Appl Math Optim (2011) 64:171–196
DOI 10.1007/s00245-011-9135-z

Simulation and Estimation of Extreme Quantiles and Extreme Probabilities

Arnaud Guyader · Nicolas Hengartner · Eric Matzner-Løber

Published online: 7 April 2011

Abstract Let X be a random vector with distribution μ on $\mathbb{R}^d$ and Φ be a mapping from $\mathbb{R}^d$ to $\mathbb{R}$. That mapping acts as a black box, e.g., the result from some computer experiments for which no analytical expression is available. This paper presents an efficient algorithm to estimate a tail probability given a quantile or a quantile given a tail probability. The algorithm improves upon existing multilevel splitting methods and can be analyzed using Poisson process tools that lead to exact description of the distribution of the estimated probabilities and quantiles. The performance of the algorithm is demonstrated in a problem related to digital watermarking.

Keywords Monte Carlo simulation · Rare event · Metropolis-Hastings · Watermarking

1 Introduction

To help motivate the work we present in this paper, consider the very concrete example of digital watermarking that represents a new field of application for rare event analysis. Digital watermarking is a set of techniques for embedding information in digital files, such as audio files, images, or video files. Ideally, this embedding should minimally distort the original, be robust to corruption, be hard to remove, and most

This work was partially supported by the Nebbiano project, ANR-06-SETI-009, and by LANL LDRD 20080391ER.

A. Guyader (✉) · E. Matzner-Løber
Université Rennes 2, 35043 Rennes Cedex, France
e-mail: arnaud.guyader@uhb.fr

A. Guyader
INRIA Rennes, 35043 Rennes Cedex, France

N. Hengartner
Information Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 86545, USA

importantly, be preserved when the file is copied. Digital watermarking with these properties enable ownership attribution of digital media that is essential for digital rights management. For example, watermarking is used for copy protection by optical disk players to prevent and deter unauthorized copying of digital media by refusing to record any watermarked content (see [17] Digital Rights Management site for DVD copy). The probability of refusing to play back a file that has not been watermarked (probability of false alarm) should be very small. In 1997, the standards group for DVD copyright protection called for technologies capable of producing at most one false alarm in 400 hours of operations. As the detection rate was one decision per ten seconds, this implies a probability of false alarm of about 7×10^{-6}. Since 2001, consumer electronics manufacturers claim no error in "316,890 years", or equivalently a false positive probability of 1×10^{-12}. A fundamental problem in developing and evaluating watermarking for digital rights management is to estimate the probability of false positive by the watermarking detection scheme.

Formally, suppose that selecting a "random" (*i.e.*, unwatermarked) digital file is equivalent to drawing a random element X from a distribution μ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For that vector X, let $\Phi(X)$ be a score that is large when a watermark is detected, *i.e.*, the device considers the file as watermarked if $\Phi(X) > q$, where q is a fixed given threshold. Because of the complexity of many decoding schemes, we view Φ as a black box, that is, we do not have an analytic expression for Φ but we can readily evaluate $\Phi(X)$ for any given instance X. Then given a threshold q, we seek to estimate the probability of false alarm, defined as the tail probability $p = \mathbb{P}(\Phi(X) > q)$ when $X \sim \mu$.

A Crude Monte Carlo (CMC) that uses an i.i.d. N-sample $X_1, \ldots, X_N$ to estimate p by the fraction $\hat{p}_{mc} = \#\{i : \Phi(X_i) > q\}/N$ is not practical when p is very small. Indeed, in order to obtain a reasonable precision of the estimate given by the relative variance $V(\hat{p}_{mc})/p^2$, which is approximately equal to $1/(Np)$, one needs to select a sample size N of order p^{-1}. For instance, a random sample of one trillion observations is needed to estimate a target probability of 10^{-12}.

Importance sampling, which draws samples according to π and weights each observation $X = x$ by $w(x) = d\mu(x)/d\pi(x)$ can decrease the variance of the estimated probability which in turn greatly reduces the need for such large sample sizes. We refer to [22] for a discussion on variance reduction techniques in general and to [6] for the application of importance sampling in the context of rare events estimation. Unfortunately, when Φ is a black box, these weights cannot be computed, and hence importance sampling is not available to us.

Multilevel splitting, introduced by Kahn and Harris [18] and Rosenbluth and Rosenbluth [24], is a powerful algorithm for rare events simulations. The basic idea of multilevel splitting, adapted to our problem, is to fix a set of increasing levels $-\infty = L_0 < L_1 < L_2 < \cdots < L_m = q$, and to decompose the tail probability

$$\mathbb{P}(\Phi(X) > q) = \prod_{j=1}^{m} \mathbb{P}(\Phi(X) > L_j | \Phi(X) > L_{j-1}).$$

Each conditional probability $p_j = \mathbb{P}(\Phi(X) > L_j | \Phi(X) > L_{j-1})$ is estimated separately. We refer the reader to the paper by Glasserman et al. [14] for an in-depth

review of the importance splitting method and a detailed list of references. Two practical issues associated with the implementation of multilevel splitting are the need for computationally efficient algorithms for estimating the successive conditional probabilities, and the optimal selection of the sequence of levels.

Recently Cérou et al. [8] bridged multilevel splitting for Markovian processes and particle methods for Feynman-Kac models, thus introducing a rigorous mathematical framework for linking the sample used to estimate p_j to the one needed to estimate p_{j+1}. Within the context of Markovian processes, Cérou and Guyader [7] proposed an algorithm to adaptively select the levels in an optimal way.

Extensions of the multilevel splitting methods beyond the Markovian process context include the problem of estimating the tail probability $p = \mathbb{P}(\Phi(X) > q)$. To our knowledge, the first instance in which static rare event simulation using splitting was proposed is [2] (see also [3]). But Au and Beck call it "Subset simulation" and do not make any connection with splitting, which is why people in the rare event community do not mention this work afterwards. The next work where a reversible transition kernel was introduced to deal with such static rare events is [13]. The paper of Cérou et al. [10] proposes to adaptively select the levels using the $(1 - p_0)$-quantiles of the conditional distributions of $\Phi(X)$ given that $\Phi(X) > L_j$. The analysis of the statistical properties of $\tilde{p}$, the tail probability estimate of Cérou et al. [10], reveals that when the number of particles N tends to infinity, the expectation and variance are respectively

$$\mathbb{E}[\tilde{p}] = p + \mathcal{O}(N^{-1}) \quad \text{and} \quad V(\tilde{p}) = \frac{p^2}{N}\left(\frac{(1-p_0)\cdot\log p}{p_0\cdot\log p_0}\right) + o(N^{-1}).$$

It is noteworthy that a very similar approach has been independently proposed by Rubinstein [25] and Botev and Kroese [4] in the context of combinatorial optimization, counting and sampling, demonstrating the performance of this algorithm via an extensive simulation study (see also [5]). It bears also a resemblance to the "Nested Sampling" approach (see [12, 27, 28]) which was proposed in the context of sampling from general distributions and estimating their normalising constants.

This paper presents a refinement of the adaptive multilevel algorithm: at each iteration j, define the new level L_j as the minimum of $\Phi(\cdot)$ evaluated on the N particles, remove the particle that achieves the minimum, and use the Metropolis-Hastings algorithm to rebranch the removed particle according to the conditional distribution of $\Phi(X)$ knowing that $\{\Phi(X) > L_j\}$. This is a crucial step of the algorithm. Ideally, we would like to exactly sample from the conditional distribution of $\Phi(X)|\{\Phi(X) > L_j\}$. In practice, this is impossible. Nevertheless, we will analyze our algorithm under that very strong assumption. To avoid any misunderstanding, we will call this the idealized algorithm. Even if it does not completely match with the algorithm used in practice, it gives us an insight about the optimal performance this latter could reach. In particular, we will show that our idealized algorithm improves the current state-of-the-art algorithms.

The analysis of the idealized algorithm uses a novel technique that exploits Poisson processes to obtain an exact description of the statistical properties of the estimate for a finite number of particles N. The analysis holds for both the problem of estimating the tail probability for a given quantile and the problem of estimating the

quantile given a specified tail probability. To our knowledge, the application of multilevel splitting techniques to quantile estimation is new. Furthermore, the idealized approach enables us to produce non-asymptotic confidence intervals for the estimated quantities with respect to the number of particles.

It will be proved that the algorithm almost achieves the asymptotic efficiency of tail probability estimation, as defined for example in [15]. Moreover the variance of the tail probability estimator is $V(\hat{p}) \approx -p^2 \log p/N$, for a computational cost $C(p, N) = -kN \log N \log p$, where k is a constant that will be precised later. Firstly, this implies that our algorithm has larger computational efficiency $E = 1/(V(\hat{p})C(p, N))$ than the current multilevel splitting methods. Secondly, comparing this computational efficiency to the computational efficiency of the CMC method reveals that our algorithm is better when

$$k \log N < \frac{1-p}{p(\log p)^2}.$$

Since the right-hand side increases to infinity when p goes to zero, our algorithm beats CMC when p is small enough. For example, if $k = 10$ and $N = 200$, then our algorithm outperforms CMC when $p < 1.0 \times 10^{-4}$.

Finally, we would like to stress that our methodology fits nicely within the modern computational Bayesian paradigm, since it provides a novel tool for computing extreme quantiles of posterior distributions of univariate functions of the parameters.

The paper is organized as follows. Section 2 presents an idealized algorithm and its mathematical analysis. Section 3 discusses the practical (but imperfect!) implementation of the idealized algorithm. In Sect. 4, we compare the computational efficiency of our idealized estimator to CMC and other multilevel splitting algorithms. We illustrate the method on the Watermarking example in Sect. 5. Once again, we do completely acknowledge that the practical implementation presents an approximation of the idealized algorithm given in Sect. 2. However, when illustrating our method on the Watermarking example, we show that our numerical results match with the theoretical ones. The proofs of our results are gathered in the Appendix.

2 Main Results

Let X be a random element on $\mathbb{R}^d$ for some $d > 0$, and denote by μ its probability distribution on the underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We assume that we know how to draw i.i.d. samples from μ. Consider the function $\Phi : \mathbb{R}^d \to \mathbb{R}$ that we assume we can evaluate at any point x in $\mathbb{R}^d$.

2.1 Algorithm

Consider the following algorithm:

- Start with an i.i.d. sample $(X_1, X_2, \ldots, X_N)$ from μ and initialize $L_0 = -\infty$ and

$$X_1^1 = X_1, \ldots, X_N^1 = X_N.$$

- For $m = 1, 2, \ldots,$ set

$$L_m = \min(\Phi(X_1^m), \ldots, \Phi(X_N^m)),$$

and define for all $i = 1, 2, \ldots, N$:

$$X_i^{m+1} = \begin{cases} X_i^m & \text{if } \Phi(X_i^m) > L_m \\ X^\star \sim \mathcal{L}(X|\Phi(X) > L_m) & \text{if } \Phi(X_i^m) = L_m, \end{cases}$$

where $X^\star$ is independent of $\{X_1^m, \ldots, X_N^m\}$.
- Stopping rules:

(1) **To estimate a tail probability** p **given a quantile** q, continue until $m = M$ where $M = \max\{m : L_m \le q\}$ and set

$$\hat{p} = \left(1 - \frac{1}{N}\right)^M.$$

We will show that M is a Poisson distributed random variable.

(2) **To estimate a quantile** q **given a tail probability** p, continue until iteration

$$m = \left\lceil \frac{\log(p)}{\log(1 - N^{-1})} \right\rceil,$$

and set $\hat{q} = L_m$. Note that this time, the number of iterations is deterministic.

Remark Simulating exactly according to $\mathcal{L}(X|\Phi(X) > L_m)$ is impossible in general and we propose in Sect. 3 to do so approximately using Markov Chain Monte Carlo techniques. However, for the theoretical analysis, we will consider only the case where that simulation could be done perfectly, and we call it the idealized algorithm.

2.2 Statistical Results on the Idealized Algorithm

Suppose that the distribution μ of X and the mapping Φ are such that the univariate random variable $Y = \Phi(X)$ has continuous cumulative distribution function F for which we only assume continuity. This is the only assumption we make in the paper about the distribution of X and the transformation Φ, unless stated otherwise. We denote the survival function and the integrated hazard function of Y by $S(y) = 1 - F(y)$, and $\Lambda(y) = -\log S(y)$, respectively. The main result in this section describes the joint distribution of the levels $L_1, L_2, L_3, \ldots$ generated by our algorithm.

Theorem 1 *The random variables* $\Lambda(L_1), \Lambda(L_2), \Lambda(L_3), \ldots$ *are distributed as the successive arrival times of a Poisson process with rate* N, *that is*,

$$\Lambda(L_m) \stackrel{d}{=} \frac{1}{N} \sum_{j=1}^{m} E_j,$$

where $E_1, \ldots, E_m$, *are i.i.d. Exponential* (1).

2.2.1 Estimation of a Tail Probability

Consider the problem of estimating the tail probability $p = \mathbb{P}(\Phi(X) > q)$ for a given quantile q. Applying the results of Theorem 1 to stopping rule number 1, we obtain the following corollary:

Corollary 1 *The random variable $M = \max\{m : L_m \leq q\}$ is distributed according to a Poisson law with parameter $-N \log p$.*

Remark It follows from the corollary that $\mathbb{E}[M] = V(M) = -N \log p$. Furthermore, the classical approximation of the Poisson distribution by a Gaussian $\mathcal{N}(-N \log p, -N \log p)$ is of course valid in our context since N is assumed to be large (at least 100) and p small.

A natural estimator for the tail probability p is

$$\hat{p} = \left(1 - \frac{1}{N}\right)^M$$

and the following proposition describes its distribution.

Proposition 1 *The estimator $\hat{p}$ for the tail probability p is a discrete random variable taking values in*

$$\mathcal{S} = \left\{1, \left(1 - \frac{1}{N}\right), \left(1 - \frac{1}{N}\right)^2, \ldots\right\},$$

with probability

$$\mathbb{P}\left[\hat{p} = \left(1 - \frac{1}{N}\right)^m\right] = \frac{p^N(-N \log p)^m}{m!}, \quad m = 0, 1, 2, \ldots.$$

It follows that $\hat{p}$ is an unbiased estimator of p with variance:

$$V(\hat{p}) = p^2\left(p^{-\frac{1}{N}} - 1\right).$$

Comparing our estimator with the one obtained through CMC is instructive. Recall that the CMC estimate for the tail probability is given by

$$\hat{p}_{mc} = \frac{\hat{N}_{mc}}{N} = \frac{\#\{i \in \{1, \ldots, N\} : \Phi(X_i) > q\}}{N},$$

where N is the size of the CMC sample. The random variable $\hat{N}_{mc}$ has a Binomial distribution with parameters (N, p), and hence $\hat{p}_{mc}$ is an unbiased estimator with relative variance

$$\frac{V(\hat{p}_{mc})}{p^2} = \frac{1 - p}{Np} \approx \frac{1}{Np}.$$

The last approximation assumes that p is small and hence $1-p \approx 1$. Thus the sample size N has to be at least as large as $1/p$ in order to get a reasonable precision. Compare the latter with the relative variance of our estimator $\hat{p}$

$$\frac{V(\hat{p})}{p^2} = \left(p^{-\frac{1}{N}} - 1\right) \approx \frac{-\log p}{N},$$

when p is very small and/or N is large. This proves that, for the same precision in terms of variance of the estimator, CMC requires about $(-p \log p)^{-1}$ more particles than the method presented in this paper. However the CMC estimator has a lower complexity $C_{mc} = N$ than our algorithm whose expected complexity value is $C(N, p) = -kN \log N \log p$. As will be discussed in Sect. 4, the reduction in the variance outweighs the increased computational costs when p is small enough, making our algorithm computationally more efficient for estimating a tail probability p.

We can use Proposition 1 to derive confidence intervals for p. Let α be a fixed number between 0 and 1 and denote by $Z_{1-\alpha/2}$ the quantile of order $1-\alpha/2$ of the standard Gaussian distribution.

Proposition 2 *Let us denote*

$$\hat{p}_{\pm} = \hat{p} \exp\left(\pm \frac{Z_{1-\alpha/2}}{\sqrt{N}} \sqrt{-\log \hat{p} + \frac{Z_{1-\alpha/2}^2}{4N}} - \frac{Z_{1-\alpha/2}^2}{2N}\right),$$

then $I_{1-\alpha}(p) = [\hat{p}_-, \hat{p}_+]$ *is a* $100(1-\alpha)\%$ *confidence interval for* p.

For example, if $\alpha = 0.05$, then $Z_{1-\alpha/2} \approx 2$, and neglecting the terms in $1/N$ gives the following 95% confidence interval for p:

$$\hat{p} \exp\left(-2\sqrt{\frac{-\log \hat{p}}{N}}\right) \leq p \leq \hat{p} \exp\left(+2\sqrt{\frac{-\log \hat{p}}{N}}\right). \tag{1}$$

The asymmetry of this interval around $\hat{p}$ arises from the distribution of $\hat{p}$ that is approximately log-normal. We will illustrate this result in Sect. 5.

2.2.2 Estimation of a Large Quantile

Consider the problem of estimating the quantile q for a given p such that $\mathbb{P}(\Phi(X) > q) = p$. Using stopping rule number 2 described in Sect. 2.1, a natural estimator for the quantile q is

$$\hat{q} = L_m,$$

where $m = \lceil \frac{\log(p)}{\log(1-N^{-1})} \rceil$. Given sufficient smoothness of the distribution at the quantile q, we obtain an asymptotic normality result for our estimator.

Proposition 3 *If cdf F is differentiable at point q, with density $f(q) \neq 0$, then*

$$\sqrt{N}(\hat{q} - q) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \frac{-p^2 \log p}{f(q)^2}\right).$$

The CMC estimator defined as $\hat{q}_{mc} = Y_{(\lfloor (1-p)N \rfloor)}$, where $Y_{(1)} \leq \cdots \leq Y_{(n)}$ are the order statistics of $\Phi(X_1), \ldots, \Phi(X_N)$ and $\lfloor y \rfloor$ stands for the integer part of y, satisfies the central limit theorem (see for example [26, Theorem 7.25])

$$\sqrt{N}(\hat{q}_{mc} - q) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \frac{p(1-p)}{f(q)^2}\right).$$

This proves again that in order to achieve the same precision in terms of variance of the estimator, CMC requires about $(-p \log p)^{-1}$ more particles than the estimator proposed here. As usual the bias is in $\mathcal{O}(1/N)$ where as standard deviation is in $\mathcal{O}(1/\sqrt{N})$, so that only the term of variance is worth of attention when N is large enough. The following proposition describes the bias of our estimator. As already noticed in [31] for the CMC estimator, the estimation of the bias requires further assumptions.

Proposition 4 *If F^{-1} is twice differentiable on $(0,1)$ with continuous second derivative on $(0,1)$, if $(F^{-1})'(t) > 0$ for $t \in (0,1)$, and if there exist non-negative numbers a and b such that $F^{-1}(t)t^a(1-t)^b$ is bounded for $t \in (0,1)$, then the bias of $\hat{q}$ is bounded from below by*

$$\lim_{N\to\infty} N(\mathbb{E}[\hat{q}] - q) \geq \left(\log p - \frac{pf'(q)}{2f(q)^2}(-2 - \log p)\right)\frac{p}{f(q)},$$

and bounded from above by

$$\lim_{N\to\infty} N(\mathbb{E}[\hat{q}] - q) \leq \left(1 + \log p - \frac{pf'(q)}{2f(q)^2}(2 - \log p)\right)\frac{p}{f(q)}.$$

Remarks

1. In these inequalities, it is assumed that $f'(q) < 0$. Suitably modified upper and lower bounds are readily obtained when $f'(q) > 0$. We chose to present the results for $f'(q) < 0$, as that assumption is more likely to hold in practice.
2. The assumptions to get expressions for the bias and the variance are the same as in CMC. For this estimator, it is known from the theory of order statistics (see for example [31, Lemma 3.2.2], or [1, p. 128]) that:

$$\mathbb{E}[\hat{q}_{mc}] = q - \frac{1}{N} \cdot \frac{p(1-p)f'(q)}{2f(q)^3} + o(1/N).$$

The obtained expression for the asymptotic variance in Proposition 3 proves that $\hat{q}$ is much more precise than the CMC estimator $\hat{q}_{mc}$, but is of limited practical use as it requires the knowledge of $f(q)$. Exploiting the connection with Poisson processes allows us to derive non asymptotic confidence intervals for q without having

to estimate the density at the quantile q. Indeed, fix $\alpha \in (0,1)$, denote by $Z_{1-\alpha/2}$ the quantile of order $1-\alpha/2$ of the standard Gaussian distribution, and define

$$m_- = \left\lfloor -N\log p - Z_{1-\alpha/2}\sqrt{-N\log p} \right\rfloor,$$
$$m^+ = \left\lceil -N\log p + Z_{1-\alpha/2}\sqrt{-N\log p} \right\rceil$$

and consider L_{m_-}, L_{m^+} the associate levels. The following proposition provides a $1-\alpha$ confidence interval for q.

Proposition 5 *Under the assumptions of Theorem 1, a* $100(1-\alpha)\%$ *confidence interval for the quantile* q *is* $I_{1-\alpha}(q) = [L_{m_-}, L_{m^+}]$.

Remarks

1. The computational price to pay to obtain the confidence interval is the cost of running the algorithm until $m = m^+$ in order to get the upper confidence bound L_{m^+}. This requires the algorithm to run around $Z_{1-\alpha/2}\sqrt{-N\log p}$ additional steps.
2. Compared to Proposition 3, the great interest of this property lies in the fact that it does not require any estimation of the probability density function f.

   This result will also be illustrated in Sect. 5.

3 Practical Implementation of the Algorithm

This section explains how to generate the random variable $X^\star$ from the conditional distribution $\mathcal{L}(X|\Phi(X) > L_m)$ that is needed at each step in the algorithm. Let us recall that μ denotes the law of X. To draw $X^\star$, we run a Monte Carlo Markov Chain with a suitable μ-symmetric and one-step μ-irreducible kernel K. That is: K satisfies the detailed balance property with μ; and from any initial point x, the Radon-Nikodym derivative $dK(x,dx')/d\mu(dx')$ is strictly positive. Either, one knows such a kernel K or otherwise could use a Metropolis-Hasting kernel K based on a one-step μ-irreducible instrumental kernel $Q(x,dx')$ (see for example [22]).

Example Let us suppose that X has a standard Gaussian distribution on $\mathbb{R}$. Then let us present two ways to get such a transition kernel K:

(a) Direct construction: fix $\sigma > 0$ and denote K the transition kernel defined by

$$K(x,dx') = \sqrt{\frac{1+\sigma^2}{2\pi\sigma^2}}\exp\left(-\frac{1+\sigma^2}{2\sigma^2}\left(x' - \frac{x}{\sqrt{1+\sigma^2}}\right)^2\right)\lambda(dx'),$$

where λ stands for Lebesgue measure on $\mathbb{R}^d$. Denoting W a Gaussian standard variable, the transition $X \rightsquigarrow X'$ proposed by K is $X' = (X+\sigma W)/\sqrt{1+\sigma^2}$.

(b) Metropolis-Hastings kernel: fix $\sigma > 0$ and denote Q the transition kernel defined by

$$Q(x, dx') = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x'-x)^2}{2\sigma^2}\right) \lambda(dx') = q(x, x')\lambda(dx').$$

Denoting W a Gaussian standard variable, the transition $X \rightsquigarrow X'$ proposed by K is $X' = X + \sigma W$. Then, starting from Q, the transition kernel K constructed by Metropolis-Hastings is μ-symmetric and one-step μ-irreducible.

3.1 Application to our Algorithm

Consider that a μ-symmetric and one-step μ-irreducible transition kernel K is available. For all $m = 1, 2, \ldots,$ knowing $L_1 = \ell_1, L_2 = \ell_2, \ldots,$ consider the sets

$$A_m = \{x \in \mathbb{R}^d \text{ s.t. } \Phi(x) > \ell_m\},$$

and let us call μ_m the normalized restriction of μ on A_m:

$$\mu_m(dx) = \frac{1}{\mu(A_m)} \mathbb{1}_{A_m}(x)\, \mu(dx).$$

We define also the transition kernel K_m by:

$$K_m(x, dx') = \mathbb{1}_{A_m^c}(x)\delta_x(dx') + \mathbb{1}_{A_m}(x)(K(x, dx')\mathbb{1}_{A_m}(x') + K(x, A_m^c)\delta_x(dx')).$$

The idea behind the definition of K_m is very simple: starting from x, the kernel K proposes a transition $x \rightsquigarrow x'$. Then, if $\Phi(x') > \ell_m$, the transition is accepted, else it is rejected and x stays at the same place.

With these notations, it easy to see that the probability measure μ_m is invariant by the transition kernel K_m. Moreover, using [29, Corollary 2], K_m is also Harris recurrent, and by Theorem 13.3.3 in [21], we have for any initial distribution ν such that $\nu(A_m) = 1$

$$\left\| \nu K_m^n - \mu_m \right\| \xrightarrow[n \to +\infty]{} 0, \tag{2}$$

where $\| \cdot \|$ is the total variation norm.

In our context, let us fix $m = 1$, so that the algorithm begins with an i.i.d. sample $(X_1, X_2, \ldots, X_N)$ from μ, and initialize

$$X_1^1 = X_1, \ldots, X_N^1 = X_N.$$

In order to simplify notations, suppose that:

$$\Phi(X_1^1) < \cdots < \Phi(X_N^1),$$

so that $L_1 = \Phi(X_1^1)$ and

$$X_2^2 = X_2^1, \ldots, X_N^2 = X_N^1.$$

Knowing $L_1 = \ell_1$, the sample $(X_2^2, \ldots, X_N^2)$ is i.i.d. with distribution μ_1. Now pick at random an integer i between 2 and N and set $X_0^\star = X_i^2$. Thus $X_0^\star$ is also distributed according to μ_1, but is not independent from $\{X_2^2, X_3^2, \ldots, X_N^2\}$. In order to get independence, apply iteratively the transition kernel K_1 to $X_0^\star$. Knowing $X_i^2 = x_i^2$, one has $\delta_{x_i^2}(A_1) = 1$ since by construction $\Phi(x_i^2) > \ell_1$. As a consequence, the result given by (2) may be applied:

$$\left\| \int \delta_{x_i^2} K_1^n - \mu_1 \right\| \xrightarrow[n \to +\infty]{} 0.$$

Thus, after "enough" applications of the kernel K_1, $X_0^\star$ has mutated into a new particle $X^\star$ that is distributed according to μ_1 and is now "almost" independent from the initial position X_i^2. Denoting by $X_1^2 = X^\star$, we have constructed a sample $(X_1^2, \ldots, X_N^2)$ of i.i.d. random variables with common distribution $\mathcal{L}(X|\Phi(X) > \ell_1)$. The principle of the algorithm is to iteratively apply this simple idea.

Remarks

1. One would theoretically have to iterate K_m an infinite number of times to get independence at each step and to match perfectly with the theoretical analysis of the idealized algorithm in Sect. 2.2. This is of course unrealistic, and in practice it is applied only a finite number of times, denoted T. In the watermarking example of Sect. 5, we have applied it $T = 20$ times at each step and this led to an excellent agreement between the idealized and empirical results. However, this is certainly due to the fact that this is an extremely regular situation, and we admit that one can undoubtedly find cases where things do not happen so nicely.
2. The second remark is about the choice of the transition kernel K. To fix ideas, let us consider the toy example where X has a standard Gaussian distribution on $\mathbb{R}$, i.e., $\mu = \mathcal{N}(0, 1)$, and $\Phi(x) = x$. Two μ-symmetric kernels have been proposed. Both require to choose the value of a standard deviation parameter σ. The value of σ has in fact a great impact on the efficiency of the algorithm. Indeed, if σ is too small, then almost all of the T proposed transitions will be accepted, but since each transition corresponds (in expectation) to a small move, it will require a large T to forget the initial position. On the other side, if σ is too large, then almost all of the T proposed transitions will be rejected, but each transition corresponds (in expectation) to a huge move, so that it will require a rather low T to forget its initial position. Consequently, a trade-off has to be found for the "mixing" parameter σ. As a rule of thumb, it seems reasonable to count the proportion of accepted transitions at each step, and if this proportion is below a certain rate (say for example 20%) then one may reduce σ (say for example by a factor of 10%). This adaptive tuning is possible since K has the desired properties with respect to μ for any value of σ. In this respect, we would like to mention that there is a huge amount of literature on appropriate scaling of random walk Metropolis algorithms, dating back at least to [23].
3. Keeping the notations of the previous remark, one could think that, as the algorithm goes on and concentrates on regions with smaller and smaller probabilities, one would have to reduce the mixing parameter σ with increasing iteration. In

fact, and as will be illustrated in Sect. 5, this is not the case when dimension d is large enough: in such a situation, a region with very small probability may indeed be very large. For our purpose, one could call this phenomenon the "blessing of dimensionality", in opposition to the statistical "curse of dimensionality".

3.2 Pseudo-Code for Estimating p

We give now the pseudo-code version of the algorithm for the tail probability estimation when q is given.

Parameters

The number N of particles, the quantile q, the number T of proposed transitions, a μ-reversible kernel transition K.

Initialization

$m = 1$.

Draw an i.i.d. N-sample $(X_1^m, \ldots, X_N^m)$ of the law μ.

Sort the vector $(\Phi(X_1^m), \ldots, \Phi(X_N^m))$.

Denote $(X_1^m, \ldots, X_N^m)$ the sorted sample according to Φ and $L_1 = \Phi(X_1^m)$.

Iterations

while $L_m < q$

Pick an integer R randomly between 2 and N.

Let $X_1^{m+1} = X_R^m$.

for $t = 1 : T$

From X_1^{m+1}, draw a new particle $X^\star \sim K(X_1^{m+1}, .)$.

If $\Phi(X^\star) > L_m$, then let $X_1^{m+1} = X^\star$.

endfor

Let $(X_2^{m+1}, \ldots, X_N^{m+1}) = (X_2^m, \ldots, X_N^m)$.

Via a dichotomic search, put $\Phi(X_1^{m+1})$ at the right place in the sorted vector $(\Phi(X_2^{m+1}), \ldots, \Phi(X_N^{m+1}))$.

Denote $(X_1^{m+1}, \ldots, X_N^{m+1})$ the sorted sample according to Φ and $L_{m+1} = \Phi(X_1^{m+1})$.

$m = m + 1$.

endwhile

Output

$\hat{p} = (1 - \frac{1}{N})^{m-1}$.

---

4 Complexity, Efficiency and Asymptotic Efficiency

In this section, we mix the theoretical results of the idealized algorithm derived in Sect. 2.2 and the computational complexity of the practical algorithm exposed in the previous section. Once again, we do acknowledge that one might not find this analysis totally convincing. However it gives us an insight about our method regarding the crucial issues of complexity and efficiency.

The expected computational complexity of our algorithm is $\mathcal{O}(N \log N \log p^{-1})$:

- A sorting of the initial sample, whose cost is (in expectation) in $\mathcal{O}(N \log N)$ via a quicksort algorithm;
- Around $-N \log p$ steps (where $p = \mathbb{P}(\Phi(X) > q)$), whose cost is decomposed in:
  - T proposed kernel transitions,
  - the dichotomic search and the insertion of the new particle at the right place in the ordered sample, whose cost is in $\mathcal{O}(\log N)$ via a min-heap algorithm (see for example [19]).

By comparison, the algorithm complexity of CMC is N. The algorithm complexity of Cérou et al. [10], where at each iteration, instead of killing and branching the smallest particle, they are branching a proportion $(1 - p_0)$ $(0 < p_0 < 1$, typically $p_0 = 3/4)$ is also in $\mathcal{O}(N \log N \log p^{-1})$.

We noticed in Sect. 2.2 that our estimator $\hat{p}$ of p has a smaller variance than $\hat{p}_{mc}$ but a larger computational complexity. To take into account both computational complexity and variance, Hammersley and Handscomb [16] have proposed to define the efficiency of a Monte Carlo process as "inversely proportional to the product of the sampling variance and the amount of labour expended in obtaining this estimate." So our method is a bit more efficient than Cérou et al. [10] because the variance of $\hat{p}$ is a bit smaller than the variance of $\tilde{p}$ while sharing similar computational costs. Specifically, the proposed estimator $\hat{p}$ is computationally more efficient than the CMC estimator $\hat{p}_{mc}$ whenever

$$V(\hat{p}) \cdot C_N \leq V(\hat{p}_{mc}) \cdot C_{mc},$$

that is

$$-\frac{p^2 \log p}{N} \cdot (-kN \log N \log p) \leq \frac{p(1-p)}{N} \cdot N \quad \text{or} \quad k \log N \leq \frac{1-p}{p(\log p)^2}.$$

That inequality is satisfied when p goes to zero since the right-hand side goes then to infinity. For example, let us fix $N = 200$ and $k = 10$, then one can check numerically that the condition

$$10 \log(200) \leq \frac{1-p}{p(\log p)^2}$$

is true as soon as $p \leq 1.0 \times 10^{-4}$.

Our calculations on $\hat{p}$ enable us to derive another efficiency result for rare event probability estimation based on the asymptotic behavior of the relative variance of the estimator when the rare event probability p goes to 0. Here we will focus only on

the asymptotic efficiency, as discussed in [15]. Recall that an estimator $\hat{p}$ for the tail probability p is said to reach asymptotic efficiency if for N fixed:

$$\lim_{p\to 0} \frac{\log\left(V(\hat{p})\times C(\hat{p})\right)}{2\log p} = 1.$$

Jensen inequality shows that for any unbiased estimator:

$$\limsup_{p\to 0} \frac{\log\left(V(\hat{p})\times C(\hat{p})\right)}{2\log p} \leq 1.$$

For example, the CMC method does not reach asymptotic efficiency since:

$$\frac{\log\left(V(\hat{p}_{mc})\times C(\hat{p}_{mc})\right)}{2\log p} = \frac{\log p + \log(1-p)}{2\log p} \xrightarrow[p\to 0]{} \frac{1}{2}.$$

Thanks to Proposition 1, we get for the proposed estimator:

$$\frac{\log\left(V(\hat{p})\times C(\hat{p})\right)}{2\log p} = 1 + \frac{\log(p^{-\frac{1}{N}}-1) + \log(kN\log N\log p^{-1})}{2\log p} \xrightarrow[p\to 0]{} 1 - \frac{1}{2N}.$$

Consequently, since the number N of particles is supposed to be large, the proposed method almost reaches asymptotic efficiency.

5 Application

As indicated in the introduction, our motivations for calculating extreme quantile and tail probabilities come from problems occurring in the protection of digital contents. Here, we apply our algorithm to a well-known watermarking detector for which there exists a closed form expression for the probability of false alarm. This allows us to benchmark our method.

For this purpose, we have selected the absolute value of the normalized correlation as the score function Φ (see for example [20]), so that X is deemed watermarked whenever

$$\Phi(X) = \frac{|X^T u|}{\|X\|} > q, \tag{3}$$

where u is a secret but fixed unit vector, and X is a d-dimensional random vector with an unknown isotropic distribution. Given a threshold value q we would like to find the tail probability (stopping rule number 1) and conversely given by the manufacturer a value p, we would like to find the threshold value q such that $\mathbb{P}(\Phi(X) > q) = p$ (stopping rule number 2).

A geometrical interpretation shows that the acceptance region is a two-sheet hypercone (see Fig. 1) whose axis is given by u and whose angle is $\theta = \cos^{-1}(q)$ (with $0 < \theta < \pi/2$).

Since X has an isotropic distribution, $X/\|X\|$ has the uniform law on the unit sphere in dimension d, so that any isotropic distribution makes the job to evaluate

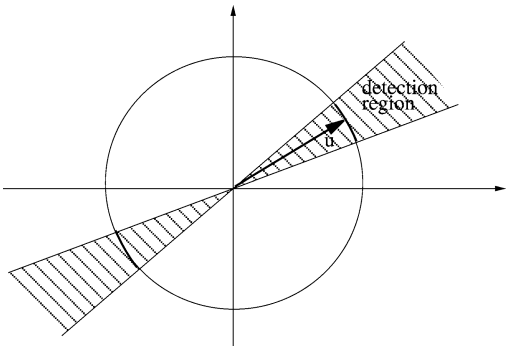

Fig. 1 Detection region for zero-bit watermarking

p or q. In the following, we propose to choose a standard Gaussian distribution: $X \sim \mathcal{N}(0, I_d)$. This allows us to derive explicit expressions for the probability of false positive detections to benchmark our algorithm. The following lemma describes the distribution of $\Phi(X)$:

Lemma 1 *Let us denote F the cdf of the random variable $Y = \Phi(X)$, G the cdf of a random variable following a Fisher-Snedecor distribution with $(1, d-1)$ degrees of freedom, f and g their respective probability densities. Then for all q in $\mathbb{R}$, we have*:

$$p = \mathbb{P}(\Phi(X) > q) = 1 - F(q) = 1 - G\left(\frac{(d-1)q^2}{1-q^2}\right),$$

from which it follows that:

$$f(q) = \frac{2(d-1)q}{(1-q^2)^2} \cdot g\left(\frac{(d-1)q^2}{1-q^2}\right).$$

In our simulations, we chose the following transition kernel for Gaussian random vectors on $\mathbb{R}^d$: Given a current location x, we propose the new position

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

where W is a $\mathcal{N}(0, I_d)$ $\mathbb{R}^d$-valued random vector and σ a positive number. In the simulations, the dimension is $d = 20$, the number of kernel transitions is $T = 20$, the numbers of particles are successively $N = 100, 200, 500, 1000, 5000$, and for each N we have run the algorithm 100 times in order to get boxplots, empirical relative standard deviations and confidence intervals. The choice $\sigma = 0.3$ has experimentally been proved to be a good trade-off for the "mixing" parameter.

Remark The fact that we do not have to tune σ on the fly might seem quite surprising at first sight. Indeed, one could think that we should reduce it adaptively since we progressively focus on smaller and smaller hypercones. Anyway, since $d = 20$, the square of the distance between a particle and the origin is distributed according to a χ^2_{20} law, which is concentrated around its mean (i.e., 20). Thus, roughly speaking,

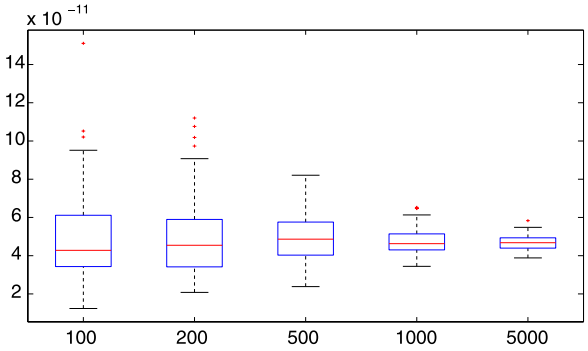

Fig. 2 Boxplots for the estimation of p obtained with 100 simulations for $N = 100$ to $N = 5{,}000$ particles

the particles are concentrated around the hypersphere centered at the origin and with radius $\sqrt{20}$. If for example $\theta = \cos^{-1}(0.95)$, then even at the end of the algorithm the distance between the axis of the hypercone and its boundary is around 1.5: this is five times larger than the standard deviation $\sigma = 0.3$ of the Gaussian moves and explains that the rate of rejection does not dramatically increase with the iterations of the algorithm.

5.1 Estimation of p

For our illustrative example, we fix $q = 0.95$ and apply Lemma 1 to conclude that the probability of interest is approximately equal to $p = 4.704 \times 10^{-11}$. Estimating such a small probability by running a CMC algorithm is of course out of question. Figure 2 summarizes the results through boxplots for our method. As the number of particles increases, the distribution of the estimator concentrates around p.

Figure 3 shows in log-log scales the theoretical and empirical relative standard deviations: the theoretical one is known thanks to Proposition 1, replacing p by the numerical value 4.704×10^{-11}, whereas the empirical one was estimated through 100 successive simulations. Let us recall that the theoretical relative standard deviations is namely

$$\frac{\sqrt{V(\hat{p})}}{p} = \sqrt{p^{-\frac{1}{N}} - 1} \approx \sqrt{\frac{-\log p}{N}},$$

the last approximation being valid when N is large enough, hence the slope equal to -0.5 on the right hand of Fig. 3. One can notice the great coincidence between theory and practice on this example.

To highlight the main difference between our method and the one proposed in [10], we have run their algorithm on the same example and with exactly the same parameters, that means: $d = 20$, $T = 20$, $\sigma = 0.3$. When the proportion of particles surviving from one step to the next is fixed to p_0, Cérou et al. prove a CLT for the estimator $\tilde{p}$ for the idealized version of their algorithm. Specifically,

$$\sqrt{N}\,(\tilde{p} - p) \xrightarrow[N\to+\infty]{\mathcal{D}} \mathcal{N}(0, \sigma^2),$$

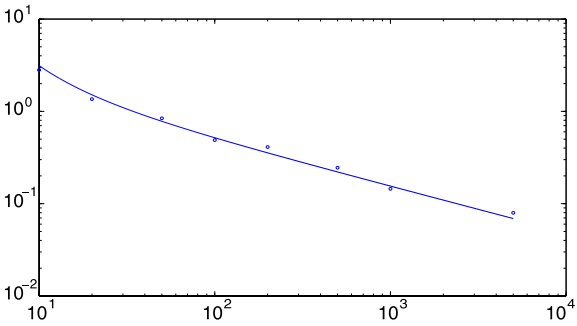

Fig. 3 Theoretical and empirical relative standard deviations with 100 simulations for $N = 10$ to $N = 5{,}000$ particles

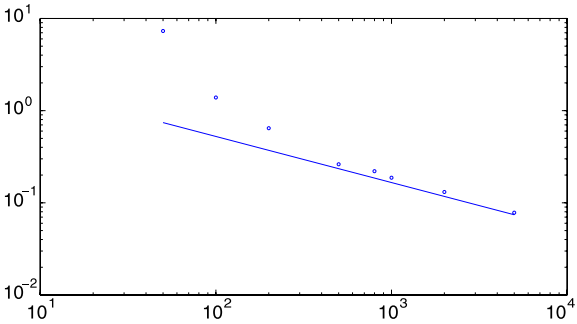

Fig. 4 Theoretical and empirical relative standard deviations with 100 simulations for $N = 50$ to $N = 5{,}000$ particles with the algorithm proposed in [10]

where

$$\sigma^2 = p^2\left(n_0\frac{1-p_0}{p_0} + \frac{1-r_0}{r_0}\right).$$

with

$$n_0 = \left\lfloor \frac{\log p}{\log p_0} \right\rfloor \quad \text{and} \quad r_0 = pp_0^{-n_0}.$$

Taking $p_0 = 0.75$ for example, it follows that $n_0 = 82$ and $r_0 \approx 0.83$. In this case, the resulting standard deviation of their estimator is only slightly larger than the standard of our estimator

$$\sqrt{n_0\frac{1-p_0}{p_0} + \frac{1-r_0}{r_0}} \approx 1.66 \gtrapprox 1.58 \approx \sqrt{-\log p}.$$

The difference becomes larger with smaller p. One consequence is that our method requires fewer particles to compute estimators for the tail probability with similar standard errors. More important, our technique gives the exact variance for as few as $N = 10$ particles, whereas the asymptotic variance proposed in [10] is reached only for $N \geq 500$. This is illustrated in Figs. 3 and 4 that graph the estimated standard deviation as a function of the number of particle (dots) for both methods, and compares it with the theoretical lower bound (line).

As a consequence, our estimator enables us to draw confidence intervals even with a low number of particles, which is not possible with the estimator proposed by Cérou

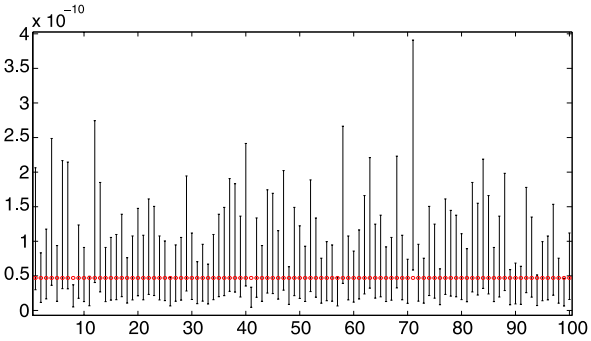

Fig. 5 95% confidence intervals for $p = 4.704 \cdot 10^{-11}$ with 100 simulations and $N = 100$ particles

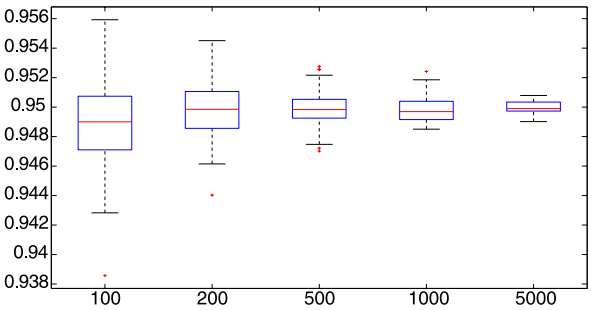

Fig. 6 Boxplots for the estimation of q obtained with 100 simulations for $N = 100$ to $N = 5{,}000$ particles

et al. In this respect, Fig. 5 illustrates the 95% confidence intervals obtained in (1) for $N = 100$ particles.

5.2 Estimation of q

Conversely, suppose that we fix $p = 4.704 \times 10^{-11}$ and seek to use our algorithm to estimate its associated tail quantile. We know that the theoretical value is $q = 0.95$. Figure 6 summarizes the results through boxplots.

Figure 7 shows in log-log scales empirical and theoretical relative standard deviations: these last ones are known thanks to Proposition 3, replacing p by the numerical value 4.704×10^{-11} and $f(q)$ by the second formula of Lemma 1. The empirical standard deviation was estimated through 100 successive simulations of the algorithm. One can notice the great coincidence between theory and practice on this example. Finally, Fig. 8 illustrates the 95% confidence intervals obtained in Proposition 5 for $N = 100$ particles.

6 Conclusion and Perspectives

We presented an efficient algorithm to estimate a tail probability given a quantile or a quantile given a tail probability. In its idealized version, the algorithm improves upon existing multilevel splitting methods and can be analyzed using Poisson process tools that lead to exact description of the distribution of the estimated probabilities

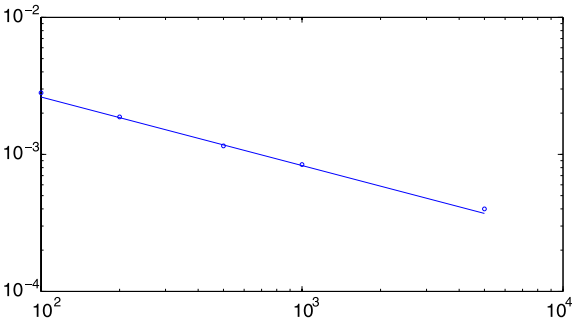

Fig. 7 Theoretical and empirical relative standard deviations with 100 simulations for $N = 100$ to $N = 5{,}000$ particles

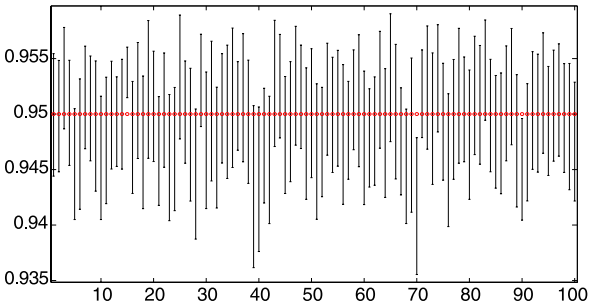

Fig. 8 95% confidence intervals for $q = 0.95$ with 100 simulations and $N = 100$ particles

and quantiles. To our knowledge, the application of multilevel splitting techniques to quantile estimation is new. Furthermore, the idealized approach enables us to produce non-asymptotic confidence intervals for the estimated quantities with respect to the number of particles. We illustrated its efficiency and accuracy on a very concrete example of digital watermarking that represents a new field of application for rare event analysis.

There are two limitations to this work: First, our analysis relies on an idealized implementation. Relaxing the assumption of perfect mixing to, say, uniform ergodicity of the Markov chain, has been considered in the recent paper by Cérou et al. [11]. While that work was developed in the context of fixed levels splitting algorithms, and does not directly apply to our context, it represents another promising way for further investigation. The second limitation is that it applies to continuous responses. The manuscript [9] suggests how to resolve that limitation in the context of counting on discrete sets associated with NP-hard discrete combinatorial problems and in particular counting the number of satisfiability assignments. The main idea is to work with an auxiliary sequence of continuous sets instead of discrete ones. The motivation of doing so is that continuous problems are typically easier than the discrete ones. Our algorithm could also be applied in this context.

Acknowledgements The authors are greatly indebted to Frédéric Cérou and Teddy Furon for valuable comments and insightful discussions on the first draft of the paper.

Appendix: Proofs of the Theoretical Results

Proof of Theorem 1 To describe the distribution of $\Lambda(L_m)$, we introduce a sequence of i.i.d. Exponential (1) random variables $E_1, E_2, E_3, \ldots$. Since Λ is assumed to be continuous with values in $]0, +\infty[$, standard calculations show that

$$\Big(\Lambda(\Phi(X_1^1)), \Lambda(\Phi(X_2^1)), \ldots, \Lambda(\Phi(X_N^1))\Big) \stackrel{d}{=} (E_1, E_2, \ldots, E_N).$$

Monotonicity of the integrated hazard entails that

$$\Lambda(L_1) = \Lambda\Big(\min_{1\le i\le N} \Phi(X_i^1)\Big) = \min_{1\le i\le N} \Lambda(\Phi(X_i^1))$$

has the same law as the minimum of N i.i.d. Exponential (1) random variables. This means that

$$\Lambda(L_1) \stackrel{d}{=} \frac{E_1}{N}.$$

Fix $m \ge 1$ and L_m and consider for $y > L_m$, the integrated hazard of the conditional distribution of $Y = \Phi(X)$, given $\Phi(X) > L_m$,

$$\Lambda_{m+1}(y) = -\log\left(\frac{1-F(y)}{1-F(L_m)}\right) = \Lambda(y) - \Lambda(L_m).$$

From the definition of X_j^{m+1}, we have again that

$$\Big(\Lambda_{m+1}(\Phi(X_1^{m+1})), \Lambda_{m+1}(\Phi(X_2^{m+1})), \ldots, \Lambda_{m+1}(\Phi(X_N^{m+1}))\Big) \stackrel{d}{=} (E_1, E_2, \ldots, E_N).$$

In light of the memoryless property of the exponential distribution, that vector is independent of $L_1, \ldots, L_m$. Hence,

$$\Lambda_{m+1}(L_{m+1}) = \Lambda(L_{m+1}) - \Lambda(L_m) \stackrel{d}{=} \frac{E_{m+1}}{N}$$

that is,

$$\Lambda(L_{m+1}) \stackrel{d}{=} \Lambda(L_m) + N^{-1}E_{m+1}.$$

Since $\Lambda_{m+1}(L_{m+1})$ is independent of $\Lambda_m(L_m)$, we get by recursion that

$$\Lambda(L_{m+1}) \stackrel{d}{=} \frac{1}{N}\sum_{j=1}^{m+1} E_j,$$

where $E_1, \ldots, E_{m+1}$ are i.i.d. Exponential (1). The random variables $T_1 = \Lambda(L_1), \ldots, T_m = \Lambda(L_m), \ldots$ can consequently be considered as the successive arrival times of a Poisson process with rate N. □

Proof of Corollary 1 The random variable M is defined as follows

$$\begin{aligned} M &= \max\{m : L_m \le q\} \\ &= \max\{m : S(L_m) \ge p\} \\ &= \max\{m : \Lambda(L_m) \le -\log p\} \\ &= \max\{m : T_m \le -\log p\}. \end{aligned}$$

Applying Theorem 1, $T_1 = \Lambda(L_1), \ldots, T_m = \Lambda(L_m), \ldots$ can be seen as the successive arrival times of a Poisson process with rate N, and hence M is simply the number of arrivals of a homogeneous Poisson point process until time $t = -\log p$. A classical result on Poisson processes implies that M is Poisson distributed with parameter $-N \log p$. □

Proof of Proposition 2 To simplify notations, let us denote respectively $\ell = \log p$ and $\hat{\ell} = \log \hat{p}$. Both ℓ and $\hat{\ell}$ are negative numbers by definition. Since $M \sim \mathcal{P}(-N \log p) \approx \mathcal{N}(-N \log p, -N \log p)$, $\hat{\ell}$ is approximately distributed as a Gaussian variable

$$\hat{\ell} = M \log\left(1 - \frac{1}{N}\right) \approx -\frac{M}{N} \overset{d}{\approx} \mathcal{N}\left(\log p, -\frac{\log p}{N}\right) = \mathcal{N}\left(\ell, -\frac{\ell}{N}\right).$$

By definition of $Z_{1-\alpha/2}$, we have that with probability $(1-\alpha)$:

$$|\ell - \hat{\ell}| \le Z_{1-\alpha/2} \sqrt{\frac{-\ell}{N}},$$

which is equivalent to

$$\ell^2 - 2\left(\hat{\ell} - \frac{Z_{1-\alpha/2}^2}{2N}\right)\ell + \hat{\ell}^2 \le 0. \tag{4}$$

The discriminant of this quadratic polynomial in ℓ is

$$\Delta = \frac{Z_{1-\alpha/2}^2}{N}\left(-\hat{\ell} + \frac{Z_{1-\alpha/2}^2}{4N}\right),$$

which is always positive since $\hat{\ell}$ is negative, so that (4) is equivalent to

$$\hat{\ell} - \frac{Z_{1-\alpha/2}^2}{2N} - \sqrt{\Delta} \le \ell \le \hat{\ell} - \frac{Z_{1-\alpha/2}^2}{2N} + \sqrt{\Delta}.$$

Taking the exponential of each term leads to the desired result

$$\hat{p} \exp\left(-\frac{Z_{1-\alpha/2}}{\sqrt{N}} \sqrt{-\log \hat{p} + \frac{Z_{1-\alpha/2}^2}{4N}} - \frac{Z_{1-\alpha/2}^2}{2N}\right) \le p$$

and

$$p \leq \hat{p} \exp\left(+ \frac{Z_{1-\alpha/2}}{\sqrt{N}} \sqrt{ - \log \hat{p} + \frac{Z_{1-\alpha/2}^2}{4N}} - \frac{Z_{1-\alpha/2}^2}{2N} \right).$$

Let us finally notice that when N is large enough, then $1/N$ is negligible compared to $1/\sqrt{N}$ and these inequalities become simply

$$\hat{p} \exp\left(- \frac{Z_{1-\alpha/2}}{\sqrt{N}} \sqrt{ - \log \hat{p}} \right) \leq p$$

and

$$p \leq \hat{p} \exp\left(+ \frac{Z_{1-\alpha/2}}{\sqrt{N}} \sqrt{ - \log \hat{p}} \right).$$

The form of this approximate confidence interval also follows from the asymptotic normality of $\log \hat{p}$

$$\sqrt{N}(\log \hat{p} - \log p) \approx \mathcal{N}(0, - \log p).$$

□

Proof of Proposition 3 By definition of $\hat{q}$, we have

$$\hat{q} = S^{-1}\left(\exp(-G_m/N)\right),$$

where $G_m = N(T_1 + \cdots + T_m)$ is a Gamma distributed random variable with rate parameter 1 and shape parameter m. The application of the Central Limit Theorem gives

$$\sqrt{m}\left(\frac{G_m}{m} - 1 \right) \xrightarrow[m \to \infty]{\mathcal{L}} \mathcal{N}(0,1).$$

The left-hand side can be expressed as follows:

$$\sqrt{m}\left(\frac{G_m}{m} - 1 \right) = \sqrt{\frac{N}{m}} \left[\sqrt{N} \left(\frac{G_m}{N} - (-\log p) \right) \right] - \frac{N}{\sqrt{m}} \left(\log p + \frac{m}{N} \right).$$

Taking into account that

$$\frac{\log(p)}{\log(1 - N^{-1})} \leq m = \left\lceil \frac{\log(p)}{\log(1 - N^{-1})} \right\rceil < 1 + \frac{\log(p)}{\log(1 - N^{-1})}$$

leads to the deterministic convergences

$$\frac{N}{\sqrt{m}} \left(\log p + \frac{m}{N} \right) \xrightarrow[N \to \infty]{} 0,$$

and

$$\sqrt{\frac{N}{m}} \xrightarrow[N \to \infty]{} \sqrt{ - \frac{1}{\log p}},$$

so that

$$\sqrt{N}\left(\frac{G_m}{N} - (-\log p)\right) \xrightarrow[m\to\infty]{\mathcal{L}} \mathcal{N}(0, -\log p).$$

We are now in a position to apply the delta method to the mapping $\varphi(x) = S^{-1}(\exp(-x))$ at point $x_0 = -\log p$, which gives

$$\sqrt{N}(\hat{q} - q) \xrightarrow[m\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \frac{-p^2 \log p}{f(q)^2}\right). \qquad \square$$

Proof of Proposition 4 We recall that $T_1, T_2, \ldots$ are the arrival times of a rate N Poisson point process, so that $G_m = N(T_1 + \cdots + T_m)$ is a Gamma distributed random variable with rate parameter 1 and shape parameter m. Since S is a one-to-one mapping, we have

$$\hat{q} = L_m = S^{-1}(\exp(-G_m/N)).$$

Upper and lower bounds for the mean are obtained by making an asymptotic expansion of S^{-1} at point p.

$$\begin{aligned}\hat{q} &= S^{-1}\left(\exp(-G_m/N)\right)\\ &= S^{-1}(p) + \left(S^{-1}\right)'(p)\left(\exp(-G_m/N) - p\right)\\ &\quad + \frac{\left(S^{-1}\right)''(p)}{2}\left(\exp(-G_m/N) - p\right)^2 + o_{\mathbb{P}}\left(\left(\exp(-G_m/N) - p\right)^2\right),\end{aligned}$$

where $X_N = o_{\mathbb{P}}(Y_N)$ means $X_N = \epsilon_N Y_N$ and ϵ_N converges to 0 in probability (see for example [30, Sect. 2.2]). Since $S^{-1}(p) = q$, $(S^{-1})'(p) = -\frac{1}{f(q)}$ and $(S^{-1})''(p) = -\frac{f'(q)}{f(q)^3}$, we can rewrite it:

$$\begin{aligned}\hat{q} &= q - \frac{1}{f(q)}\left(\exp(-G_m/N) - p\right) - \frac{f'(q)}{2f(q)^3}\left(\exp(-G_m/N) - p\right)^2\\ &\quad + o_{\mathbb{P}}\left(\left(\exp(-G_m/N) - p\right)^2\right).\end{aligned}$$

Taking the expectation on both sides leads to:

$$\begin{aligned}\mathbb{E}\left[\hat{q}\right] &= q - \frac{1}{f(q)}\left(\mathbb{E}\left[\exp(-G_m/N)\right] - p\right) - \frac{f'(q)}{2f(q)^3}\mathbb{E}\left[\left(\exp(-G_m/N) - p\right)^2\right]\\ &\quad + \mathbb{E}\left[o_{\mathbb{P}}\left(\left(\exp(-G_m/N) - p\right)^2\right)\right]. \qquad (5)\end{aligned}$$

The number of steps of the algorithm can be lower and upper bounded as follows:

$$\frac{\log(p)}{\log(1 - N^{-1})} \le m = \left\lceil \frac{\log(p)}{\log(1 - N^{-1})} \right\rceil \le 1 + \frac{\log(p)}{\log(1 - N^{-1})}.$$

Standard computations on the Gamma distribution give:

$$\mathbb{E}\left[\exp(-G_m/N)\right] = \left(1+\frac{1}{N}\right)^{-m} = \exp\left(-m\log\left(1+\frac{1}{N}\right)\right).$$

Now, taking into account that:

$$\frac{\log(1+N^{-1})}{\log(1-N^{-1})} = -1+\frac{1}{N}+o(1/N),$$

we get:

$$p-\frac{p(1+\log p)}{N}+o(1/N) \le \mathbb{E}\left[\exp(-G_m/N)\right] \le p-\frac{p\log p}{N}+o(1/N).$$

Let us now develop the second expectation term in formula (5):

$$\mathbb{E}\left[(\exp(-G_m/N)-p)^2\right] = \mathbb{E}\left[\exp(-2G_m/N)\right] - 2p\,\mathbb{E}\left[\exp(-G_m/N)\right] + p^2.$$

This time we have:

$$\mathbb{E}\left[\exp(-2G_m/N)\right] = \left(1+\frac{2}{N}\right)^{-m} = \exp\left(-m\log\left(1+\frac{2}{N}\right)\right).$$

And the asymptotic expansion:

$$\frac{\log(1+2N^{-1})}{\log(1-N^{-1})} = -2+\frac{3}{N}+o(1/N),$$

leads to:

$$\frac{(-2-\log p)p^2}{N}+o(1/N) \le \mathbb{E}\left[(\exp(-G_m/N)-p)^2\right] \le \frac{(2-\log p)p^2}{N}+o(1/N).$$

For the last term in (5), it remains to prove that:

$$\mathbb{E}\left[o_{\mathbb{P}}\left((\exp(-G_m/N)-p)^2\right)\right] = o(1/N).$$

This step requires the regularity of F'' and can be done in the same way as for the estimation of the bias in the CMC estimation of the quantile. We refer the reader to the rigorous proof given by Van Zwet [31, Chap. 3], Lemma 3.2.2. Finally, putting all pieces together, and assuming that $f'(q) < 0$, we obtain the lower bound:

$$\lim_{N\to\infty} N(\mathbb{E}\left[\hat{q}\right]-q) \ge \left(\log p - \frac{pf'(q)}{2f(q)^2}(-2-\log p)\right)\frac{p}{f(q)},$$

and the upper bound:

$$\lim_{N\to\infty} N(\mathbb{E}\left[\hat{q}\right]-q) \le \left(1+\log p - \frac{pf'(q)}{2f(q)^2}(2-\log p)\right)\frac{p}{f(q)}.$$

This concludes the proof of Proposition 4. □

Proof of Proposition 5 We have for all $k \geq 0$:

$$\mathbb{P}(L_k \leq q < L_{k+1}) = \mathbb{P}(T_k \leq -\log p < T_{k+1}).$$

For the above-mentioned Poisson process with rate N, this means that at time $-\log p$, there have been exactly k arrivals. Since the number of arrivals until time $-\log p$ is a Poisson random variable M, with $M \sim \mathcal{P}(-N \log p)$, it comes:

$$\mathbb{P}(L_k \leq q < L_{k+1}) = \mathbb{P}(M = k).$$

Thus we are simply looking for the shortest interval $[L_k, L_K]$ such that:

$$\sum_{j=k}^{K} \mathbb{P}(M = j) \geq 1 - \alpha.$$

This can be solved exactly by numerical computation or approximately thanks to the following remark: since the distribution of M is approximately Gaussian $\mathcal{N}(-N \log p, -N \log p)$, we have:

$$\mathbb{P}(-N \log p - Z_{1-\alpha/2}\sqrt{-N \log p} \leq M \leq -N \log p + Z_{1-\alpha/2}\sqrt{-N \log p}) \approx 1 - \alpha.$$

We would like to stress that, unless the number of particles is really small, the parameter $-N \log p$ is large, so that the approximation $\mathcal{P}(-N \log p) \approx \mathcal{N}(-N \log p, -N \log p)$ is excellent. The result of Proposition 5 follows if we denote

$$\begin{cases} m_- = \lfloor -N \log p - Z_{1-\alpha/2}\sqrt{-N \log p} \rfloor, \\ m^+ = \lceil -N \log p + Z_{1-\alpha/2}\sqrt{-N \log p} \rceil \end{cases}$$

and L_{m_-}, L_{m^+} the associated levels. □

Proof of Lemma 1 Denote by P_u the orthogonal projection onto u. We have

$$p = \mathbb{P}(\Phi(X) \geq q) = \mathbb{P}\left(\frac{\|P_u X\|}{\|X\|} \geq q\right) = \mathbb{P}\left(\|P_u X\|^2 > q^2 \|X\|^2\right).$$

The orthogonal decomposition of X gives:

$$p = \mathbb{P}\left(\|P_u X\|^2 > q^2(\|P_u X\|^2 + \|I - P_u X\|^2)\right),$$

which we can rewrite:

$$p = \mathbb{P}\left(\frac{\|P_u X\|^2}{\|I - P_u X\|^2/(d-1)} > (d-1)\frac{q^2}{1-q^2}\right),$$

so that finally we obtain:

$$p = \mathbb{P}\left(Z > (d-1)\frac{q^2}{1-q^2}\right),$$

where Z is a real random variable following a Fisher-Snedecor distribution with 1 and $(d-1)$ degrees of freedom. □

References

1. Arnold, B., Balakrishnan, N., Nagaraja, H.: A First Course in Order Statistics. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. Wiley, New York (1992)
2. Au, S.K., Beck, J.L.: Estimation of small failure probabilities in high dimensions by subset simulation. Probab. Eng. Mech. **16**(4), 263–277 (2001)
3. Au, S.K., Beck, J.L.: Subset simulation and its application to seismic risk based on dynamic analysis. J. Eng. Mech. **129**(8), 901–917 (2003)
4. Botev, Z.I., Kroese, D.P.: An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. Methodol. Comput. Appl. Probab. **10**(4), 471–505 (2008)
5. Botev, Z.I., Kroese, D.P.: Efficient Monte Carlo simulation via the generalized splitting method. Stat. Comput. (2011)
6. Bucklew, J.: Introduction to Rare Event Simulation. Springer Series in Statistics. Springer, New York (2004)
7. Cérou, F., Guyader, A.: Adaptive multilevel splitting for rare event analysis. Stoch. Anal. Appl. **25**(2), 417–443 (2007)
8. Cérou, F., Del Moral, P., Le Gland, F., Lezaud, P.: Genetic genealogical models in rare event analysis. ALEA Lat. Am. J. Probab. Math. Stat. **1**, 181–203 (2006)
9. Cérou, F., Guyader, A., Rubinstein, R., Vaisman, R.: Smoothed splitting method for counting (2011, submitted)
10. Cérou, F., Del Moral, P., Furon, T., Guyader, A.: Sequential Monte Carlo for rare event estimation. Stat. Comput. (2011)
11. Cérou, F., Del Moral, P., Guyader, A.: A non asymptotic theorem for unnormalized Feynman-Kac particle models. Ann. IHP (Probab. Stat.) (2011)
12. Chopin, N., Robert, C.P.: Properties of nested sampling. Biometrika **97**(3), 741–755 (2010)
13. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc., Ser. B, Stat. Methodol. **68**(3), 411–436 (2006)
14. Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T.: Multilevel splitting for estimating rare event probabilities. Oper. Res. **47**(4), 585–600 (1999)
15. Glynn, P.W., Whitt, W.: The asymptotic efficiency of simulation estimators. Oper. Res. **40**(3), 505–520 (1992)
16. Hammersley, J., Handscomb, D.: Monte Carlo Methods. Methuen, London (1965)
17. IBM (2001). www.trl.ibm.com/projects/RightsManagement/datahiding/dhvgx_e.htm
18. Kahn, H., Harris, T.: Estimation of particle transmission by random sampling. Natl. Bur. Stand., Appl. Math. Ser. **12**, 27–30 (1951)
19. Knuth, D.: The Art of Computer Programming, Sorting and Searching. Addison-Wesley Series in Computer Science and Information Processing, vol. 3. Addison-Wesley, Reading (1973)
20. Merhav, N., Sabbag, E.: Optimal watermarking embedding and detection strategies under limited detection resources. IEEE Trans. Inf. Theory **54**(1), 255–274 (2008)
21. Meyn, S., Tweedie, R.: Markov Chains and Stochastic Stability. Communications and Control Engineering Series. Springer, London (1993)
22. Robert, C., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer Texts in Statistics. Springer, New York (2004)
23. Roberts, G.O., Gelman, A., Gilks, W.R.: Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Appl. Probab. **7**(1), 110–120 (1997)
24. Rosenbluth, M., Rosenbluth, A.: Monte Carlo calculation of the average extension of molecular chains. J. Chem. Phys. **23**(2), 356–359 (1955)
25. Rubinstein, R.: The Gibbs cloner for combinatorial optimization, counting and sampling. Methodol. Comput. Appl. Probab. **11**(4), 491–549 (2009)
26. Schervish, M.J.: Theory of Statistics. Springer Series in Statistics. Springer, New York (1995)
27. Skilling, J.: Nested sampling for general Bayesian computation. Bayesian Anal. **1**(4), 833–859 (2006) (electronic)
28. Skilling, J.: Nested sampling for Bayesian computations. In: Bayesian Statistics (Oxford Sci. Publ.), vol. 8, pp. 491–524. Oxford Univ. Press, Oxford (2007)
29. Tierney, L.: Markov chains for exploring posterior distributions. Ann. Stat. **22**(4), 1701–1762 (1994) With discussion and a rejoinder by the author
30. van der Vaart, A.: Asymptotic Statistics. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge (1998)
31. Van Zwet, W.R.: Convex Transformations of Random Variables. Mathematical Centre Tracts, vol. 7. Mathematisch Centrum, Amsterdam (1964)