Estimating a Probability of Failure with an Adaptive Algorithm

Lucie Bernard & Arnaud Guyader (LPSM - Sorbonne Université) & Philippe Leduc (STMicroelectronics - Tours) & Florent Malrieu (LMPT - Univ. Tours)

1. Framework & Objectives

Let us consider:
- A cube $A = [a, b]^d$.
- An expensive-to-evaluate black-box function $g : A \rightarrow \mathbb{R}$.
- A random variable $X \sim P_x$ on $A$.
- A threshold $T \in \mathbb{R}$.

We want to provide an estimation of the small probability of failure $p$ defined by

$$p = P(g(X) > T) = P_X(S),$$

where $S = \{x \in A, g(x) > T\}$, performing at most $n$ evaluations of $g$.

### Main assumptions

1. **Level set condition**

   We assume that
   $$\lambda(\{x \in A \mid |g(x) - T| \leq \delta\}) \leq L \delta, \quad \delta > 0,$$
   where $\lambda(E)$ is the Lebesgue measure of a set $E$. See [1].

2. **Lipschitz condition**

   We assume that $g$ is Lipschitz with a known constant $M > 0$, meaning that for all $(x, x') \in A \times A$, we have
   $$|g(x) - g(x')| \leq M |x - x'|_\infty.$$

#### Implementation of the algorithm

The algorithm defines a decreasing sequence $(A_k)_{k \geq 0}$, such that $A_k$ is the set of all cubes $Q$ of side length $2^{-k}|b - a|$ where $g$ is still likely to exceed $T$. It also defines increasing sequences $(S_k)_{k \geq 0}$ and $(O_k)_{k \geq 0}$ such that $S_0$ (resp. $O_0$) is the set of all cubes $Q \in A_k$ where $g$ is always above (resp. below) $T$. It stops when $n$ evaluations of $g$ have been performed.

- **Step $k = 0$**

  We defined $A_0 = \{x\}$ and $S_0 = O_0 = \emptyset$.

- **Step $k = 1, 2, \ldots$**

  For all cube $Q \in A_{k-1}$, we evaluate $g$ at the center $c_Q$ of $Q$.

  - We define $S_{k-1}$ (resp. $O_{k-1}$) as the set of all cubes $Q \in A_{k-1}$ for which
    $$g(c_Q) - 2^{-k}M|b - a| > T$$
    (resp. $g(c_Q) + 2^{-k}M|b - a| \leq T$), that is the set of all cubes $Q \in A_{k-1}$ where $g$ is always above (resp. below) $T$.

  - We define $A_{k-1} = A_{k-1} \setminus (S_{k-1} \cup O_{k-1})$, that is the set of all cubes $Q \in A_{k-1}$ where $g$ is still likely to exceed $T$.

  Then, we define:
  - $S_k = S_{k-1} \cup S_{k-1}$ (resp. $O_k = O_{k-1} \cup O_{k-1}$) as the set of all cubes $Q \in A_k$ where $g$ is always above (resp. below) $T$.
  - $A_k$ as the set of all cubes of side length $2^{-k}|b - a|$, which are the children of $A_{k-1}$, so that $g(A_k) = 2^{-k}|\#A_{k-1}|$.

  - At the end of step $k$, we have the upper-bound $p_k \geq p$ defined by:
    $$p_k = P_X(A_k) + P_S(S_k).$$

2. **Improvement of the algorithm in practice**

   - At step $k \geq 1$, one can take into account of all information provided by the Lipschitz condition and extract, for any $Q \in A_{k-1}$, an hyper rectangle $H_Q$ of center $c_Q$, where $g$ is always above of below $T$.

   - At the end of step $k$, we have the upper-bound $p_k \geq p$ defined by:
     $$p_k = P_X(A_k) + P_S(S_k).$$

3. **Some remarks**

   1. At the end of step $k \geq 2$, the number $n_k$ of evaluations of $g$ satisfies:
      $$n_k = 1 + 2^d \sum_{j=1}^k \#(A_{j-1})$$
      $$\leq \begin{cases} 
      1 + LM |b - a|2(k - 1) & \text{if } d = 1, \\
      1 + LM |b - a|2^2(2^d - 1) - 2^2(2^d - 1) & \text{if } d \geq 2.
      \end{cases}$$

   2. For all $k \geq 0$, we have:
      $$P(X \in S_k) \leq p \leq P(X \in A_k) + P(X \in S_k) = p_k.$$

   3. Let $f_k$ be the density of $X$ and $m = \text{sup}_{x \in A} f_k(x)$. For all $k \geq 1$, we have:
      $$P(X \in A_k) \leq m |b - a|^{d+1} LM \times 2^{1-d}.$$

In case of $P_X$ has a density only known up to a normalizing constant, we use an Adaptive Multilevel Splitting method, also called Subset Simulation (see e.g. [2, 3]) to give an estimation of $p_k$. The principle is as follows:

- For $k \geq 1$, let $Q_k$ be a cube in $A_k$ and $Q_{k-1}$ be its parent, with $Q_0 = A$ and $P(X \in Q_0) = 1$.

- Since we have
  $$P(X \in Q_k) = P(X \in Q_k | X \in Q_{k-1}) P(X \in Q_{k-1}),$$
  then the probability $P(X \in Q_1)$ can be easily estimated.

- The law $P_X$ has a density which is known up to a normalizing constant, so that Metropolis-Hasting algorithm techniques can be applied to draw a Monte Carlo sample $X_1, \ldots, X_{n_k}$ from the restriction of $P_X$ to $Q_{k-1}$ (i.e. from $L(X | X \in Q_{k-1})$).

- At the beginning of the step $k \geq 1$, the probability $P(X \in Q_{k-1})$ has already been estimated.

Finally, the precision of the estimation of $p$ only depends on the budget $n$ and the size $N$ of the Monte Carlo sample.

### References

