

Rare event simulation for a static distribution

F. Cérou — P. Del Moral — T. Furon — A. Guyader

N° 6792

Janvier 2009

Thèmes COG et NUM

 ***Rapport
de recherche***

Rare event simulation for a static distribution

F. Cérou* , P. Del Moral† , T. Furon‡ , A. Guyader§

Thèmes COG et NUM — Systèmes cognitifs et Systèmes numériques
Équipes-Projets ASPI, CQFD et TEMICS

Rapport de recherche n° 6792 — Janvier 2009 — 18 pages

Abstract: This paper discusses the rare event simulation for a fixed probability law. The motivation comes from problems occurring in watermarking and fingerprinting of digital contents, which is a new application of rare event simulation techniques. We provide two versions of our algorithm, and discuss the convergence properties and implementation issues. A discussion on recent related works is also provided. Finally, we give some numerical results in watermarking context.

Key-words: Rare events, adaptive multilevel simulation, asymptotic normality, probability of false alarm.

The author names appear in alphabetical order.

This work was partially supported by the French Agence Nationale de la Recherche (ANR), project Nebbiano, number ANR-06-SETI-009.

* INRIA Rennes - Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France

† INRIA Bordeaux Sud-Ouest & Institut de Mathématiques de Bordeaux, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France

‡ INRIA Rennes - Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France

§ Equipe de Statistique IRMAR, Université de Haute Bretagne, Place du Recteur H. Le Moal, CS 24307, 35043 Rennes Cedex, France

Simulation d'événements rares pour une loi de probabilité fixée

Résumé : Cet article traite de la simulation d'événements rares pour une loi de probabilité donnée. Il trouve son origine dans des problèmes d'estimation de fausse alarme en watermarking et fingerprinting, domaines d'applications nouveaux pour les techniques d'événements rares. Nous proposons deux versions de notre algorithme, donnons une étude théorique des propriétés de convergence et détaillons les questions d'implémentation. Ceci est finalement illustré par des simulations dans le domaine du watermarking.

Mots-clés : Événements rares, simulation multi-niveaux adaptative, normalité asymptotique, probabilité de fausse alarme.

1 Introduction and motivation

The goal of this work is to deal with rare events for a fixed probability law. Unlike many other works concerning rare event estimation and simulation, we are simply concerned here with events of the type $\{X \in A\}$ for some random element X , with $\mathbb{P}(X \in A) \ll 1$, and with no dynamical model for X (i.e. X is not a process indexed by the time). In order to use the framework developed for Markov processes (see [5, 10]), we construct a family of Markov transition kernels whose invariant measures are the law of X restricted on smaller and smaller sets, the smallest being A . As usual when using a splitting technique in rare event simulation, we decompose the rare event in not so rare nested events, with the product of probabilities being the probability of the rare event.

Our motivation for this framework comes from problems occurring in watermarking of digital contents. Here the term watermarking refers to a set of techniques for embedding/hiding information in a digital file (typically audio or video), such that the change is not noticed, and very hard to remove. See [17] for details.

In order to be used in an application, a watermarking technique must be reliable. Here are two application scenarii where a wrong estimation of the probability of error could lead to a disaster.

Copy protection. Assume commercial contents are encrypted and watermarked and that future consumer electronics storage devices have a watermark detector. These devices refuse to record a watermarked content. The probability of false alarm is the probability that the detector considers an original piece of content (which has not been watermarked) as protected. The movie that a user shot during his holidays could be rejected by this storage device. This absolutely non user-friendly behavior really scares consumer electronics manufacturers. In the past, the Copy Protection Working Group of the DVD forum evaluated that at most one false alarm should happen in 400 hours of video [17]. As the detection rate was one decision per ten seconds, this implies a probability of false alarm in the order of 10^{-5} . An accurate experimental assessment of such a low probability of false alarm would demand to feed a real-time watermarking detector with non-watermarked content during 40,000 hours, *i.e.* more than 4 years! Proposals in response of the CPTWG's call were, at that time, never able to guarantee this level of reliability.

Fingerprinting. In this application, users' identifiers are embedded in purchased content. When content is found in an illegal place (*e.g.* a P2P network), the right holders decode the hidden message, find a serial number, and thus they can trace the traitor, *i.e.* the customer who has illegally broadcast their copy. However, the task is not that simple because dishonest users might collude. For security reason, anti-collusion codes have to be employed. Yet, these solutions (also called weak traceability codes [2]) have a non-zero probability of error (defined as the probability of accusing an innocent). This probability should be, of course, extremely low, but it is also a very sensitive parameter: anti-collusion codes get longer (in terms of the number of bits to be hidden in content) as the probability of error decreases. Fingerprint designers have to strike a trade-off, which is hard to conceive when only rough estimation of the

probability of error is known. The major issue for fingerprinting algorithms is the fact that embedding large sequences implies also assessing reliability on a huge amount of data which may be practically unachievable without using rare event analysis.

2 Assumptions and ingredients

We assume that X is a random element on \mathbb{R}^d for some $d > 0$, and denote by μ its probability law on the underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We denote by A the rare set of interest, and we assume that $A = \{x \in \mathbb{R}^d \text{ s.t. } \Phi(x) \geq L\}$ for some continuous function $\Phi : \mathbb{R}^d \mapsto \mathbb{R}$ and some real number L . We also assume that we know how to draw i.i.d. samples from μ .

Now to construct the algorithm, we will need to choose the following ingredients. First we need to choose an increasing sequence in $\overline{\mathbb{R}} \{L_0, \dots, L_n\}$, with $L_0 = -\infty$ and $L_n = L$. If we denote $A_j = \{x \in \mathbb{R}^d, \Phi(x) \geq L_j\}$, we get a family of nested sets $\mathbb{R}^d = A_0 \supset A_1 \supset \dots \supset A_n = A$ such that $\mathbb{P}(X \in A_k | X \in A_{k-1})$ is not too small. For indices $m > n$, we assume that $L_m = L_n$ and $A_m = A_n$. We also need to choose a Markov transition kernel K on \mathbb{R}^d which is μ -symmetric, that is

$$\forall (x, y) \in \mathbb{R}^{2d}, \mu(dx)K(x, dy) = \mu(dy)K(y, dx).$$

As a consequence, K has μ as an invariant measure.

As we will see in the sequel, the choice of the L_j 's can be made adaptive and is thus not an issue. But the choice of the kernel K is crucial. Even if any μ -symmetric kernel would eventually do the job, we need to carefully choose it to make the algorithm efficient. We will discuss this point later on.

Now we can consider the following Markov chain: $X_0 \sim \mu$ and the inhomogeneous transitions given by $P(X_k \in dy | X_{k-1} = x) = M_k^K(x, dy)$, with

$$M_k^K(x, dy) = K(x, dy) \mathbb{1}_{A_k}(y) + K(x, A_k^c) \delta_x(dy).$$

For $k \in \{0, \dots, n\}$, let us denote $\mu_k(dx) = \frac{1}{\mu(A_k)} \mathbb{1}_{A_k}(x) \mu(dx)$ the normalized restriction of μ on A_k .

Proposition 1. *The measure μ_k is invariant by the transition kernel M_k^K .*

Proof We have

$$\begin{aligned} & \int_x \mu_k(dx) M_k^K(x, dy) \\ &= \int_x \mu_k(dx) (K(x, dy) \mathbb{1}_{A_k}(y) + K(x, A_k^c) \delta_x(dy)) \\ &= \int_x \int_z \mu_k(dx) K(x, dz) (\mathbb{1}_{A_k}(z) \delta_z(dy) + \mathbb{1}_{A_k^c}(z) \delta_x(dy)) \end{aligned}$$

so that

$$\begin{aligned}
& \int_x \mu_k(dx) M_k^K(x, dy) \\
&= \int_x \frac{1}{\mu(A_k)} \mathbb{1}_{A_k}(x) \mu(dx) K(x, dy) \mathbb{1}_{A_k}(y) \\
&\quad + \int_z \frac{1}{\mu(A_k)} \mathbb{1}_{A_k}(y) \mu(dy) K(y, dz) \mathbb{1}_{A_k^c}(z) \\
&= \int_x \frac{1}{\mu(A_k)} \mathbb{1}_{A_k}(x) \mu(dx) K(x, dy) \mathbb{1}_{A_k}(y) \\
&\quad + \int_z \frac{1}{\mu(A_k)} \mathbb{1}_{A_k}(y) \mu(dz) K(z, dy) \mathbb{1}_{A_k^c}(z) \\
&= \mu_k(dy).
\end{aligned}$$

■

Proposition 2. For every test function φ , for $k \in \{0, \dots, n\}$, we have the following Feynman-Kac representation

$$\mu_{k+1}(\varphi) = \frac{\mathbb{E}[\varphi(X_k) \prod_{j=0}^k \mathbb{1}_{A_{j+1}}(X_j)]}{\mathbb{E}[\prod_{j=0}^k \mathbb{1}_{A_{j+1}}(X_j)]}.$$

Proof We use induction to show that

$$\mathbb{E}[\varphi(X_k) \prod_{j=0}^k \mathbb{1}_{A_{j+1}}(X_j)] = \mu(A_{k+1}) \mu_{k+1}(\varphi).$$

The case $k = 0$ is obvious. Then assume the property true for $k - 1$. We write, using the Markov property and proposition 1,

$$\begin{aligned}
& \mathbb{E}[\varphi(X_k) \prod_{j=0}^k \mathbb{1}_{A_{j+1}}(X_j)] \\
&= \mathbb{E}[\mathbb{E}[\varphi(X_k) \mathbb{1}_{A_{k+1}}(X_k) | X_0, \dots, X_{k-1}] \prod_{j=0}^{k-1} \mathbb{1}_{A_{j+1}}(X_j)] \\
&= \mathbb{E}[M_k^K(\varphi \mathbb{1}_{A_{k+1}})(X_{k-1}) \prod_{j=0}^{k-1} \mathbb{1}_{A_{j+1}}(X_j)] \\
&= \mu(A_k) \mu_k(M_k^K(\varphi \mathbb{1}_{A_{k+1}})) \\
&= \mu(A_k) \mu_k(\varphi \mathbb{1}_{A_{k+1}}).
\end{aligned}$$

Now it is obvious that

$$\mu_k(\varphi \mathbb{1}_{A_{k+1}}) = \mu_{k+1}(\varphi) \frac{\mu(A_{k+1})}{\mu(A_k)}.$$

Then taking the case $\varphi = \mathbb{1}$ we have

$$\mathbb{E}\left[\prod_{j=0}^k \mathbb{1}_{A_{j+1}}(X_j)\right] = \mu(A_{k+1}),$$

which concludes the proof. ■

3 The algorithm

3.1 Fixed levels

From proposition 2 we see that we are in the framework of Feynman-Kac formulae, and thus we can construct an approximation of the associated measures using an interacting particle method as the one studied in [8]. Basically, at each iteration k , it consists in propagating the particles according to the transitions given by M_k^K , and then in selecting the particles according to the potentials, here $\mathbb{1}_{A_{k+1}}$ (*i.e.* a zero-one valued function in this case).

Also note that moving a particle according to M_k^K is twofold: first we propose a new transition according to K , and accept the transition only if it stays in A_k , keeping the old position otherwise.

Concerning the approximation of the rare event probability, we just consider the following obvious property

$$\begin{aligned} \mathbb{P}(X \in A_n) &= \prod_{k=0}^{n-1} \mathbb{P}(X \in A_{k+1} | X \in A_k) \\ &= \mathbb{E}\left[\prod_{k=0}^{n-1} \mathbb{1}_{A_{k+1}}(X_k)\right] \\ &= \prod_{k=0}^{n-1} \frac{\mathbb{E}[\mathbb{1}_{A_{k+1}}(X_k) \prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)]}{\mathbb{E}[\prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)]}. \end{aligned}$$

We see here that we can approximate at each stage $\mathbb{P}(X \in A_{k+1} | X \in A_k)$ by the proportion of the particles already in the next set, and the total probability is estimated as the product of those.

This gives the algorithm 1, which can also be considered in the rare event simulation framework as a kind of importance splitting method, introduced for example by [13] in the context of particle physics, or by [21] in the telecommunication area.

Algorithm 1

Parameters

N the number of particles, the sequence $\{L_0, \dots, L_n\}$ of levels.

Initialization

Draw an i.i.d. N -sample $\xi_0^j, j = 1, \dots, N$ of the law μ .

Iterations

for $k = 0$ to $n - 1$ /* level number */

Let $I_k = \{j / \xi_k^j \in A_{k+1}\}$.

for $j \in I_k$, let $\tilde{\xi}_{k+1}^j = \xi_k^j$, and for $j \notin I_k$, let $\tilde{\xi}_{k+1}^j$ be a copy of ξ_k^ℓ where ℓ is chosen randomly in I_k with uniform probabilities.

Let $p_k = \frac{|I_k|}{N}$.

From each sample $\tilde{\xi}_{k+1}^j, j = 1, \dots, N$, draw a new sample $\hat{\xi}_{k+1}^j \sim K(\tilde{\xi}_{k+1}^j, \cdot)$.

If $\hat{\xi}_{k+1}^j \in A_{k+1}$ then let $\xi_{k+1}^j = \hat{\xi}_{k+1}^j$, and $\xi_{k+1}^j = \tilde{\xi}_{k+1}^j$ otherwise.

endfor

Output

Estimate the probability of the rare event by $\hat{P}_A = \prod_{k=0}^{n-1} p_k$.

The last set of particles is a (non independent) identically distributed sample of the law of the rare event μ_n .

The asymptotic behavior as the number of particles $N \rightarrow \infty$ of the interacting particle model we have constructed has been extensively studied in [8]. For example Proposition 9.4.1 and Remark 9.4.1 give that

$$\sqrt{N}(\hat{P}_A - P(X \in A)) \xrightarrow[N \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, \sigma^2)$$

Unfortunately, the asymptotic variance σ^2 is often not explicit, and depends on the kernel K in a complicated way. All we can say is that it would be minimal if we could (which is obviously unrealistic) draw an i.i.d. sample of the law of X conditionally on the event $\{\Phi(X) \geq L_k\}$ at each step k , or equivalently apply an infinite number of time the kernel M_k^K at each step k (see section 4.2). In this case, from the discussion in [5], we would have for the asymptotic variance

$$\sigma^2 = \sum_{k=0}^{n-1} \frac{1 - p_k}{p_k},$$

with $p_k = \mathbb{P}(\Phi(X) \geq L_k | \Phi(X) \geq L_{k-1})$. This in turn would be minimal¹, for a fixed value of P_A and a fixed number of levels, if $p_k \equiv p_0$ for all k . In this case, the asymptotic variance is simply $n \frac{1-p_0}{p_0}$, with $p_0 = (P_A)^{\frac{1}{n}}$.

If one is not interested in a convergence in distribution like the CLT mentioned above, but only in the variance of our estimate of the rare event probability, then

¹This is a simple constrained optimization problem: $\min \sum_{k=0}^{n-1} \frac{1-p_k}{p_k}$ with the constraint $\prod_{k=0}^{n-1} p_k = P_A$.

we can use the recent non asymptotic results obtained by some of the authors in [4]. Under some regularity conditions (mainly mixing property of the kernel M_k^K), using corollary 5.2 of [4], we have that: there exist positive constants α_k , for $0 \leq k \leq n-1$, such that for $N \geq N_0 = \sum_{k=0}^{n-1} \frac{\alpha_k}{p_k}$,

$$\mathbb{E} \left(\left[\frac{\hat{P}_A - P_A}{P_A} \right]^2 \right) \leq 4 \frac{N_0}{N},$$

with p_k defined as above. If we assume again the very optimistic (and also very unrealistic) case where we have an i.i.d. sample, then all the α_j 's are equal to 1, and $N_0 = \sum_{k=0}^{n-1} \frac{1}{p_k}$.

It is also worth mentioning that as discussed in [4], this non asymptotic results can be used to get efficiency result when the probability of the rare event goes to 0, at least for some favorable settings.

3.2 Adaptive version of the algorithm

As we may not have a great insight about the law μ , the choice of the levels L_1, \dots, L_n might prove to be quite tricky. For example, we want to avoid the particle system to completely die by choosing two consecutive levels too far apart. As proposed in [7], the level sets can be chosen adaptively, ensuring that the particle system will not die, and that the levels are distributed in a way to minimize the asymptotic variance of the estimate of the rare event probability. The method is very simple. We choose a prescribed success rate p_0 between two consecutive levels. In practice, $0.75 \leq p_0 \leq 0.8$ works very well. After each application of the kernel M_k^K , we sort the particles ξ_{k+1}^j according to their scores $\Phi(\xi_{k+1}^j)$. Then we choose as the next level the empirical quantile $\hat{L}_{k+1} = \Phi(\xi_{k+1}^{j_0})$ such that a proportion p_0 of the particles are above it. From this level \hat{L}_{k+1} , one defines A_{k+1} , and the rest of the algorithm is unchanged. We end up with a variant where the levels are evenly spaced in terms of probability of success, which, as mentioned in [14] and [5], gives a minimal asymptotic variance.

The algorithm then stops when some $\hat{L}_{n_0+1} \geq L$, and the probability is estimated by $\hat{P}_A = p_0^{n_0} r$, with r being the actual number of particles in the last iteration being above level L . Note that the number n_0 of steps is in theory random, but with N reasonably large, it is fixed by the ratio of the logarithms

$$\left\lfloor \frac{\log \mathbb{P}(X \in A)}{\log p_0} \right\rfloor \quad (1)$$

with a probability very close to 1.

The adaptive algorithm is given in Algorithm 2 below.

Algorithm 2

Parameters

N the number of particles, the number $N_0 < N$ of succeeding particles, and let $p_0 = N_0/N$.

Initialization

Draw an i.i.d. N -sample $\xi_0^j, j = 1, \dots, N$ of the law μ .

Compute \hat{L}_1 the $1 - p_0$ quantile of $\Phi(\xi_0^j), j = 1, \dots, N$

$k = 0$;

Iterations

while $\hat{L}_{k+1} < L$ do

Let $I_k = \{j / \Phi(\xi_k^j) \geq \hat{L}_{k+1}\}$.

for $j \in I_k$, let $\tilde{\xi}_{k+1}^j = \xi_k^j$, and for $j \notin I_k$, let $\tilde{\xi}_{k+1}^j$ be a copy of ξ_k^ℓ where ℓ is chosen randomly in I_k with uniform probabilities.

From each sample $\tilde{\xi}_{k+1}^j, j = 1, \dots, N$, draw a new sample $\hat{\xi}_{k+1}^j \sim K(\tilde{\xi}_{k+1}^j, \cdot)$.

If $\Phi(\hat{\xi}_{k+1}^j) \geq \hat{L}_{k+1}$ then let $\xi_{k+1}^j = \hat{\xi}_{k+1}^j$, and $\xi_{k+1}^j = \tilde{\xi}_{k+1}^j$ otherwise.

Compute \hat{L}_{k+2} the $1 - p_0$ quantile of $\Phi(\xi_{k+1}^j), j = 1, \dots, N$.

$k = k + 1$;

endwhile

Let N_L the number of particles $\xi_k^j, j = 1, \dots, N$, such that $\Phi(\xi_k^j) \geq L$.

Output

Estimate the probability of the rare event by $\hat{P}_A = \frac{N_L}{N} p_0^k$.

The last set of particles is a (non independent) identically distributed sample of the law of the rare event μ_n .

The cost of adaptive levels in term of complexity is just a $\log N$ factor (from the quick sort), and in terms of accuracy it introduces a bias which is asymptotically zero, as shown in [7]. But the bias was not much discussed in the latter, so we will give an estimate of it below.

3.3 On the bias for the adaptive algorithm

To be able to use results in [7], we will assume, that at each stage, the new sample is i.i.d., or equivalently that we apply the kernel an infinite number of times (see section 4.2 below). This is not realistic, but will provide a useful insight on how much we should worry about the bias.

Let us write the rare event probability as $p = P_A = r p_0^{n_0}$, with $n_0 = \lfloor \frac{\log p}{\log p_0} \rfloor$ and $r = p p_0^{-n_0}$. In the same way we write $\hat{p} = \hat{P}_A = \hat{r} p_0^{\hat{n}_0}$, with \hat{n}_0 the number of steps before the algorithm stops. As shown in [7], the actual number \hat{n}_0 of steps is equal to n_0 with any large probability provided N is large enough, so we assume all the expectations will be taken on the subset of Ω on which $\hat{n}_0 = n_0$.

We denote $Z = \Phi(X)$, F the cumulative distribution function of Z , and L_{n_0} the exact level such that $\mathbb{P}(Z \geq L_{n_0}) = p_0^{n_0}$. Thus the relative bias is

$$\begin{aligned} \frac{\mathbb{E}[\hat{p}] - p}{p} &= \frac{\mathbb{E}[\hat{r}] - r}{r} \\ &= \frac{\mathbb{E}[\mathbb{P}(Z \geq L | Z \geq \hat{L}_{n_0})] - \mathbb{P}(Z \geq L | Z \geq L_{n_0})}{\mathbb{P}(Z \geq L | Z \geq L_{n_0})} \\ &= \mathbb{E} \left[\frac{\mathbb{P}(Z \geq L_{n_0}) - \mathbb{P}(Z \geq \hat{L}_{n_0})}{\mathbb{P}(Z \geq \hat{L}_{n_0})} \right] \\ &= \mathbb{E} \left[\frac{F(\hat{L}_{n_0}) - F(L_{n_0})}{1 - F(\hat{L}_{n_0})} \right] \\ &= \mathbb{E} \left[\frac{W}{a - W} \right] \end{aligned}$$

with $W = F(\hat{L}_{n_0}) - F(L_{n_0}) = F(\hat{L}_{n_0}) - (1 - p_0^{n_0})$, and $a = 1 - F(L_{n_0}) = p_0^{n_0} \approx 0$. Now considering that (see *e.g.* [1])

$$\frac{W}{a} = \frac{F(\hat{L}_{n_0}) - (1 - p_0^{n_0})}{p_0^{n_0}} \xrightarrow[N \rightarrow +\infty]{a.s.} 0,$$

we rewrite

$$\frac{\mathbb{E}[\hat{p}] - p}{p} = \frac{1}{a} \mathbb{E} \left[W \frac{1}{1 - \frac{W}{a}} \right],$$

and make an asymptotic expansion near 0,

$$\begin{aligned} \frac{\mathbb{E}[\hat{p}] - p}{p} &= \frac{1}{a} \mathbb{E} \left[W \left(1 + \frac{W}{a} + o_{\mathbb{P}} \left(\frac{W}{a} \right) \right) \right] \\ &= \frac{1}{a} \mathbb{E}[W] + \frac{1}{a^2} \mathbb{E}[W^2] + \frac{1}{a^2} o(\mathbb{E}[W^2]). \end{aligned}$$

Then we will use the following lemma.

Lemma 1.

$$\mathbb{E}[W] = \mathbb{E}[F(\hat{L}_{n_0}) - (1 - p_0^{n_0})] = 0$$

and

$$\mathbb{E}[W^2] = \text{var}(F(\hat{L}_{n_0}) - F(L_{n_0})) = \frac{n_0}{N} p^{2n_0-1} (1-p) + o\left(\frac{1}{N}\right).$$

The proof is left to the appendix. From the lemma, we see that the bias is positive and of order $\frac{1}{N}$:

$$\frac{\mathbb{E}[\hat{p}] - p}{p} \sim \frac{1}{N} \frac{n_0(1-p)}{p}.$$

Using the result in [7] on the asymptotic variance, we can write the following expansion:

$$\hat{p} = p \left(1 + \frac{1}{\sqrt{N}} \sqrt{n_0 \frac{1-p}{p} + \frac{1-r}{r}} X + \frac{1}{N} n_0 \frac{1-p}{p} + o_{\mathbb{P}} \left(\frac{1}{N} \right) \right),$$

where X is a standard Gaussian variable. Of course if one does not want any bias, then the solution is to make a first run to compute the levels, and a second run to actually compute the probability, but we can see from the above formula that, for the same computational cost, it is better in term of relative error to use directly our algorithm 2 and compute the levels on the fly.

Another remark worth mentioning is that the bias is always positive, giving a slightly over valued estimate. When we deal with catastrophic events, which is usually the case in rare event analysis, and provide an estimate, it is not a bad thing that the real value be a bit lower. This result will be later shown to prove efficient by a numerical example.

4 Tuning the algorithm

4.1 Choice of the kernel K

The choice of the transition kernel K is of course critical, and in practice will depend on the application, so that we cannot give a completely general way of finding it. But in the case of a Gibbs measure given by a bounded potential, we can use the Metropolis algorithm, as first proposed by [15], or a variant later proposed by Hastings [12].

4.2 Less dependent sample

Unlike all importance sampling based methods, our algorithm gives a sample distributed according to the real law of the rare event μ_n , but not independent. This may lead to poor variance behavior in some cases.

The samples are not independent because of the splitting of successful particles. But the effect of M_k^K is towards more independence among particles. Thus we can think of iterating the kernel a fixed number of times, or until a fixed number of particles, say 90 or 95%, have actually moved from their first position (i.e. at least one proposed K transition has been accepted).

If we consider all the particles together, each application of the kernel can be seen as applying a kernel $(M_k^K)^{\otimes N}$. It is obvious from proposition 1 that $(\mu_k)^{\otimes N}$ (the joint law of an i.i.d. sample) is an invariant measure for $(M_k^K)^{\otimes N}$. Then from [16], Proposition 13.3.2, we get that the total variation norm between the law of the sample as we iterate the kernel, and $(\mu_k)^{\otimes N}$, is non increasing. So even if it is not very helpful, these iterations at least do not harm.

When the chosen kernel K is of Metropolis-Hastings type, so is M_k^K (with a potential that can be infinite). Then, using [20], we can say a bit more, provided (which is generally the case) that the kernel used for the proposed transitions is aperiodic and irreducible. By Corollary 2 in [20], the Metropolis is also Harris recurrent, and then by Theorem 13.3.3 in [16], we have for any initial distribution λ

$$\| \int \lambda(dx) (M_k^K)^m(x, \cdot) - \mu_k \| \rightarrow 0 \text{ when } m \rightarrow +\infty,$$

where the norm is in total variation. Then we have for any initial cloud of particles $\Xi = (\xi^1, \dots, \xi^N)$, and any test function (ϕ^1, \dots, ϕ^N) ,

$$\begin{aligned} & |\delta_{\Xi}((M_k^K)^{\otimes N})^m(\bigotimes_{j=1}^N \phi_j) - \prod_{j=1}^N \mu_k(\phi_j)| \\ &= \left| \prod_{j=1}^N (M_k^K)^m(\phi_j)(\xi_j) - \prod_{j=1}^N \mu_k(\phi_j) \right| \rightarrow 0 \text{ when } m \rightarrow +\infty. \end{aligned}$$

By a standard density argument, we get that for all test functions ϕ on $(\mathbb{R}^d)^N$,

$$|\delta_{\Xi}((M_k^K)^{\otimes N})^m(\phi) - \mu_k^{\otimes N}(\phi)| \rightarrow 0 \text{ when } m \rightarrow +\infty.$$

This means that the more iterations we do with the kernel, the closer we get to an independent sample.

4.3 Mixing property of the kernel K

We have written the algorithms using a unique kernel K for all the iterations. Usually it is quite easy to construct not only one, but a family of kernel that are all μ -symmetric, and with different mixing properties. This fact can be useful when we see that, when applying M_k^K to the current particles, most of the transitions are refused (they are below the current threshold). In this case, we propose to change the kernel K for another one which is less mixing, that is in some way with "smaller steps", thus with a lower probability to go beyond the current level L_k . On the other hand, when almost all the transitions are accepted, it means that the kernel is poorly mixing, and that we could decrease the variance by choosing a kernel K that is more mixing, *i.e.* with "larger steps". For example, in section 6.1 below, this is tuned by changing the parameter α .

5 Discussion about related works

We will not discuss here the large amount of literature on importance sampling approach, but only focus on interacting particle approaches for the analysis of static rare events. As far as we know, the first work where algorithm 1 was outlined to deal with static rare events is [9]. This article is written in a much larger framework, and thus does not deal with the practical details of our precise setting.

More recently, and independently from the authors of the present work, a very similar algorithm has been proposed in [3]. These authors propose a very similar algorithm, including the use of quantiles of the cost/importance function Φ on the swarm of particles to estimate the next level. One main difference here is that they propose to first run the algorithm just to compute the levels, and then restart from the beginning with the proposed levels (two stage procedure), like in [11]. Actually we show here that computing the levels on the fly (within the same run as the one to compute the rare event probability, see algorithm 2 above, one stage procedure) we only pay a small bias on the estimate. In this regard the remark 3.2 page 489 of [3] might be misleading: if the levels are chosen from start, or using a preliminary run, then the resulting probability

estimate is unbiased, even if the level crossing probabilities are indeed dependent (see [5]).

Note that in [3] the general construction of the kernel M_k^K is not addressed, as the authors consider only examples where they can derive a Gibbs sampler for each μ_k . This is mainly possible because their function Φ is linear. Note that the authors stress that in the adaptive levels case, the resulting probability estimate is only asymptotically unbiased. They also mention that to increase the diversity, one can apply the MCMC move several times.

Another very similar approach was proposed in the context of combinatorial counting in [18]. This work is focused on discrete but very large state space, and mainly on optimization and counting problems in which μ is the uniform probability. The main difference between the proposed algorithm and ours is that, instead of a resampling with replacement procedure, the author uses what he calls a cloning procedure, where the number of offsprings is fixed (*i.e.* the same for all the particles in the sample), but adaptive to keep the number roughly constant. The algorithm also uses the specificity of the uniform law by removing redundant particles after the MCMC step (actually several applications of the MCMC transition kernel on each particle). This step is called screening. Note also that all the intermediate positions of the MCMC steps are kept, or only a fraction of them if he needs to improve the diversity of the sample. It should be mentioned that the author describes a variant which combines an interacting (cloning) particle approach with an importance sampling step using a change of measure given by cross entropy. Finally, here again the author uses a Gibbs sampler whose invariant law is exactly μ_k .

We also would like to mention that these last two papers do not provide rigorous mathematical analysis of the convergence properties of the proposed algorithms.

6 Applications

6.1 Zero-bit watermarking

The zero-bit watermarking is a toy example where X is a Gaussian vector in \mathbb{R}^d , with zero mean and identity covariance matrix, $\Phi(X) = \frac{\langle X, u \rangle}{\|X\|}$ and u is a fixed normalized vector. Then the region $A = \{x \in \mathbb{R}^d \text{ s.t. } \Phi(x) \geq L\}$ is a cone as shown on figure 1. For a Gaussian distribution, the obvious choice for the kernel is the following. If we start from any point x , then the new position is distributed like

$$x' = \frac{x + \alpha W}{\sqrt{1 + \alpha^2}},$$

where W is a $\mathcal{N}(0, I_d)$ \mathbb{R}^d valued random vector and α a positive number.

Note that here we can compare our estimates of the rare event probability with the result of a numerical integration. For example, in our simulations $d = 20$ and $L = 0.95$, so that $\mathbb{P}(\Phi(X) \geq L) \approx 4.710^{-11}$. We have run our algorithm in this case with adaptive levels and iterations of the kernel until 90% of the particles have moved at each step.

For several numbers of particles, we have done the complete algorithm 200 times in order to estimate the bias and variance. Figure 2 shows the rate of convergence in N^{-1} for the relative bias, and figure 3 shows the convergence of

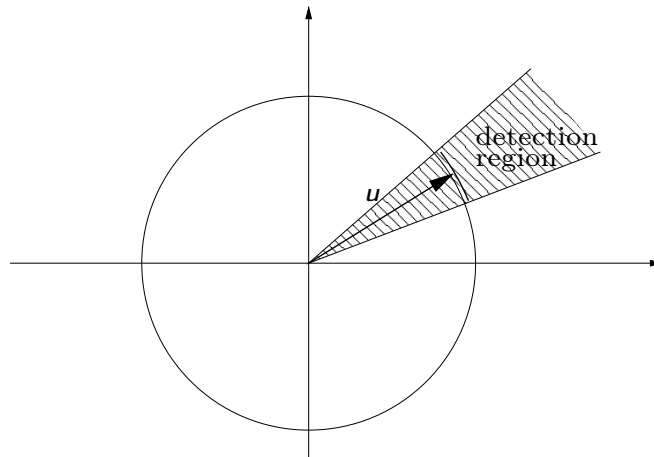


Figure 1: Detection region.

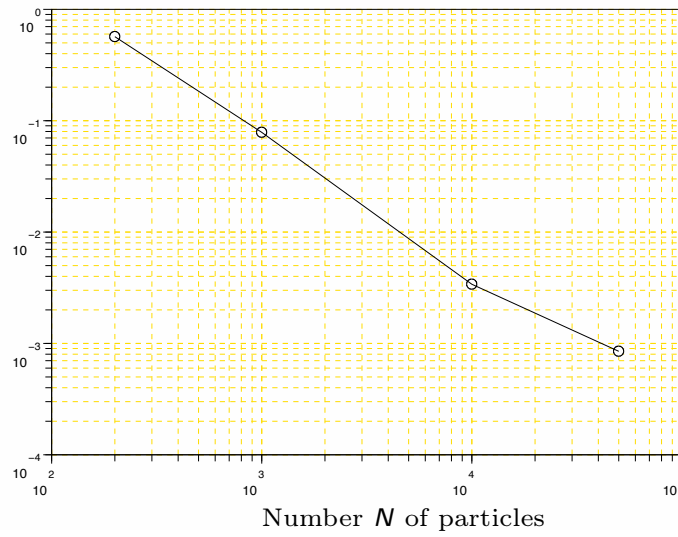


Figure 2: Relative bias.

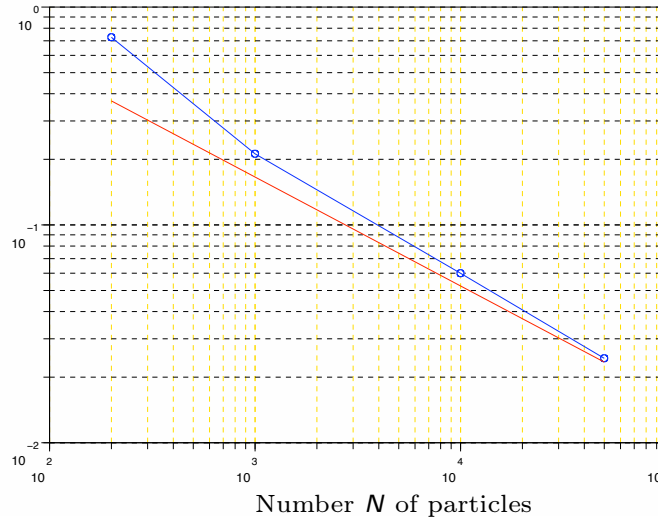


Figure 3: Normalized standard deviation.

the (normalized by the rare event probability) standard deviation to minimum achievable, which is that of i.i.d. samples at each stage, that is

$$\sqrt{N} \left(n_0 \frac{1-p_0}{p_0} + r_0 \right)$$

where n_0 is given by equation (1), and r_0 is such that $\mathbb{P}(X \in A) = p_0^{n_0} r_0$ (see [7]).

6.2 Tardos probabilistic codes

We are interested here in embedding an identifier in each copy of a purchased content. Then a copy, maybe the result of several manipulations, or even a collusion, is found on the web, and we want to decide whether or not it can be originated from a certain user. The rare event will be to accuse an innocent user to be guilty for this.

The embedded message consists of bits $X = (X_1, \dots, X_m)$, where each X_i is independent from the others, and drawn from a Bernoulli $B(p_i)$. The p_i 's are themselves random, drawn from a given distribution with density f on $[0, 1]$. Then we find a copy with fingerprint $y = (y_1, \dots, y_m) \in \{0, 1\}^m$. We conclude that a user is guilty if the score

$$S(X) = \sum_{i=1}^m y_i g_i(X_i)$$

is larger than some value L , for some given functions g_i 's. This approach was proposed by Tardos in [19], where he derives good choices for f and the g_i 's.

To apply our algorithm, we need to choose the kernel K . As the X_i 's are independent, we choose randomly r indices $\{j_1, \dots, j_r\} \in \{1, \dots, m\}$, with r being a fixed parameter. Then for each j_l , we draw a new X'_{j_l} independently from $B(p_i)$.

Extensive numerical results for this type of fingerprinting using our algorithm can be found in [6].

A Proof of lemma 1

First of all, some notation. Let U_j , $j = 1, \dots, N$ an i.i.d. family of random variables uniformly distributed on $(0, 1)$. We denote by $U_{(j)}$ the j th largest sample. We have then $0 \leq U_{(1)} \leq \dots \leq U_{(j-1)} \leq U_{(j)} \leq U_{(j+1)} \leq \dots \leq U_{(N)} \leq 1$. For simplicity, we will assume that $p_0 = \frac{k}{N}$ for some $1 \leq k \leq N$. Then it is well known that (see *e.g.* [1] formula (2.2.20) page 14)

$$\mathbb{E}[U_{(N-k)}] = 1 - p_0. \quad (2)$$

Expectation of W . We will show it is zero by induction on n_0 .

Case $n_0 = 1$. $F(\hat{L}_1)$ has the same law as $U_{(N-k)}$, thus the result is obvious by equation (2).

Induction. Assume $\mathbb{E}[F(\hat{L}_{n_0-1})] = 1 - p^{n_0-1}$. We use the notation:

$$\mathbb{F}(x, x') = \frac{F(x') - F(x)}{1 - F(x)}.$$

Clearly, we have, with the convention $\hat{L}_0 = -\infty$,

$$\prod_{k=1}^{n_0} (1 - \mathbb{F}(\hat{L}_{k-1}, \hat{L}_k)) = 1 - F(\hat{L}_{n_0}),$$

and thus

$$\begin{aligned} & \mathbb{E}[1 - F(\hat{L}_{n_0})] \\ &= \mathbb{E}\left[\prod_{k=1}^{n_0} (1 - \mathbb{F}(\hat{L}_{k-1}, \hat{L}_k))\right] \\ &= \mathbb{E}\left[\mathbb{E}[1 - \mathbb{F}(\hat{L}_{n_0-1}, \hat{L}_{n_0}) | \hat{L}_{n_0-1}] \prod_{k=1}^{n_0-1} (1 - \mathbb{F}(\hat{L}_{k-1}, \hat{L}_k))\right]. \end{aligned}$$

From lemma 5 in [7], we have for any test function φ ,

$$\mathbb{E}[\varphi(\mathbb{F}(\hat{L}_{n_0-1}, \hat{L}_{n_0})) | \hat{L}_{n_0-1}] = \mathbb{E}[\varphi(U_{(N-k)})],$$

from which we have

$$\mathbb{E}[1 - \mathbb{F}(\hat{L}_{n_0-1}, \hat{L}_{n_0}) | \hat{L}_{n_0-1}] = \mathbb{E}[1 - U_{(N-k)}] = p_0.$$

Thus

$$\mathbb{E}[1 - F(\hat{L}_{n_0})] = p_0 \mathbb{E}\left[\prod_{k=1}^{n_0-1} (1 - \mathbb{F}(\hat{L}_{k-1}, \hat{L}_k))\right] = p_0^{n_0},$$

using the induction property for $n_0 - 1$, which proves that W has zero mean.

Variance of W . From the proof of theorem 2 in [7], we have

$$\sqrt{N} \left(\prod_{k=1}^{n_0} (1 - \mathbb{F}(\hat{L}_{k-1}, \hat{L}_k)) - p_0^{n_0} \right) \xrightarrow[N \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, \sigma_{n_0}^2),$$

with $\sigma_{n_0}^2 = n_0 \frac{1-p_0}{p_0} p_0^{2n_0}$. So we have

$$\sqrt{N}(1 - F(\hat{L}_{n_0}) - p_0^{n_0}) \xrightarrow[N \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, \sigma_{n_0}^2),$$

and by symmetry,

$$\sqrt{N}(F(\hat{L}_{n_0}) - F(L_{n_0})) \xrightarrow[N \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, \sigma_{n_0}^2).$$

Which means that

$$\text{var}(F(\hat{L}_{n_0}) - F(L_{n_0})) = \frac{1}{N} \sigma_{n_0}^2 + o\left(\frac{1}{N}\right),$$

which concludes the proof.

References

- [1] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A first course in order statistics*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, 1992. A Wiley-Interscience Publication.
- [2] A. Barg, G. R. Blakley, and G. A. Kabatiansky. Digital fingerprinting codes: problem statements, constructions, identification of traitors. *IEEE Trans. on Signal Processing*, 51(4):960–980, April 2003.
- [3] Z. I. Botev and D. P. Kroese. An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. *Methodology and Computing in Applied Probability*, 10(4):471–505, 2008.
- [4] F. Cérou, P. Del Moral, and A. Guyader. A non asymptotic variance theorem for unnormalized Feynman-Kac particle models. Technical Report RR-6716, Inria, November 2008. Submitted.
- [5] F. Cérou, P. Del Moral, F. Le Gland, and P. Lezaud. Genetic genealogical models in rare event analysis. *Latin American Journal of Probability and Mathematical Statistics*, 1, 2006.
- [6] F. Cérou, T. Furon, and A. Guyader. Experimental assessment of the reliability for watermarking and fingerprinting schemes. *EURASIP Journal on Information Security*, 2008.
- [7] F. Cérou and A. Guyader. Adaptive multilevel splitting for rare event analysis. *Stoch. Anal. Appl.*, 25(2):417–443, 2007.
- [8] P. Del Moral. *Feynman-Kac formulae, Genealogical and interacting particle systems with applications*. Probability and its Applications. Springer-Verlag, New York, 2004.

- [9] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(3):411–436, 2006.
- [10] P. Del Moral and P. Lezaud. Branching and interacting particle interpretation of rare event probabilities. In Henk Blom and John Lygeros, editors, *Stochastic Hybrid Systems : Theory and Safety Critical Applications*, number 337 in Lecture Notes in Control and Information Sciences, pages 277–323. Springer–Verlag, Berlin, 2006.
- [11] M.J.J. Garvels. *The splitting method in rare event simulation*. Thesis, University of Twente, Twente, May 2000.
- [12] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.
- [13] H. Kahn and T.E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Appl. Math. Series*, 12:27–30, 1951.
- [14] A. Lagnoux. Rare event simulation. *Probability in the Engineering and Informational Sciences*, 20(1):45–66, 2006.
- [15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [16] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1993.
- [17] Copy protection technical working group. www.cptwg.org.
- [18] R. Rubinstein. The Gibbs cloner for combinatorial optimization, counting and sampling. *Methodology and Computing in Applied Probability*, 2008.
- [19] G. Tardos. Optimal probabilistic fingerprint codes. In *Proc. of the 35th annual ACM symposium on theory of computing*, pages 116–125, San Diego, 2003. ACM.
- [20] L. Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1762, 1994. With discussion and a rejoinder by the author.
- [21] M. Vill en-Altamirano and J. Vill en-Altamirano. RESTART : a straightforward method for fast simulation of rare events. In Jeffrey D. Tew, Mani S. Manivannan, Deborah A. Sadowski, and Andrew F. Seila, editors, *Proceedings of the 1994 Winter Simulation Conference, Orlando 1994*, pages 282–289, December 1994.



Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399