Analyse the uncertainty of your system

Sensitivity Analysis in Python with `scipy.stats.sobol_indices`

?

How should you measure the intrinsic variability of a system?

!

Tempted to use derivatives around a design point to gauge uncertainty?
Don't! The indices of Sobol', a variance-based Global Sensitivity Analysis method, are more powerful.

The variance can be decomposed along each parameters:

$$S_i = \frac{\mathbb{V}_i(Y)}{\mathbb{V}[Y]} \qquad S_{ij} = \frac{\mathbb{V}_{ij}(Y)}{\mathbb{V}[Y]} \qquad S_{T_i} = S_i + \sum_j S_{ij} + \sum_{j,k} S_{ijk} + \ldots = 1 - \frac{\mathbb{V}[\mathbb{E}(Y|x_{\sim i})]}{\mathbb{V}[Y]}$$

S_i First/second order indices: individual/interaction contribution of the parameters

S_{T_i} Total order indices: global contribution of the parameters

Indices of Sobol', and other powerful methods, are also available in the specialized library SALib:

- FAST
- Moment independent
- PAWN
- HDMR, ...

Simple API

SciPy 1.11

1 Define your function

$$Y(\mathbf{x}) = \sin x_1 + 7 \sin^2 x_2 + 0.1 x_3^4 \sin x_1 \quad \mathbf{x} \in [-\pi, \pi]^3$$

```
import numpy as np
def f_ishigami(x):
    return (
        np.sin(x[0])
        + 7 * np.sin(x[1])*$2
        + 0.1 * (x[2]*$4) * np.sin(x[0])
    )
```

2 Define how each input is distributed

```
import scipy as sp
dists=[
    sp.stats.uniform(loc=-np.pi, scale=2*np.pi),
    sp.stats.uniform(loc=-np.pi, scale=2*np.pi),
    sp.stats.uniform(loc=-np.pi, scale=2*np.pi)
]
```

3 Compute the indices of Sobol'

```
indices = sp.stats.sobol_indices(
    func=f_ishigami, n=1024,
    dists=dists,
    random_state=rng
)
indices.first_order
array([0.31637954, 0.43781162, 0.00318825])
indices.total_order
array([0.56122127, 0.44287857, 0.24229595])
```

4 Use this new insight on your problem!

Pamphile T. Roy tupui

Quansight