Exercises in R on sensitivity analysis

Bertrand Iooss

Introduction

We study the flood model with its two model outputs: the overflowing height S and the annual cost of dyke maintenance Cp.

Edit the file TD4.R which first load the required packages and functions. This will be the main program. It initializes the nominal values of input variables of the flood function, as their standard deviation, minimum value and maximum value. An example file launch Fcrues.R, which contains the function of the flood model is given. Finally, EchantFcrues.R file that contains the function of the same name, to simulate samples of the input variables according to their probability law. You can edit the files and Fcrues.R EchantFcrues.R to understand their function.

Quadratic summation

Compute the importance factors of inputs on the overflow S by the quadratic summation formula. To calculate the derivatives, we can:
• Either use finite differences (for example with a perturbation of 1%);
• Either use the formal derivatives of the flood model using the functions deriv() and eval().

Morris method

Apply the method of Morris (on the outputs S and Cp) which can be found in the "sensitivity" package, with a number of levels equal to 4 and a number of repetitions of the OAT design equal to 10. For the lower and upper bounds of the non bounded laws, we take the 5% and 95%-quantiles. Interpret the results.

Correlation/regression based sensitivity indices

a) Simulate a random sample of the input vector x of size N = 100 by using the SampleFlood() function). Evaluate S and Cp model responses and Cp of this sample. For each output, plot on the same graph the scatterplots between the output and the 8 inputs of the model.

b) We want to test the validity of the linear relationship output / input. Write the linear formula:
   as.formula formula = (z ~ Q + Ks + Zv + Zm + Hd + Cb + B + L)
   and perform a linear regression between each of the two outputs and the 8 inputs:
   model = lm (formula, data = data.frame (x, y = z))

Analyze this linear model for S and Cp (functions print(), summary() and plot()). For each output, give the value of the R^2 and show the graph: observed values vs. predicted values. Are the relationships linear? If yes, calculate the standardized regression coefficients (SRC). Calculate the sensitivity indices SRC^2. Compare with

sensitivity indices from quadratic summation.

Sobol indices

a) Using the function sobol2002() of the package "sensitivity", compute Sobol indices on the output Cp. This function takes as input 2 independent Monte Carlo samples $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ of size 10000. It is recommended to increment the parameter nboot in order to have some confidence intervals on the estimators. Analyze the results with the print() and plot() functions. If the results do not seem satisfactory, increase the size of the Monte Carlo samples to increase the accuracy of the indices (cost problems in time computation and memory may arise).

We recall the formula for estimating the numerator of the Sobol index of first order:

$$\hat{D}_i = \frac{1}{S}\sum_{k=1}^{S} f(X_{k,1}^{(2)},\ldots,X_{k,p}^{(2)})\, f(X_{k,1}^{(1)},\ldots,X_{k,i-1}^{(1)},X_{k,i}^{(2)},X_{k,i+1}^{(1)},\ldots,X_{k,p}^{(1)}) - \hat{D}_0{}^2 .$$

The numerator of the total Sobol index is estimated by:

$$\hat{D}_{\sim i} = \frac{1}{S}\sum_{k=1}^{S} f(X_{k,1}^{(1)},\ldots,X_{k,p}^{(1)})\, f(X_{k,1}^{(1)},\ldots,X_{k,i-1}^{(1)},X_{k,i}^{(2)},X_{k,i+1}^{(1)},\ldots,X_{k,p}^{(1)}) - \hat{D}_0{}^2 .$$

b) Perform the same analysis with the functions sobol2007() and soboljansen() that encode more accurate formulas for the numerator of the Sobol indices, respectively

$$\hat{D}_i = \frac{1}{S}\sum_{k=1}^{S} f(X_{k,1}^{(2)},\ldots,X_{k,p}^{(2)})\left[f(X_{k,1}^{(1)},\ldots,X_{k,i-1}^{(1)},X_{k,i}^{(2)},X_{k,i+1}^{(1)},\ldots,X_{k,p}^{(1)}) - f(X_{k,1}^{(1)},\ldots,X_{k,p}^{(1)})\right],$$

$$\hat{D}_{\sim i} = Var(Y) - \frac{1}{S}\sum_{k=1}^{S} f(X_{k,1}^{(1)},\ldots,X_{k,p}^{(1)})\left[f(X_{k,1}^{(1)},\ldots,X_{k,p}^{(1)}) - f(X_{k,1}^{(1)},\ldots,X_{k,i-1}^{(1)},X_{k,i}^{(2)},X_{k,i+1}^{(1)},\ldots,X_{k,p}^{(1)})\right]$$

and

$$\hat{D}_i = Var(Y) - \frac{1}{2S}\sum_{k=1}^{S}\left[f(X_{k,1}^{(2)},\ldots,X_{k,p}^{(2)}) - f(X_{k,1}^{(1)},\ldots,X_{k,i-1}^{(1)},X_{k,i}^{(2)},X_{k,i+1}^{(1)},\ldots,X_{k,p}^{(1)})\right]^2$$

$$\hat{D}_{\sim i} = Var(Y) - \frac{1}{2S}\sum_{k=1}^{S}\left[f(X_{k,1}^{(1)},\ldots,X_{k,p}^{(1)}) - f(X_{k,1}^{(1)},\ldots,X_{k,i-1}^{(1)},X_{k,i}^{(2)},X_{k,i+1}^{(1)},\ldots,X_{k,p}^{(1)})\right]^2 .$$

c) The convergence of the Sobol indices are very slow with Monte Carlo samples; change the type of the samples in the input function soboljansen() by taking a sample size of 1000 and a quasi-Monte Carlo sequence (sobol() function of package randtoolbox). The only constraint is that these two samples must be independent (one trick is to be found) and simulated variables follow the required probability laws (see file SampleFlood.R). It will transform the variables U[0,1] in their respective law by:

```
X[,1] = qtgumbel(X[,1],loc=1013.0,scale=558.0,min=500,max=3000)
X[,2] = qtnorm(X[,2],mean=30.0,sd=8,min=15.);
X[,3] = qtriangle(X[,3],a=49,b=51,c=50);
X[,4] = qtriangle(X[,4],a=54,b=56,c=55);
X[,5] = qunif(X[,5],min=7,max=9);
X[,6] = qtriangle(X[,6],a=55,b=56,c=55.5);
X[,7] = qtriangle(X[,7],a=4990,b=5010,c=5000);
X[,8] = qtriangle(X[,8],a=295,b=305,c=300);
```