Design of Experiments and Sensitivity analysis

Course and practical application

eX Modelo Summer School

OpenMOLE

June 24, 2019

- Interactive model exploration by hand and the need for preliminary experiments
- The Design of Experiments (DOE) as the definition of computational experiments to extract information from the simulation model
- Example: NetLogo behavior space: basic grid DOE
- Sensitivity analysis as an advanced DOE

Remark 1: *terminology strongly depends on disciplines and practices*

Remark 2: *most are generally **preliminary experiments** to prepare more elaborated, question-related, experiments*

Outline

1. Basic experiments

2. High-dimensional samplings

3. Sensitivity analysis

Basic experiments

Provide explicitly sampling points on which the model (or its replication task) will be run: notion of ***direct sampling*** *in OpenMOLE (corresponds to DOE in the literature)*

- full samplings
- elaborated sampling for high dimensions given a low computational budget (**the curse of dimensionality**)

Syntax of the direct sampling method in OpenMOLE:

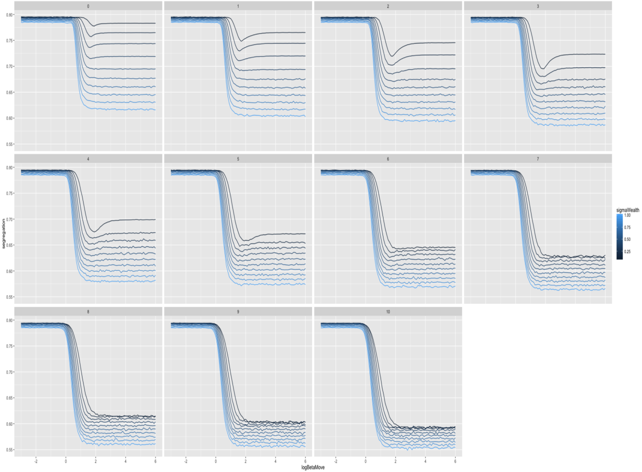
```
val explo = DirectSampling(
  evaluation = (model hook reshook on env by 500),
  sampling =
    (humanFollowProbability in (0.0 to 1.0 by 0.1))
    x (humanInformedRatio in (0.0 to 1.0 by 0.1))
    x (humanInformProbability in (0.0 to 1.0 by 0.1))
    x (replication in UniformDistribution[Long](100000) take 100)
 )
```

One factor at a time

Cheapest and intuitive DOE: *all factors have nominal values and a discrete variation set, in which each is varied while others remaining fixed*

- when model is slow - or computational budget highly limited
- does not capture interaction between parameters, and highly dependent on nominal values
- seen as a bad practice **BUT** useful for models taking significant time, and prone to thematic interpretation

Example where One-At-a-Time fails



Indicator variations in a 3D parameter space: some nominal values make non-monotonous effects disappear

Grid sampling

Brute force DOE: *ensemble product of discrete variation ranges for factors (usually a regular grid but not necessarily)*

- quickly limited by the curse of dimensionality - in practice still powerful with a quick model and a low number of parameters
- naive approach, but remains only DOE for many "simulation-newcomers" disciplines

One-factor sampling:

```
sampling = OneFactorSampling(
    (x1 in (0.0 to 1.0 by 0.2)) nominal 0.5,
    (x2 in (0.0 to 1.0 by 0.2)) nominal 0.5
)
```

Grid sampling:

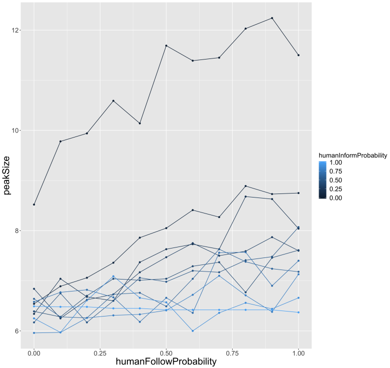
```
sampling =
     (x1 in (0.0 to 1.0 by 0.5)) x
     (x2 in (0.0 to 1.0 by 0.5))
```

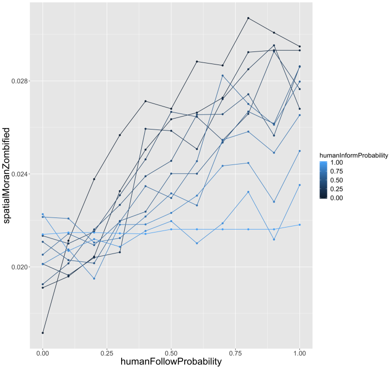
Practical application

- Given the described zombie model, what first experiment beyond stochasticity would be relevant ?
- Explore and test the `directsampling.oms` script available in the downloaded archive (see chat for link)

Example of direct sampling results

Regular grid sampling for the three parameters of the basic ZOMBIE model, with 100 replications





High-dimensional samplings

Computational limitations $\implies$ *need specific methods to efficiently sample the parameter space*

Different methods for improving sampling in numerical experiments given limited computational resources have been proposed, as for example:

- Sobol sequences (quicker convergence of for Monte Carlo estimation of integrals)
- Latin Hypercube Sampling
- Orthogonal sampling

Minimizing discrepancy for a point cloud: intuitively being spread evenly across the definition space

L2-discrepancy given for normalized data points $\mathbf{X} = (x_{ij}) \in [0,1]^d$ by

$$\left\| \mathbf{t} = (t_j) \in [0,1]^d \mapsto \frac{1}{n} \sum_i \mathbb{1}_{\prod_j x_{ij} < t_j} - \prod_j t_j \right\|_2$$

Explanation: $\prod_j t_j$ *is the volume of the hypercube between* $\mathbf{t}$ *and the origin; the sum of indicator functions counts the points within that hypercube; the difference between expected volume and point number is integrated over the whole hypercube.*

Latin Hypercube Sampling

X				
	X			
				X
			X	
		X		

Latin cube: one point in each row and column; hypercube generalization in any dimension

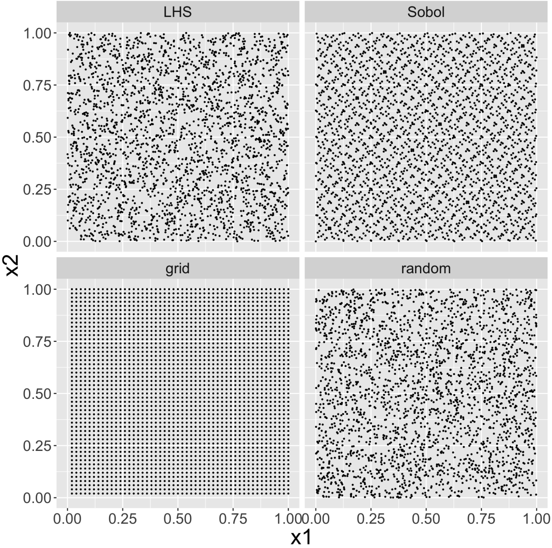
Sobol sequences are a case of quasi-random sequences with low discrepancy (also Halton sequences e.g.)

- Estimate integrals in $1/N$ instead of $1/\sqrt{N}$ with random sampling
- Constructed recursively (using bit representations)

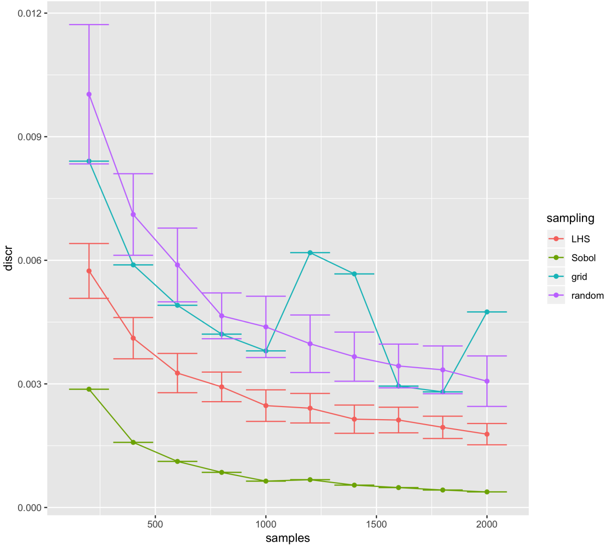
Comparison of samplings

OpenMOLE

For $N = 2500$ samples in 2 dimensions



Estimated discrepancies for repetitions of samplings as a function of sample size



LHS Sampling

```
sampling = LHS(
    100,
    x1 in (0.0,1.0),
    x2 in (0.0,1.0)
)
```

Sobol sampling

```
sampling = SobolSampling(
     100,
    x1 in (0.0,1.0),
    x2 in (0.0,1.0)
)
```

→ **Practice:** test these samplings in `directsampling.oms`

Summary of samplings characteristics

	Coverage	**Interpretability**	**Budget**
One factor at a time	✗	✓	✓
Complete plan	✓	✓	✗
LHS/Sobol	✓	✗	✓

Aim of sensitivity analysis methods *How to summarize model sensitivity and isolate principal factors ?*

- Most methods are *global*, i.e. provide an aggregate of factor effect on the full parameter space
- Advanced methods, still useful for preliminary experiments e.g. to discard factors from further experiments
- Examples: Morris and Saltelli methods

Morris method

Idea: *Sample trajectories in the parameter space in a One-At-a-Time manner. Screening method isolating* ***elementary effects***

- isolate local effects of factors
- more efficient than point sampling to get individual effects
- useful as a first experiment to understand the relative influence of factors

Introduced by [Morris, 1991], improved by [Saltelli et al., 2004], [Campolongo et al., 2011] propose to extend the method with Sobol sequences

Let δ be step for parameter variation (all assumed normalized in $[0;1]$), the *elementary effect* for parameter i on output Y at point $\vec{x}$ is given by

$$\varepsilon_i(\vec{x}) = \frac{Y(\vec{x} + \delta \cdot \vec{e}_i) - Y(\vec{x})}{\delta}$$

With N parameter trajectories randomly sampled (each trajectory varying all parameters), the sensitivity index is given by

$$\mu_i = \frac{\sum_{k=1}^{N} \varepsilon_i(\vec{x}_k)}{N}$$

and complementary indices by

$$\sigma_i = \frac{\sum_{k=1}^{N} (\varepsilon_i(\vec{x}_k) - \mu_i)^2}{N}$$

$$\mu_i^* = \frac{\sum_{k=1}^{N} |\varepsilon_i(\vec{x}_k)|}{N}$$

In OpenMOLE, Morris is a method in itself (and not a sampling)

```
SensitivityMorris(
    evaluation = (model on env by 5000),
    inputs = List(
       humanFollowProbability in (0.0,1.0),
       humanInformedRatio in (0.0,1.0),
       humanInformProbability in (0.0,1.0)
    ),
    outputs = List(totalZombified,halfZombified),
    repetitions = 1000,
    levels = 20
) hook CSVHook(workDirectory / "morris_result.csv")
```

→ **Practice:** explore the script `morris.oms`, comment the results obtained with a large-scale experiment `morrisresults`

Method based on the estimation of conditional relative variances [Saltelli et al., 2010]

First order index

$$S_i = \frac{Var\left[E_{\mathbf{X}_{\sim i}}(Y|X_i)\right]}{Var(Y)}$$

is the expected relative variance reduction if X_i would be fixed

Total effect index

$$ST_i = \frac{E_{\mathbf{X}_{\sim i}}\left[Var(Y|\mathbf{X}_{\sim i})\right]}{Var(Y)}$$

is the expected relative variance if all factors but X_i are fixed (includes interaction effects)

In OpenMOLE, Saltelli is also a method

```
val sen = SensitivitySaltelli(
  evaluation = (model on env by 1000),
  samples = 100000,
  inputs = List(
    humanFollowProbability in (0.0,1.0),
    humanInformedRatio in (0.0,1.0),
    humanInformProbability in (0.0,1.0)
  ),
  outputs = List(totalZombified,halfZombified)
)

sen hook SaltelliHook(sen,workDirectory)
```

Summary of sensitivity methods

	Coverage	Interpretability	Budget
Morris	✗	✓	✓
Saltelli	✓	✓	✗

Conclusion

OpenMOLE

Take-home messages:

- Direct sampling can be useful as preliminary experiments, but also experiments in themselves
- Sensitivity analysis methods are useful for a global knowledge on influence of factors
- Find a good balance interpretability/computational budget/information extracted
- The experiments you choose depend on your questions but also on your discipline

References I

Campolongo, F., Saltelli, A., and Cariboni, J. (2011).
From screening to quantitative sensitivity analysis. a unified approach.
Computer Physics Communications, 182(4):978–988.

Morris, M. D. (1991).
Factorial sampling plans for preliminary computational experiments.
Technometrics, 33(2):161–174.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S. (2010).
Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index.
Computer Physics Communications, 181(2):259–270.

References II

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M. (2004).
Sensitivity analysis in practice: a guide to assessing scientific models.
Chichester, England.