proximity assumption: first order

If rLHS (with $r = 2$) or Lp_τ (with $N = 2^n$) this provides exactly the standard Sobol' estimator as in Helton et al. review

Proximity assumption: second order

$$\begin{aligned}\widetilde{V_I} &= V(E(y|x_I)), I = \{i_1, i_2\} \\ &= \sum_{j=1}^{N} f(x^a_{j1}, \ldots, x^a_{jI}, \ldots, x^a_{jk}) f(x^b_{j'1}, \ldots, x^b_{j'I}, \ldots, x^b_{j'k})\end{aligned}$$

where $j' = j(a, b)$ is the row index of the *re-sample* matrix M_b whose entries in columns i_1, i_2 have the smallest distance from the jth row-index entries in columns i_1, i_2 of matrix M_a.

Extended Sobol' estimator

Using such approach we aim to estimate with $2N$ model evaluations first and second order sensitivity indices.

Note that with Saltelli's improvements we need $N(2k+2)$ to jointly compute first and second order indices (as well as $(k-2)$th order and Total).

MC Experiments

We do MC experiments (25 replicas) to evaluate the new estimator w.r.t. to Saltelli (2002)

Case 1 Sobol' g function with 8 parameters

Case 2 Sobol' g function with 15 parameters

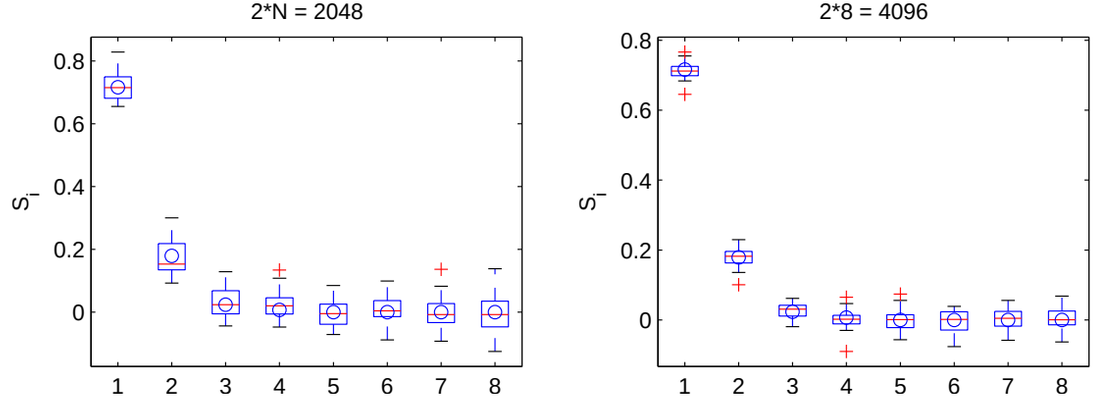
Case 3 Oakley and O'Hagan function with 15 parameters

Sobol' g function with 8 parameters

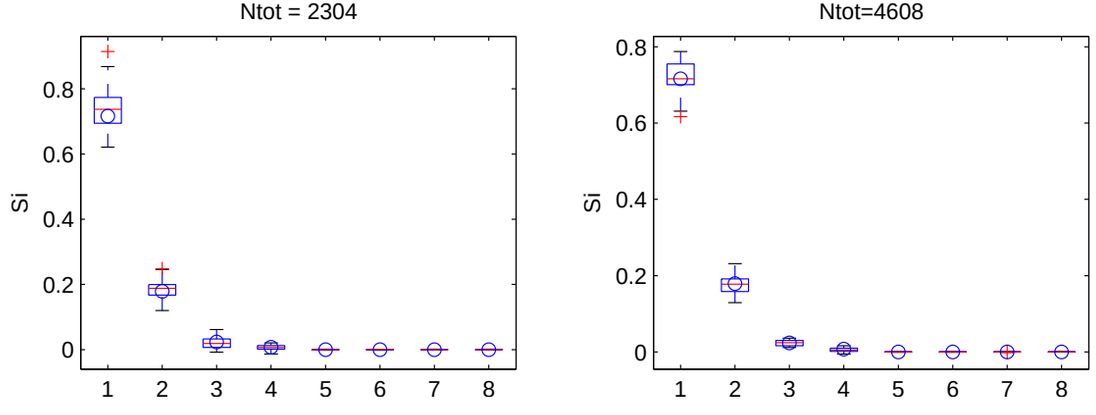
We consider the Sobol g function whose parameters are characterized by the array

$$a = (0, 1, 4.5, 9, 99, 99, 99, 99)$$

g-function with 8 parameters: approx. Sobol'



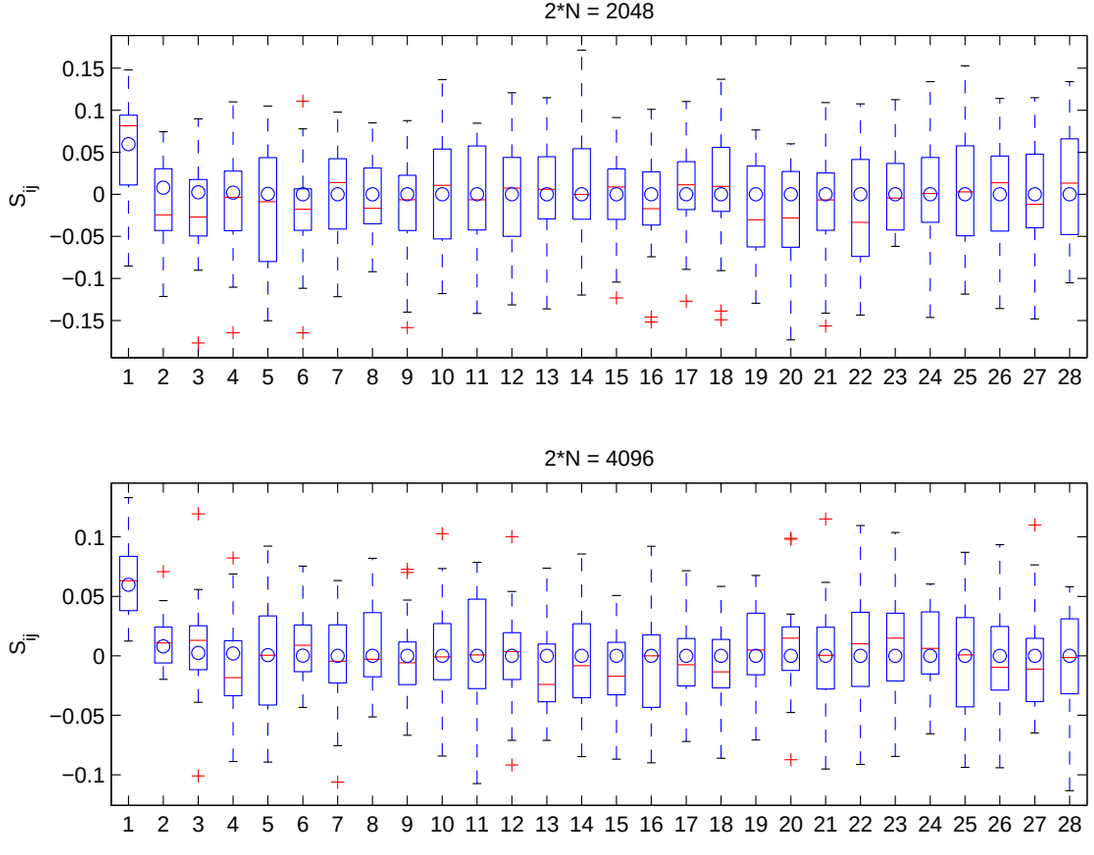
g-function with 8 parameters: Saltelli



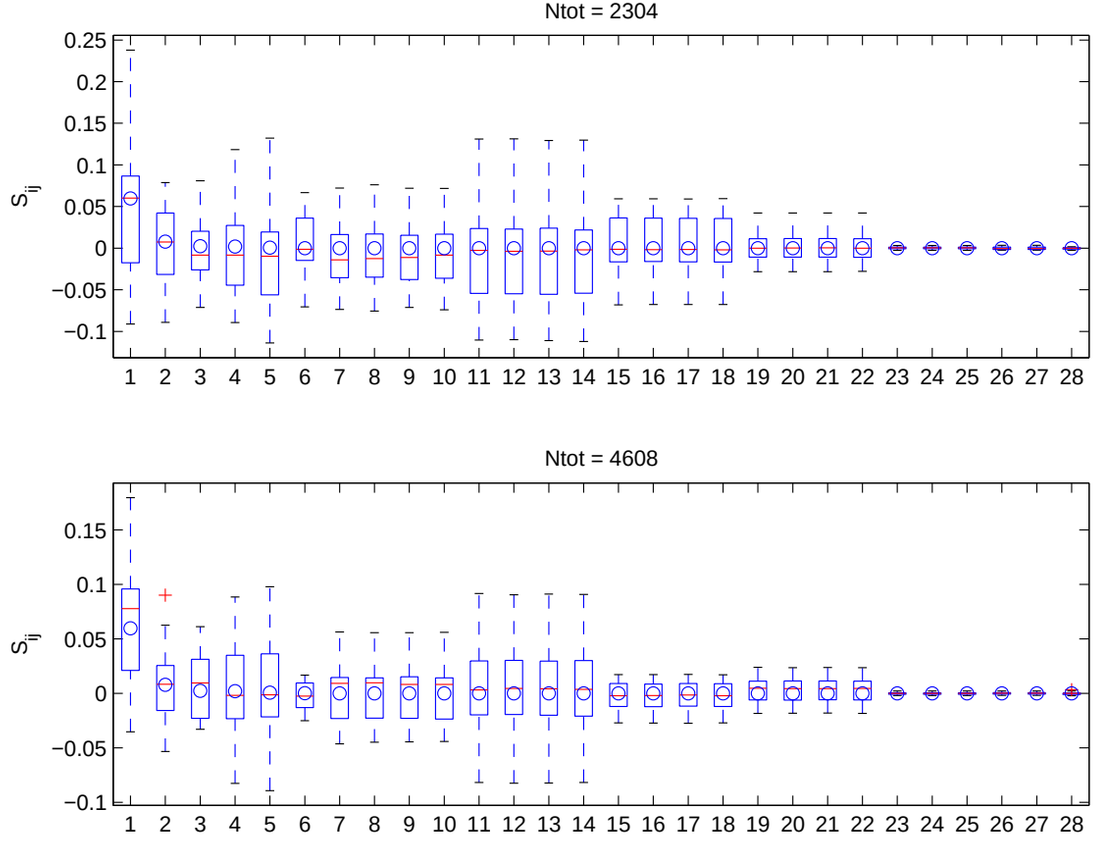
Sobol' g function with 8 parameters

With the same samples we have also computed the *second order indices*.

g-function with 8 parameters



g-function with 8 parameters



Sobol' g function with 15 parameters

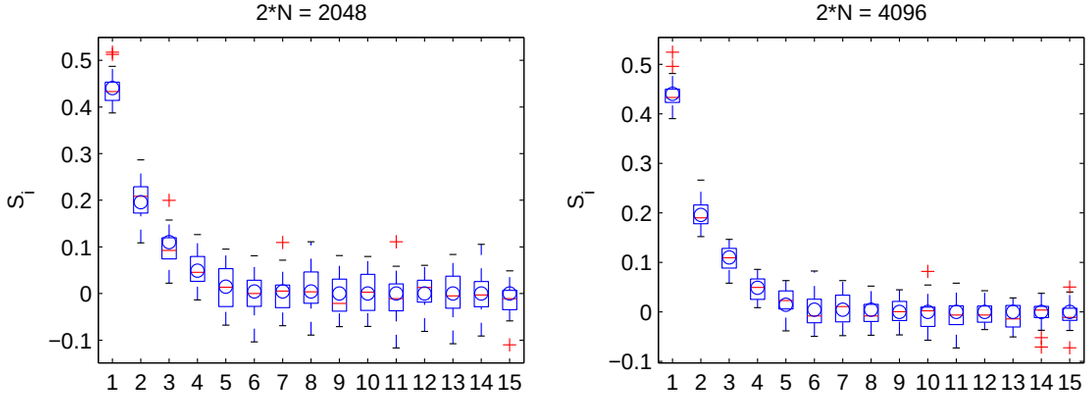
To appreciate the effect of increasing k, we consider the Sobol g function whose parameters are characterized by the array

$$a = (0, 0.5, 1, 2, 4.5, 9, 9, 9, 99, 99, 99, 99, 99, 99, 99)$$

Sobol' g function with 15 parameters

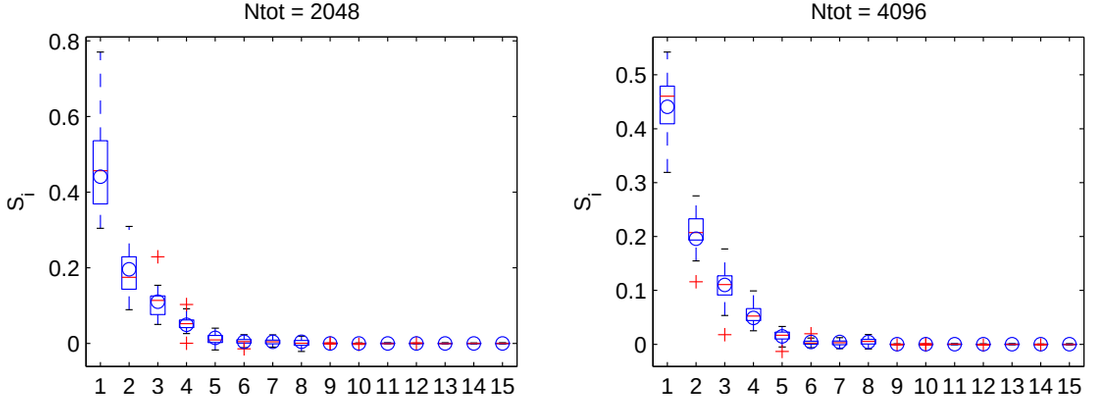
As we did for the 8 parameters g function we have run 25 MC replicas to estimate both *first and second order sensitivity indices*

Sobol' g function with 15 parameters



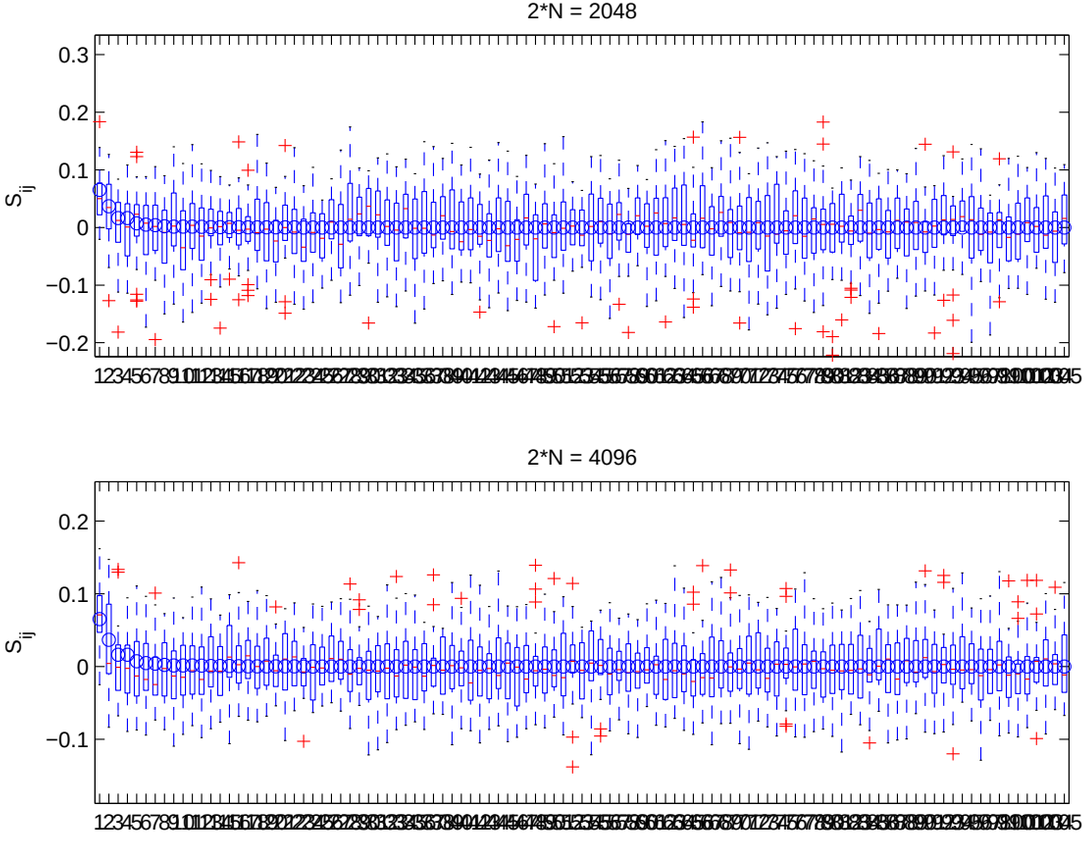
Approximated Sobol'

Sobol' g function with 15 parameters



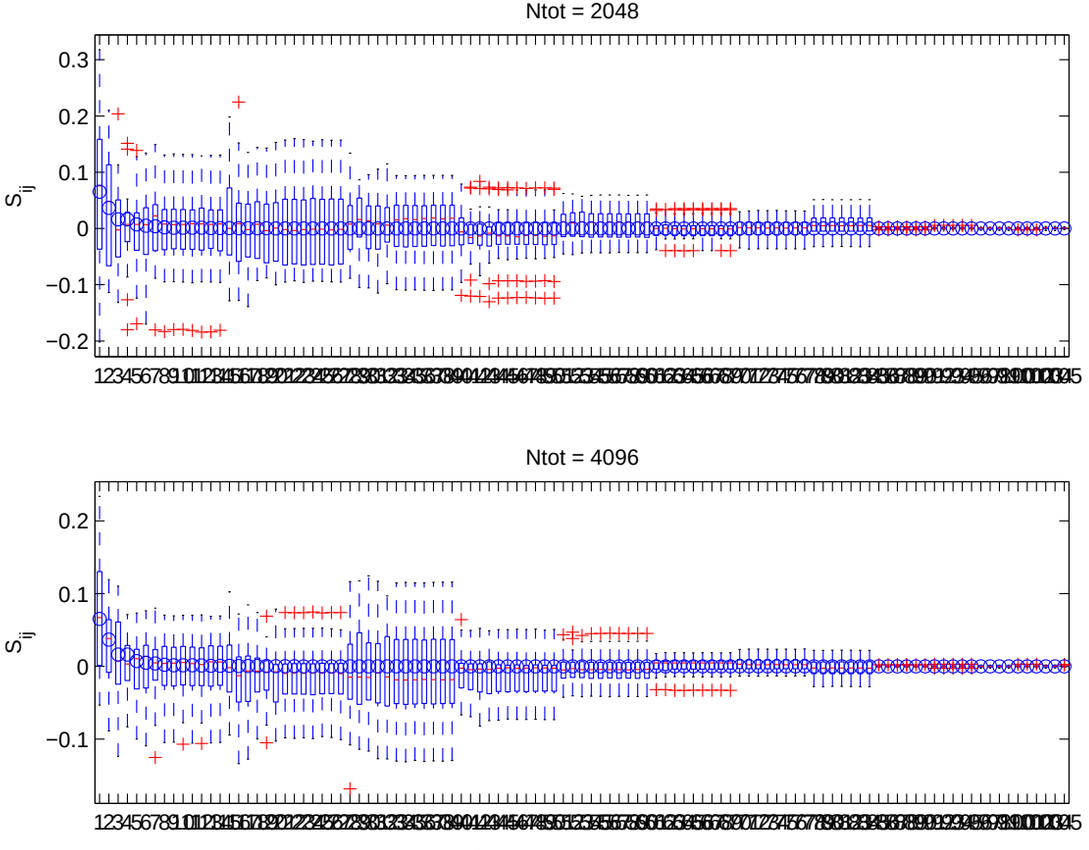
Saltelli (2002)

Sobol' g function with 15 parameters



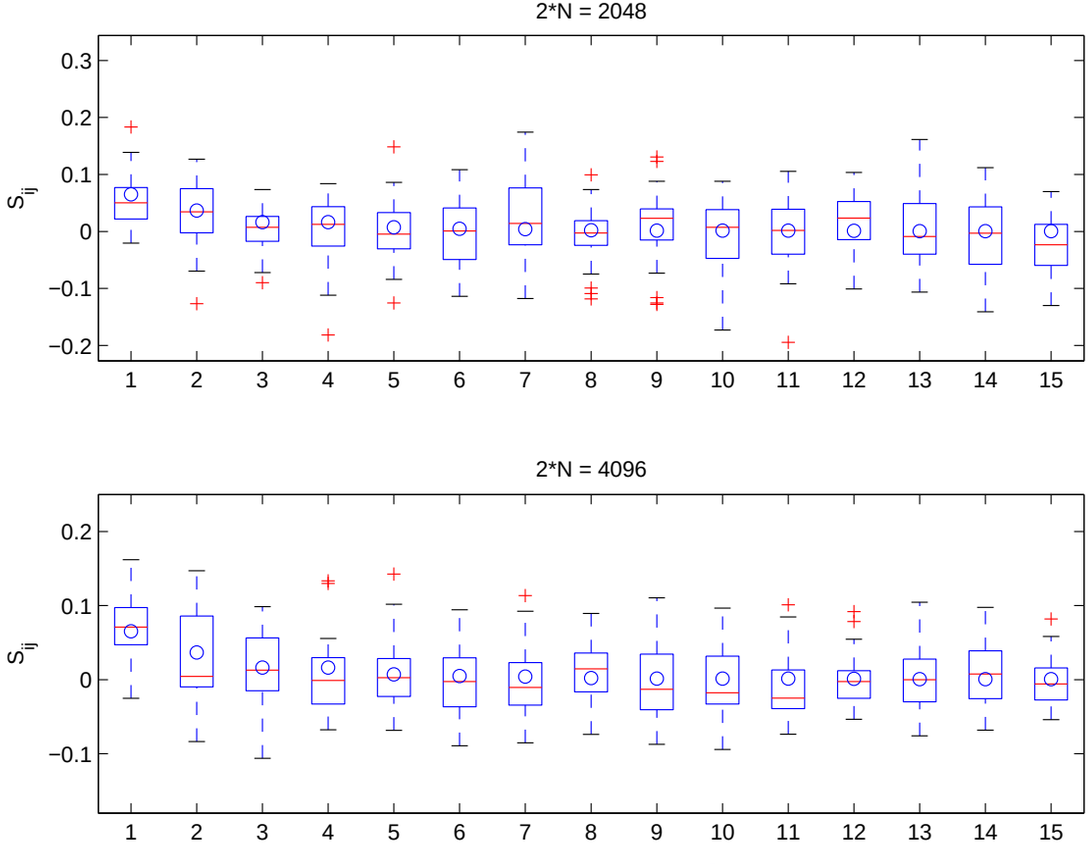
Second order, approximated Sobol'

Sobol' g function with 15 parameters



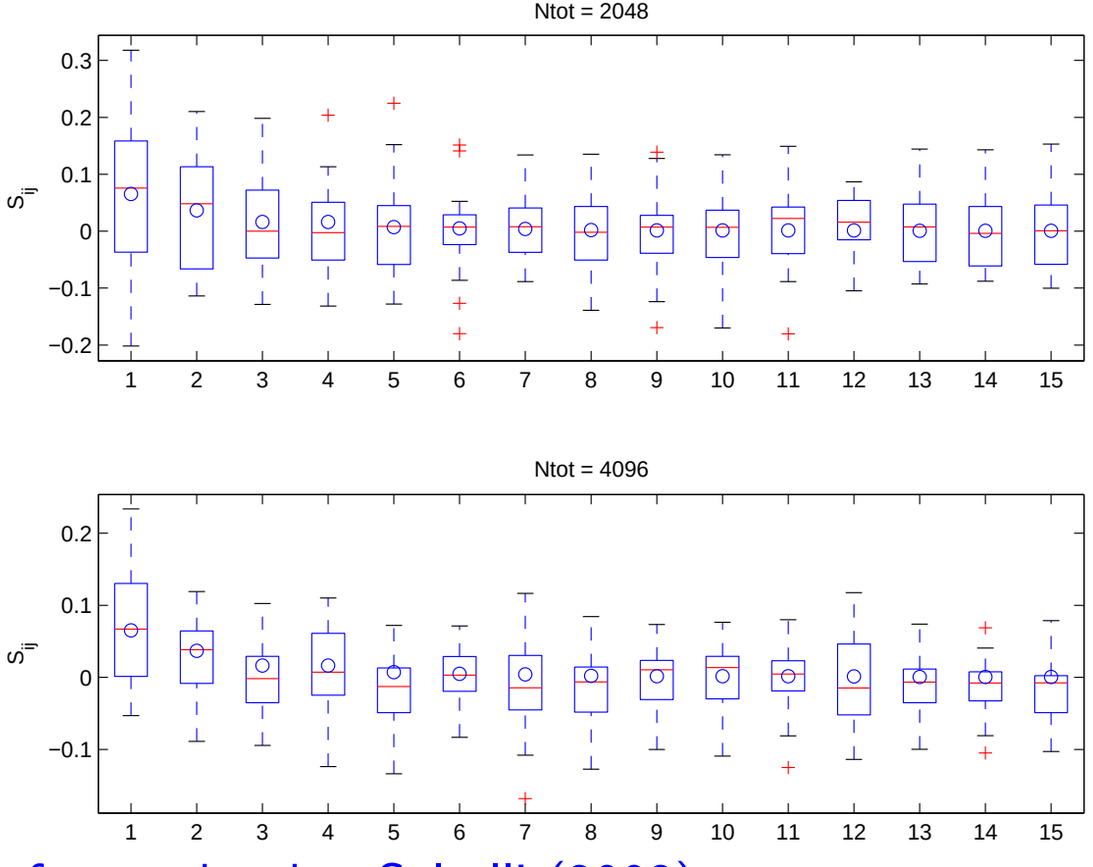
Second order, Saltelli (2002)

Sobol' g function with 15 parameters



Zoom of second order, approximated Sobol'

Sobol' g function with 15 parameters



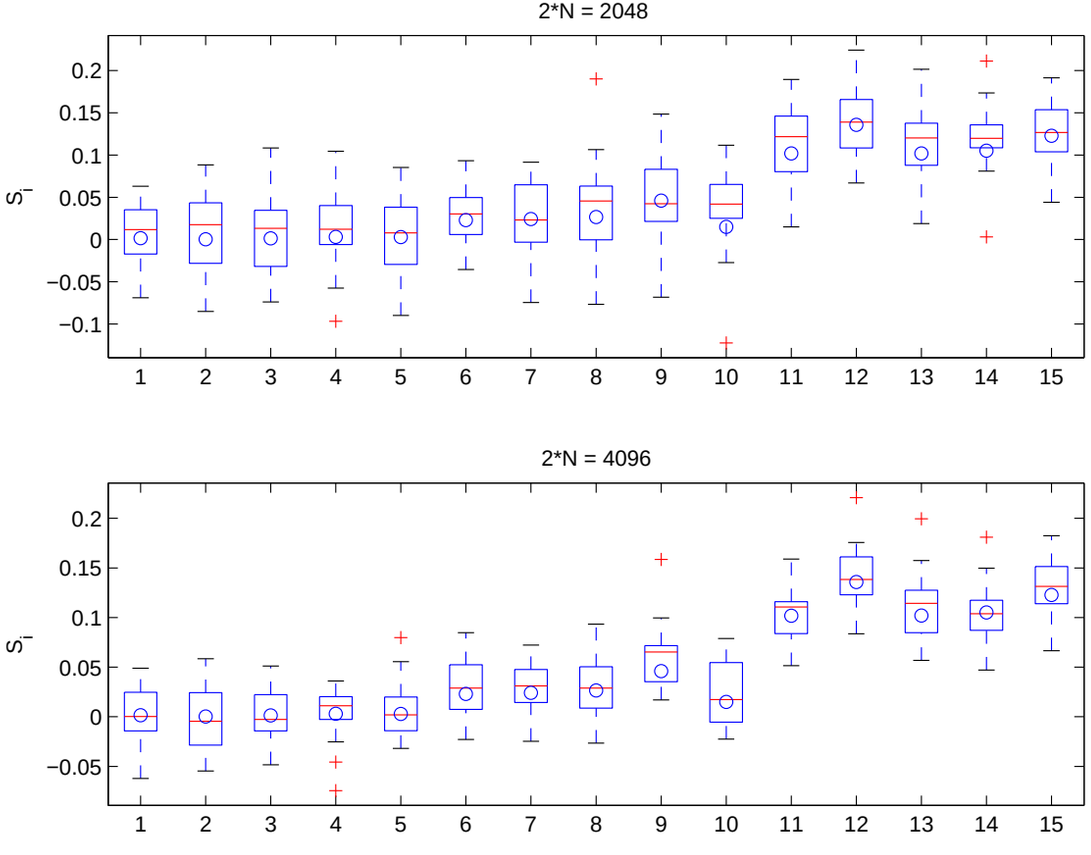
Zoom of second order, Saltelli (2002)

Oakley-o'Hagan test function

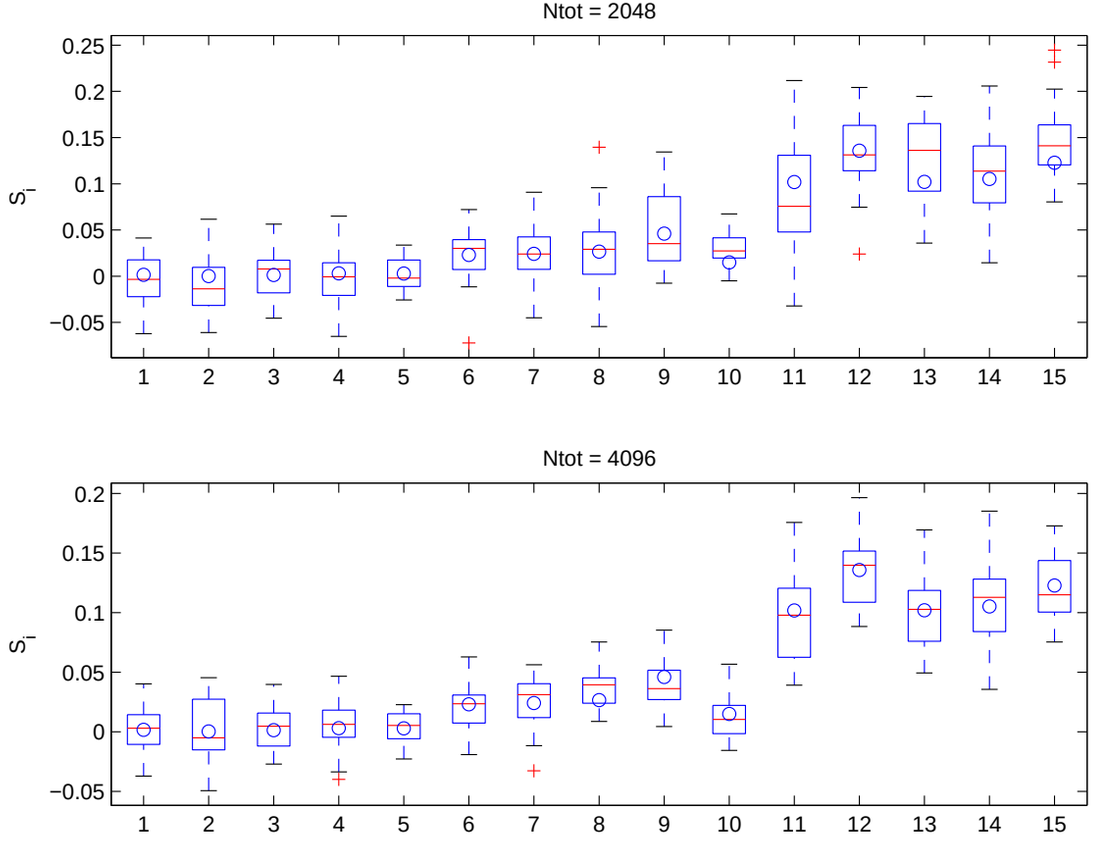
We consider the Oakley-o'Hagan test function as in Oakley and o'Hagan (2004)

For this function, we compute the *first order indices*

Oakley-o'Hagan test function: approx. Sobol'



Oakley-o'Hagan test function: Saltelli



Final remarks (1)

The proximity strategy allows to use all MC runs for all sensitivity indices, thus reducing the dependency on k, while adding some noise to the estimate; the larger k, the better it is w.r.t. standard Sobol' (Saltelli) estimator.

Final remarks (2)

The proximity strategy can be replaced (in particular if one is only interested in first order indices) with the following;

> instead of computing the nearest point in the *re-sample* matrix M_b one can re-order the *sample* matrix according to the increasing sorting given by the chosen index (or indices) and take the scalar product between sorted indices $f(1, \ldots, N-1) \cdot f(2, \ldots, N)$.

Hence, to compute first order indices we only need N model runs.

Final remarks (3)

First attempt to bridge the gap between sampling based and smoothing metamodelling techniques.

First result is to get rid of the link to k.

Potential for an extended use of the approximated Sobol' estimator in combination with smoothing/kriging metamodels.