Soft level splitting for rare event estimation

F. Cerou[1] A. Guyader[2] N. Hengartner[3]

[1]Inria Rennes Bretagne Atlantique [2]Université Rennes 2

[3]Los Alamos National Laboratory

RESIM 2012
June 25-27, Trondheim

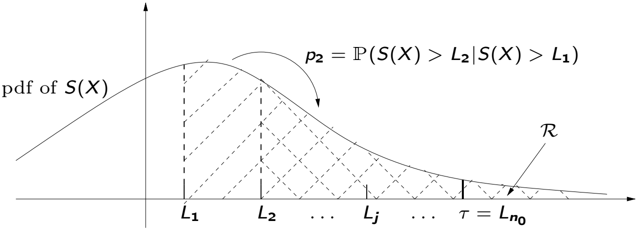
The problem

$X \sim \mu$, with μ probability measure on E ($\mathbb{R}^d$, or a subset)
We know how to draw samples from μ
Given a function $S : E \longmapsto \mathbb{R}$, we look at the rare event

$$\mathcal{R} = \{S(X) > \tau\}$$

We want to compute $\mu(\mathcal{R}) = \mathbb{P}(X \in \mathcal{R})$, and draw samples from $\mu_{\mathcal{R}}(dx) = \frac{1}{\mu(\mathcal{R})} \mathbb{1}_{\mathcal{R}}(x)\mu(dx)$

Main Idea



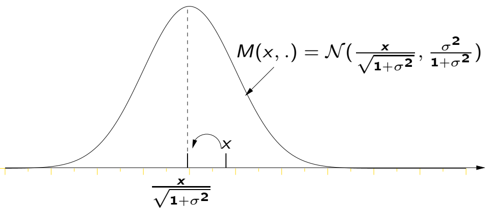
- **Ingredients**: fix n_0 and $L_1 < \cdots < L_{n_0} = \tau$ so that each $p_j = \mathbb{P}(S(X) > L_j | S(X) > L_{j-1})$ is not too small.
- **Bayes decomposition**: $\alpha = p_1 p_2 \dots p_{n_0}$.
- **Unreasonable assumption**: suppose we can estimate each p_j independently with usual Monte-Carlo: $p_j \approx \hat{p}_j = N_j / N$.
- **Multilevel Estimator**: $\hat{\alpha}_N = \hat{p}_1 \hat{p}_2 \dots \hat{p}_{n_0}$.

The Shaker

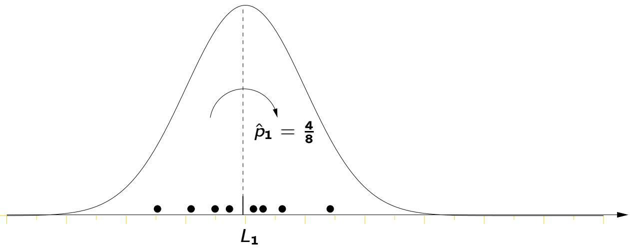
- **Recall**: $X \sim \mu$ on E.
- **Ingredient**: a $\mu-$reversible transition kernel $M(x, dx')$ on E:

$$\forall (x, x') \in E^2 \qquad \mu(dx)M(x, dx') = \mu(dx')M(x', dx).$$

- **Consequence**: $\mu M = \mu$.
- **Example**: if $X \sim \mathcal{N}(0,1)$ then $X' = \frac{X+\sigma W}{\sqrt{1+\sigma^2}} \sim \mathcal{N}(0,1)$, i.e. $M(x, dx') \sim \mathcal{N}(\frac{x}{\sqrt{1+\sigma^2}}, \frac{\sigma^2}{1+\sigma^2})(dx')$ is a "good shaker".



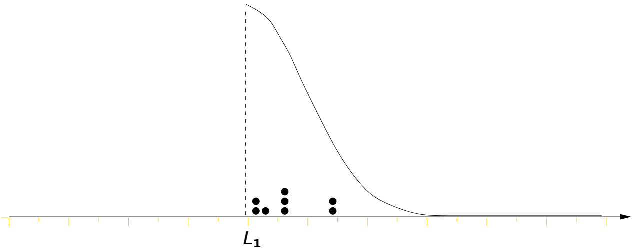
A Selection/Mutation Algorithm



- **Initialization**: Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- **Selection**: $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- **Mutation**: $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

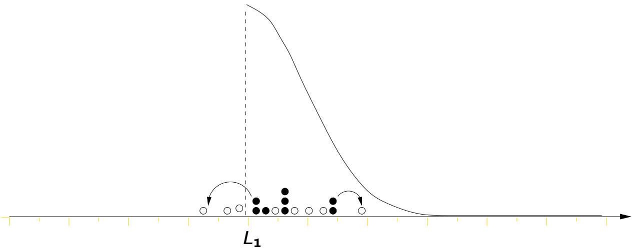
A Selection/Mutation Algorithm



- **Initialization**: Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- **Selection**: $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- **Mutation**: $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

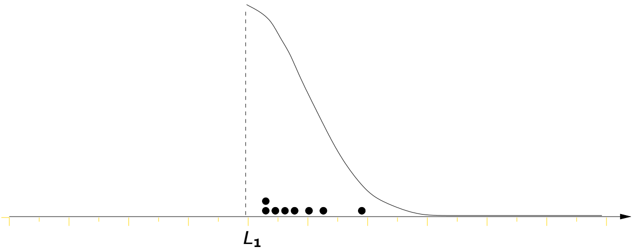
A Selection/Mutation Algorithm



- **Initialization**: Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- **Selection**: $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- **Mutation**: $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

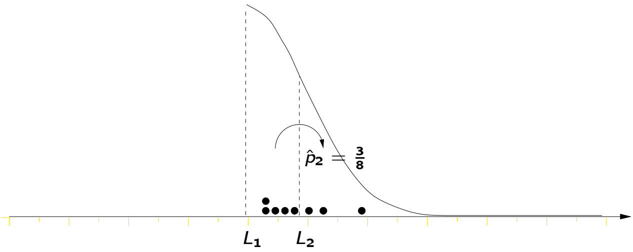
A Selection/Mutation Algorithm



- **Initialization**: Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- **Selection**: $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- **Mutation**: $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

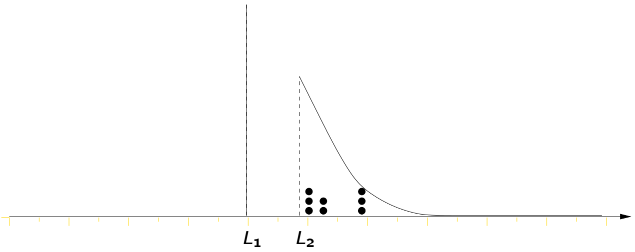
A Selection/Mutation Algorithm



- **Initialization**: Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- **Selection**: $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- **Mutation**: $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

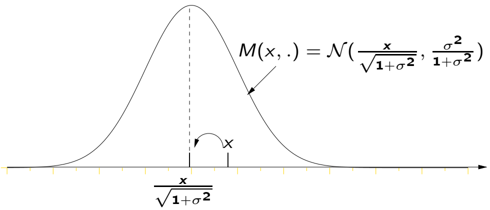
A Selection/Mutation Algorithm



- **Initialization**: Simulate an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N \sim \mu$.
- **Selection**: $\hat{\xi}_0^i = \xi_0^i$ if $S(\xi_0^i) > L_1$, else pick at random among the N_1 selected particles.
- **Mutation**: $\tilde{\xi}_0^i \sim M(\hat{\xi}_0^i, dx')$ and

$$\forall i \in \{1, \ldots, N\} \qquad \xi_1^i = \begin{cases} \tilde{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) > L_1 \\ \hat{\xi}_1^i & \text{if } S(\tilde{\xi}_1^i) \leq L_1 \end{cases}$$

The Impact of the Kernel



- **The model**: $X' = \frac{X+\sigma W}{\sqrt{1+\sigma^2}} \sim \mathcal{N}(0,1)$.
- **Expected square distance**: $\mathbb{E}[(X'-X)^2] = 2\left(1 - \frac{1}{\sqrt{1+\sigma^2}}\right)$.

Trade-off between two drawbacks:

- σ too large: most proposed mutations are refused.
- σ too small: particles almost don't move.

Variance Optimization

- **Multilevel Estimator**: $\hat{\alpha}_N = \hat{p}_1 \hat{p}_2 \dots \hat{p}_{n_0}$.
- **Fluctuations**: If the $\hat{p}_i$'s are independent, then

$$\sqrt{N} \cdot \frac{\hat{\alpha}_N - \alpha}{\alpha} \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \sum_{j=1}^{n_0} \frac{1-p_j}{p_j}\right).$$

- **Constrained Minimization**:

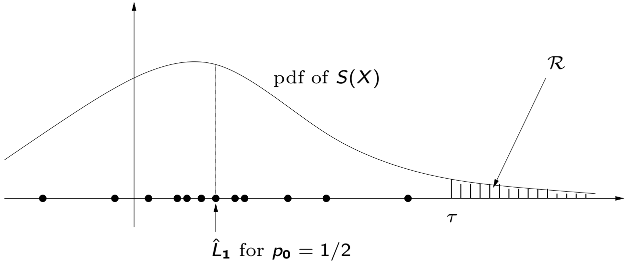
$$\arg\min_{p_1,\dots,p_{n_0}} \sum_{j=1}^{n_0} \frac{1-p_j}{p_j} \qquad \text{s.t.} \qquad \prod_{j=1}^{n_0} p_j = \alpha.$$

- **Optimum**: $p_1 = \dots = p_{n_0} = \alpha^{1/n_0}$.

$\Rightarrow$ **Solution**: Adaptive levels.

Adaptive Levels

Parameter: fix a proportion p_0 of surviving particles from one step to another rather than n_0 and the levels $L_1, \ldots, L_{n_0}$.



$\Rightarrow$ **Adaptive multilevel estimator**:

$$\alpha = r \times p_0^{n_0} \approx \hat{\alpha}_N = \hat{r} \times p_0^{\hat{n}_0},$$

with $n_0 = \left\lfloor \frac{\log \mathbb{P}(S(X) > \tau)}{\log p_0} \right\rfloor$ and $p_0 < r \leq 1$.

Adaptive Levels

Best adaptive asymtotic variance:

$$n_0 \frac{1 - p_0}{p_0} + \frac{1 - r}{r}.$$

Expression decreasing with larger values of p_0 (constraint $\alpha = r \times p_0^{n_0}$).

Limit case: $p_0 = 1 - \frac{1}{N}$, normalized asymptotic variance $\simeq \frac{-\log \alpha}{N}$.

Question: How to use the whole empirical c.d.f. of the scores ?

Adaptive Tilting

Idea: exponential tilting to improve the likelihood of large values of $S(X)$, using only the current empirical distribution.

Let F_Y the c.d.f. of $Y = S(X)$. The rare event is

$$\mathbb{R} = \{Y > L\} = \{-\log F_Y(Y) < -\log F_Y(L)\}.$$

$E = -\log F_Y(Y)$ is exp(1) distributed (F_Y continuous).
We want $X_k^1, \dots, X_k^N$ (approx.) distributed such that $-\log F_Y(\phi(X_k^1)), \dots, -\log F_Y(\phi(X_k^N))$ are a sample of $\exp(a_k)$.
With the knowledge of F_Y, this could be done by a MCMC approach.
μ_k the target distribution, absolutely continuous w.r.t. μ:

$$\frac{d\mu_k}{d\mu}(x) = a_k \exp(-(a_k - 1)(-\log F_Y(\phi(x)))) = a_k (F_Y(S(x)))^{a_k - 1}.$$

Adaptive Tilting

At iteration k, we have a sample $y_1 = S(x_1), \dots, y_N = S(x_N)$ of a random variable Y_k such that $-\log(F_Y(Y_k))$ is distributed as $\exp(a_k)$. Let us denote F_k its c.d.f.

$$F_k(h) = \mathbb{P}(Y_k \leq h) = (F_Y(h))^{a_k}.$$

Estimate of F_Y *on the region of interest* by

$$F_Y(h) \simeq (F_k^N(h))^{\frac{1}{a_k}},$$

where F_k^N is the empirical c.d.f. of the current sample.

Level Splitting $\longleftrightarrow$ New empirical c.d.f.

New Algorithm

- Start with a μ sample $x_1^1, \dots, x_N^1$.
- Compute $y_1^1 = S(x_1^1), \dots, y_N^1 = S(x_N^1)$.
- Sort $y_1^1, \dots, y_N^1$. This gives the empirical c.d.f F_1^N.
- $k = 1$, $a_1 = 1$.
- Iterate
  - $k = k + 1$.
  - $a_k = a_{k-1}/p_0$.
  - Metropolis-Hastings MCMC in parallel with target distribution μ_k (for the whole sample $x_1^{k-1}, \dots, x_N^{k-1}$, see next algorithm). This will make use of F_{k-1}^N.
  - This will give the next samples $x_1^k, \dots, x_N^k$ and $y_1^k = S(x_1^k), \dots, y_N^k = S(x_N^k)$.
  - Sort $y_1^k, \dots, y_N^k$. This gives the empirical c.d.f F_k^N.
- Until $\hat{q} = \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{y_j^k > L} > p_0$.
- Return $\hat{p} = 1 - (1 - \hat{q})^{\frac{1}{a_k}}$.

Metropolis-Hastings iteration

- Draw a proposal $\tilde{X}_{m+1} \sim K(X_m, .)$.
- Compute Metropolis-Hastings ratio

$$
\begin{aligned}
r &= \frac{K(\tilde{X}_{m+1}, X_m)\mu(\tilde{X}_{m+1})\frac{d\mu_k}{d\mu}(\tilde{X}_{m+1})}{K(X_m, \tilde{X}_{m+1})\mu(X_m)\frac{d\mu_k}{d\mu}(X_m)} \\
&= \frac{K(\tilde{X}_{m+1}, X_m)\mu(\tilde{X}_{m+1})}{K(X_m, \tilde{X}_{m+1})\mu(X_m)} \left[\frac{F_{k-1}^N(\tilde{X}_{m+1})}{F_{k-1}^N(X_m)}\right]^{\frac{a_k}{a_{k-1}} - \frac{1}{a_{k-1}}} .
\end{aligned}
$$

- Draw $U \sim U(0,1)$.
- If $U < r$, then $X_{m+1} = \tilde{X}_{m+1}$ else $X_{m+1} = X_m$.
- $m = m + 1$.

Numerical Results

Toy example: $\mathbb{P}(N(0,1) > 5) = 2.8665 \times 10^{-07}$

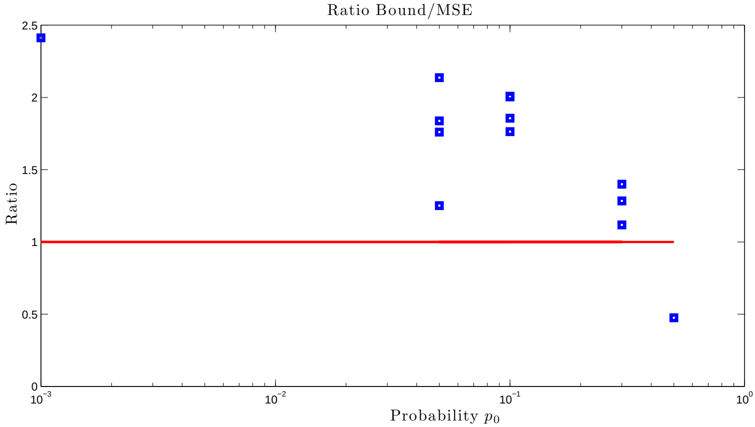


Figure: Ratio of splitting variance bound over empirical MSE for different values of p_0.

Conclusion

- It works ! (quite unexpected...)
- Performs better than splitting for $p_0 \leq 1/3$, but worse than "last particle" algorithm.
- Fully parallel algorithm.
- Possible improvements: population MCMC, estimate of the rare probability.
- To do: theoretical analysis...