
Toward Optimal Stratification for Stratified Monte-Carlo Integration

Alexandra Carpentier

Statistical Laboratory, Center for Mathematical Sciences, Wilberforce Road, CB3 0WB Cambridge, United Kingdom

A.CARPENTIER@STATSLAB.CAM.AC.UK

Rémi Munos

INRIA Lille - Nord Europe, Parc Scientifique de la Haute-Borne, 40 Avenue Halley, 59650 Villeneuve d'Ascq, France

REMI.MUNOS@INRIA.FR

Abstract

We consider the problem of adaptive stratified sampling for Monte Carlo integration of a noisy function, given a finite budget n of noisy evaluations to the function. We tackle in this paper the problem of adapting to the function at the same time the number of samples into each stratum and the partition itself. More precisely, it is interesting to refine the partition of the domain in area where the noise to the function, or where the variations of the function, are very heterogeneous. On the other hand, having a (too) refined stratification is not optimal. Indeed, the more refined the stratification, the more difficult it is to adjust the allocation of the samples to the stratification, i.e. sample more points where the noise or variations of the function are larger. We provide in this paper an algorithm that selects online, among a large class of partitions, the partition that provides the optimal trade-off, and allocates the samples almost optimally on this partition.

1. Introduction

The objective of this paper is to provide an efficient strategy for integrating a noisy function F . The learner can sample n times the function. If it samples the function at a time t in a point x_t of the domain \mathcal{X} that it can choose to its convenience, it obtains the noisy sample $F(x_t, \epsilon_t)$, where ϵ_t is drawn independently at random from some distribution \mathcal{L}_{x_t} , where \mathcal{L}_x is a probability distribution that depends on x .

If the variations of the function F are known to the

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

learner, an efficient strategy is to sample more points in parts of the domain \mathcal{X} where the variations of F are larger. This intuition is explained more formally in the setting of *Stratified Sampling* (see e.g. (Rubinstein and Kroese, 2008)).

More precisely, assume that the domain \mathcal{X} is divided in $K_{\mathcal{N}}$ regions (according to the usual terminology of stratified sampling, we refer to these regions as strata) that form a partition \mathcal{N} of \mathcal{X} . It is optimal (for an oracle) to allocate a number of points in each stratum proportional to the measure of the stratum times a quantity depending of the variations of F in the stratum (see Subsection 5.5 of (Rubinstein and Kroese, 2008)). We refer to this strategy as optimal oracle strategy for partition \mathcal{N} .

The problem is that the variations of the function F in each stratum of \mathcal{N} are unknown to the learner. In the papers (Etoré and Jourdain, 2010; Grover, 2009; Carpentier and Munos, 2011a), the authors expose the problem of, at the same time, estimating the variations of F in each stratum, and allocating the samples optimally among the strata according to these estimates. Up to some variation in efficiency or assumptions, these papers provide learners that are indeed able to learn about the variations of the function and allocate optimally the samples in the strata, up to a negligible term. However, all these papers make explicit in the theoretical bounds, or at least intuitively, the existence of a natural trade-off in terms of the refinement of the partition. The more refined the partition (especially if it gets more refined where variations of F are larger), the smaller the variance of the estimate outputted by the optimal oracle strategy. However, the larger the error of an adaptive strategy with respect to this optimal oracle strategy, since the more strata there are, the harder it is to adapt to each stratum.

It is thus important to adapt also the partition to the function, and refine more the strata where variations

of the function F are larger, while at the same time limiting the number of strata. As a matter of fact, a good partition of the domain is such that, inside each stratum, the values taken by F are as homogeneous as possible (see Subsection 5.5 of (Rubinstein and Kroese, 2008)), while at the same time the number of strata is not too large.

There are some recent papers on how to stratify efficiently the space, e.g. (Glasserman et al., 1999; Kawai, 2010; Etoré et al., 2011; Carpentier and Munos, 2012a;b). More specifically, in the recent paper (Etoré et al., 2011), the authors propose an algorithm for performing this task online and efficiently. They do not provide proofs of convergence for their algorithm, but they give some properties of optimal stratified estimate when the number of strata goes to infinity, notably convergence results under the optimal allocation. They also give some intuitions on how to split efficiently the strata. Having an asymptotic vision of this problem prevents them however from giving clear directions on how exactly to adapt the strata, as well as from providing theoretical guarantees. In paper (Carpentier and Munos, 2012a), the authors propose to stratify the domain according to some preliminary knowledge on the class of smoothness of the function. They however fix the partition before sampling and thus do not consider *online* adaptation of the partition to the function. Finally, although considering online adaptation of the partition to the function, the paper (Carpentier and Munos, 2012b) considers the specific and somehow very different¹ setting where the noise ϵ to the function F is null, and where F is differentiable according to x .

Contributions: We consider in this paper the problem of designing efficiently and according to the function a partition of the space, and of allocating the samples efficiently on this partition. More precisely, our aim is to build an algorithm that allocates the samples almost in an oracle way on the best possible partition (adaptive to the function F , i.e. that solves the trade-off that we named before) in a large class of partitions. We consider in this paper the class of partition to be the set of partitions defined by a hierarchical partitioning of the domain (as for instance what was considered in (?)) for function optimization).

- We provide new, to the best of our knowledge, ideas for sampling a domain very homogeneously, i.e. such that the samples are well scattered. The sampling schemes we introduce share ideas with

¹In this setting where the function F is noiseless and very regular, efficient strategies share ideas with quasi Monte-Carlo strategies, and the number of strata should be almost equal to the budget n .

low discrepancy schemes (see e.g. (Niederreiter, 2010)), and provide some theoretic guarantees for their efficiency.

- We provide an algorithm, called Monte-Carlo Upper Lower Confidence band. We prove that it manages to at the same time select an optimal partition of the hierarchical partitioning and then to allocate the samples in this partition almost as an oracle would do. More precisely, we prove that its pseudo-risk is smaller, up to a constant, than the pseudo-risk of MC-UCB on *any* partition of the hierarchical partitioning.

The rest of the paper is organised as follows. In Section 2 we formalise the problem and introduce the notations used throughout the paper. We also remind the problem independent bound for algorithm MC-UCB. Section 3 presents algorithm MC-ULCB, and its bound on the pseudo-risk. After a technical part on notations, we introduce what we call Balanced Sampling Scheme (BSS) and a variant of it, BSS-A. These are sampling schemes for allocating samples in a random yet almost low discrepancy way, on a domain. Algorithm MC-ULCB that we present afterwards relies heavily on them. We also discuss the results, and finally conclude the paper.

2. Preliminaries

2.1. The function

Consider a noisy function $F : (x, \epsilon) \in \mathcal{X} \times \Omega \rightarrow \mathbb{R}$.

In this definition, \mathcal{X} is the domain on which the learner can *choose* in which point x to sample, and Ω is a space on which the noise to the function ϵ is defined. We define for any $x \in \mathcal{X}$ the distribution of noise ϵ conditional to x as \mathcal{L}_x . We also define a *finite* measure ν on \mathcal{X} corresponding to a σ -algebra whose sets belong to \mathcal{X} . Without loss of generality, we assume that $\nu(\mathcal{X}) = 1$ (ν is a probability measure).

The objective of the learner is to sample the domain \mathcal{X} in order to build an efficient estimate of the integral of the noisy function F according to the measure $(\nu, \mathcal{L}_x|x)$, that is to say $\int_{\mathcal{X}} \mathbb{E}_{\epsilon_x \sim \mathcal{L}_x} F(x, \epsilon_x) d(\nu)(x)$. The learner can sample sequentially the function n times, and observe noisy samples. When sampling the function at time t in x_t , it observes a noisy sample $F(x_t, \epsilon_t)$. The noise $\epsilon_t \sim \mathcal{L}_{x_t}$ conditional to x_t is independent of the previous samples $(x_i, \epsilon_i)_{i < t}$.

For any point $x \in \mathcal{X}$, define

$$g(x) = \mathbb{E}_{\epsilon \sim \mathcal{L}_x} F(x, \epsilon) \text{ and } s(x) = \sqrt{\mathbb{E}_{\epsilon \sim \mathcal{L}_x} [(F(x, \epsilon) - g(x))^2]}.$$

We state the following Assumption on the function

Assumption 1 *We assume that both g and s are bounded in absolute value by a constant f_{\max} . Let*

$v(x, \epsilon) = \frac{F(x, \epsilon) - g(x)}{s(x)}$ (if $s(x) = 0$, set $v(x, \epsilon) = 0$). We assume that $\exists b$ such that $\forall \lambda < \frac{1}{b}$,

$$\mathbb{E}_{\epsilon \sim \mathcal{L}_x} \left[\exp(\lambda v(x, \epsilon)) \right] \leq \exp\left(\frac{\lambda^2}{2(1-\lambda b)}\right), \text{ and}$$

$$\mathbb{E}_{\epsilon \sim \mathcal{L}_x} \left[\exp(\lambda v(x, \epsilon)^2 - \lambda) \right] \leq \exp\left(\frac{\lambda^2}{2(1-\lambda b)}\right).$$

Assumption 1 means that the variations coming from the noise in F , although potentially unbounded, are not too large². We believe that it is rather general. In particular, it is satisfied if F is bounded, or also for e.g. a bounded function perturbed by an additive, heterocedastic, (sub-)Gaussian noise.

2.2. Notations for a hierarchical partitioning

The strategies that we are going to consider for integration are allowed to choose where to sample the domain. In order to do that, the strategies we consider will partition the domain \mathcal{X} into strata and sample randomly in the strata. In theory the stratification is at the discretion of the strategy and can be arbitrary. However in practice, we will consider strategies that rely on given hierarchical partitioning.

Define a dyadic hierarchical partitioning \mathcal{T} of the domain \mathcal{X} . More precisely, we consider a set of partitions of \mathcal{X} at every depth $h \geq 0$: for any integer h , \mathcal{X} is partitioned into a set of 2^h strata $\mathcal{X}_{[h,i]}$, where $0 \leq i \leq 2^h - 1$. This partitioning can be represented by a dyadic tree structure, where each stratum $\mathcal{X}_{[h,i]}$ corresponds to a node $[h, i]$ of the tree (indexed by its depth h and index i). Each node $[h, i]$ has 2 children nodes $[h+1, 2i]$ and $[h+1, 2i+1]$. In addition, the strata of the children form a sub-partition of the parents stratum $\mathcal{X}_{[h,i]}$. The root of the tree corresponds to the whole domain \mathcal{X} .

We state the following assumption on the measurability and on the measure of any stratum of the hierarchical partitioning.

Assumption 2 $\forall [h, i] \in \mathcal{T}$, the stratum $\mathcal{X}_{[h,i]}$ is measurable according to the σ -algebra on which the probability measure ν is defined.

We write $w_{[h,i]}$ the measure of stratum $\mathcal{X}_{[h,i]}$, i.e. $w_{[h,i]} = \nu(\mathcal{X}_{[h,i]})$. We also assume that the hierarchical partitioning is such that all the strata of a given depth have same measure, i.e. $w_{[h,i]} = w_h$.

Assumption 3 $\forall [h, i] \in \mathcal{T}$, the children strata of $[h, i]$ are such that $w_{h+1} = \nu(\mathcal{X}_{[h+1, 2i]}) = \nu(\mathcal{X}_{[h+1, 2i+1]}) = \frac{\nu(\mathcal{X}_{[h,i]})}{2} = \frac{w_h}{2}$.

²This assumption implies that the variations induced by the noise are sub-Gaussian. It is actually slightly stronger than the usual sub-Gaussian assumption. Nevertheless, e.g. bounded random variables and Gaussian random variables satisfy it.

If for example $\mathcal{X} = [0, 1]$, a hierarchical partitioning that satisfies the previous assumptions with the Lebesgue measure is illustrated in Figure 1.

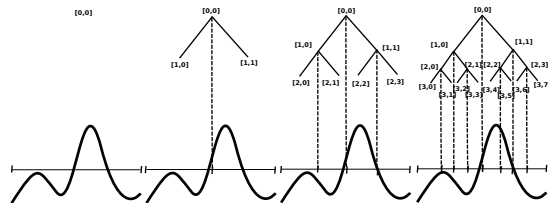


Figure 1. Example of hierarchical partitioning in dimension 1.

We write mean and variance of stratum $\mathcal{X}_{[h,i]}$ the mean and variance of a sample of the function F , collected in the point X , where X is drawn at random according to ν conditioned to stratum $\mathcal{X}_{[h,i]}$. We write

$$\mu_{[h,i]} = \mathbb{E}_{X \sim \nu_{\mathcal{X}_{[h,i]}}} \left[\mathbb{E}_{\epsilon \sim \mathcal{L}_X} [F(X, \epsilon)] \right] = \frac{1}{w_h} \int_{\mathcal{X}_{[h,i]}} g(x) d\nu(x),$$

the mean and

$$\sigma_{[h,i]}^2 = \mathbb{V}_{X \sim \nu_{\mathcal{X}_{[h,i]}}} [F(X, \epsilon)]$$

$$= \frac{1}{w_h} \int_{\mathcal{X}_{[h,i]}} (g(x) - \mu_{[h,i]})^2 d\nu(x) + \frac{1}{w_h} \int_{\mathcal{X}_{[h,i]}} s^2(x) d\nu(x),$$

the variance (we remind that g and s are defined in Assumption 1).

2.3. Pseudo-performance of an algorithm and optimal static strategies

We denote by \mathcal{A} an algorithm that allocates the budget n and returns a partition $\mathcal{N}_n = (\mathcal{X}_{[h,i]})_{[h,i] \in \mathcal{N}_n}$ included in the hierarchical partitioning \mathcal{T} of the domain. In each node $[h, i]$ of \mathcal{N}_n , algorithm \mathcal{A} allocates uniformly $T_{[h,i],n}$ random samples. We write $(X_{[h,i],t})_{[h,i] \in \mathcal{N}_n, t \leq T_{[h,i],n}}$ these samples, and we write $\hat{\mu}_{[h,i],n} = \frac{1}{T_{[h,i],n}} \sum_{t=1}^{T_{[h,i],n}} X_{[h,i],t}$ the empirical mean built with these samples. We estimate the integral of F on \mathcal{X} by $\hat{\mu}_n = \sum_{[h,i] \in \mathcal{N}_n} w_h \hat{\mu}_{[h,i],n}$. This is the estimate returned by the algorithm.

If \mathcal{N}_n is fixed as well as the number $T_{[h,i],n}$ of samples in each stratum, and if the $T_{[h,i],n}$ samples are independent and chosen uniformly according to the measure ν restricted to each stratum $\mathcal{X}_{[h,i]}$, we have

$$\mathbb{E}(\hat{\mu}_n) = \sum_{[h,i] \in \mathcal{N}_n} w_h \mu_{[h,i]} = \sum_{[h,i] \in \mathcal{N}_n} \int_{\mathcal{X}_{[h,i]}} g(u) d\nu(u) = \mu,$$

and also

$$\mathbb{V}(\hat{\mu}_n) = \sum_{[h,i] \in \mathcal{N}_n} w_h^2 \mathbb{E}(\hat{\mu}_{[h,i],n} - \mu_{[h,i]})^2 = \sum_{[h,i] \in \mathcal{N}_n} \frac{w_h^2 \sigma_{[h,i]}^2}{T_{[h,i],n}},$$

where the expectations and variance are computed with respect to the samples collected in the strata.

For a given algorithm \mathcal{A} , we denote by *pseudo-risk* the quantity

$$L_n(\mathcal{A}) = \sum_{[h,i] \in \mathcal{N}_n} \frac{w_h^2 \sigma_{[h,i]}^2}{T_{[h,i],n}}. \quad (1)$$

This measure of performance is discussed more in depths in papers (Grover, 2009; Carpentier and Munos, 2011b). In particular, paper (Carpentier and Munos, 2011b) links it with the mean squared error.

Note that if, for a given partition \mathcal{N} , an algorithm $\mathcal{A}_{\mathcal{N}}^*$ would have access the variances $\sigma_{[h,i]}^2$ of the strata in \mathcal{N} , it could allocate the budget in order to minimise the pseudo-risk, by choosing to pick in each stratum $\mathcal{X}_{[h,i]}$ (up to rounding issues) $T_{[h,i]}^* = \frac{w_h \sigma_{[h,i]} n}{\sum_{x \in \mathcal{N}} w_x \sigma_x}$ samples. The pseudo risk for this oracle strategy is then

$$L_n(\mathcal{A}_{\mathcal{N}}^*) = \frac{\left(\sum_{[h,i] \in \mathcal{N}} w_h \sigma_{[h,i]} \right)^2}{n} = \frac{\Sigma_{\mathcal{N}}^2}{n}, \quad (2)$$

where we write $\Sigma_{\mathcal{N}} = \sum_{x \in \mathcal{N}} w_x \sigma_x$. We also refer, in the sequel, as optimal allocation (for a partition \mathcal{N}), to $\lambda_{[h,i],\mathcal{N}} = \frac{w_h \sigma_{[h,i]}}{\Sigma_{\mathcal{N}}}$. Even when the optimal allocation is not realizable because of rounding issues, it can still be used as a benchmark since the quantity $L_n(\mathcal{A}_{\mathcal{N}}^*)$ is a lower bound on the variance of the estimate outputted by any oracle strategy.

2.4. Main result for algorithm MC-UCB and point of comparison

Let us consider a fixed partition \mathcal{N} of the domain, and write $K_{\mathcal{N}}$ for the number of strata it contains. We first recall (and slightly adapt) one of the main results of paper (Carpentier and Munos, 2011b) (Theorem 2). It provides a result on the pseudo-risk of an algorithm called MC-UCB. This algorithm takes some parameters linked to upper bounds on the variability of the function³, a small probability δ , and the partition \mathcal{N} . MC-UCB builds, for each stratum in the *fixed*⁴ partition \mathcal{N} , an upper confidence band (UCB) on it's standard deviation, and allocates the samples proportionnal to the measure of each stratum times this UCB. Its pseudo-risk is bounded in high probability by $\frac{\Sigma_{\mathcal{N}}^2}{n} + \Sigma_{\mathcal{N}} O\left(\frac{K_{\mathcal{N}}^{1/3}}{n^{4/3}}\right)$. This theorem holds also in our setting. The fact that the measure ν is finite together with Assumptions 2 and 1 imply that the distribution of the samples obtained by sampling in the strata are sub-Gaussian (as a bounded mixture of sub-Gaussian random variables). We remind and slightly improve this theorem.

³It is needed that the function is bounded and that the noise to the function is sub-Gaussian.

⁴It is very important to note that the partition is fixed for this algorithm and that it only adapts the allocation to the function.

Theorem 1 *Under Assumptions 2 and 1, the pseudo-risk of MC-UCB⁵ launched on partition \mathcal{N} with parameters f_{\max} , b and δ is bounded, if $n \geq 4K$, with probability $1 - \delta$,*

$$L_n(\mathcal{A}_{MC-UCB}) \leq \frac{\Sigma_{\mathcal{N}}^2}{n} + C_{\min} \Sigma_{\mathcal{N}} \sum_{x \in \mathcal{N}} \frac{w_x^{2/3}}{n^{4/3}},$$

where $C_{\min} = \frac{(4\sqrt{2}\sqrt{A} + 3f_{\max}A)}{2\sqrt{2}(1 + 3b + 4f_{\max}) \log(4n^2(3f_{\max})^3/\delta)}$ and $A =$

The bound in this Theorem is slightly sharper than in the original paper. The (improved) proof is in the Supplementary Material, see Appendix C.2

We will use in the sequel the bound in this Theorem as a benchmark for the efficiency of any algorithm that adapts the partition. The aim will be to construct a strategy whose pseudo-regret is almost as small as the minimum of this bound over a large class of partitions (e.g. the partitions defined by the hierarchical partitioning). In paper (Carpentier and Munos, 2012a), it was proved that this bound is minimax optimal which makes it a sensible benchmark.

The bound in this Theorem depends on two terms. The first, $\frac{\Sigma_{\mathcal{N}}^2}{n}$, which is the oracle optimal variance of the estimate on partition \mathcal{N} , decreases with the number of strata, and more specifically if the strata are “well-shaped” (i.e. more strata where the variations of g and s are larger). On the other hand, the second term, $\sum_{x \in \mathcal{N}} \frac{w_x^{2/3}}{n^{4/3}}$, increases when the partition is more refined. There are however two extremal situations for this term, leading to two very different behaviours with the number of strata. If the strata have all the same measure $\frac{1}{K_{\mathcal{N}}}$ where $K_{\mathcal{N}}$ is the number of strata in partition \mathcal{N} , then $\sum_{x \in \mathcal{N}} \frac{w_x^{2/3}}{n^{4/3}} = \frac{K_{\mathcal{N}}^{1/3}}{n^{4/3}}$. Now if the partition is very localised (i.e. exponential decrease of the measure of the strata), then *whatever* the number of strata, $\sum_{x \in \mathcal{N}} \frac{w_x^{2/3}}{n^{4/3}}$ is of order $O\left(\frac{1}{n^{4/3}}\right)$, and the number of strata $K_{\mathcal{N}}$ has no more influence than a constant.

These two facts enlighten the importance of adapting the *shape* of the partition to the function by having potentially strata of heterogeneous measure.

⁵In order to fit with the assumptions of this paper, we redefine $\forall x \in \mathcal{N}$ and $\forall t \leq n$ the upper confidence bound defined in the original paper as $B_{x,t} = \frac{1}{T_{x,t-1}} w_x \left(\hat{\sigma}_{x,t} + \frac{A}{\sqrt{T_{x,t}}} \right)$.

3. Algorithm MC-ULCB

3.1. Additional definitions for algorithm MC-ULCB

Let $\delta > 0$. We first define $A = 2\sqrt{2(1+3b+4f_{\max})\log(4n^2(3f_{\max})^3/\delta)}$ where f_{\max} and b are chosen such that they satisfy Assumption 1. Set also for any h , $t_h = \lfloor Aw_h^{2/3}n^{2/3} \rfloor$.

Let $[h, i]$ be a node of the hierarchical partitioning.

Assume that the children ($[h+1, 2i], [h+1, 2i+1]$) of node $[h, i]$ have received at least t_{h+1} samples (and stratum $\mathcal{X}_{[h,i]}$ has received at least $2t_{h+1}$ samples). The standard deviations $\hat{\sigma}_{[h+1,j]}$ (for $j \in \{2i, 2i+1\}$) are computed using the first t_{h+1} samples only:

$$\hat{\sigma}_{[h+1,j]} = \sqrt{\frac{1}{t_{h+1}} \sum_{u=1}^{t_{h+1}} (X_{[h+1,j],u} - \frac{1}{t_{h+1}} \sum_{k=1}^{t_{h+1}} X_{[h+1,j],k})^2},$$

where $X_{[h+1,j],u}$ is the u -th sample in stratum $\mathcal{X}_{[h+1,j]}$. We also introduce another estimate for the standard deviation $\hat{\sigma}_{[h,i]}$, namely $\tilde{\sigma}_{[h,i]}$, which is computed with the first $2t_{h+1}$ samples in stratum $\mathcal{X}_{[h,i]}$ (and not with the first t_h samples as $\hat{\sigma}_{[h,i]}$):

$$\tilde{\sigma}_{[h,i]} = \sqrt{\frac{1}{2t_{h+1}} \sum_{u=1}^{2t_{h+1}} (X_{[h,i],u} - \frac{1}{2t_{h+1}} \sum_{k=1}^{2t_{h+1}} X_{[h,i],k})^2}.$$

We use this estimate for technical purposes only. (4)

We now define by induction the value r for any stratum $\mathcal{X}_{[h,i]}$. We initialise the r when there is enough points i.e. at least t_0 points in stratum $\mathcal{X}_{[0,0]}$, by $r_{[0,0]} = \hat{\sigma}_{[0,0]} - \frac{c\sqrt{A}}{n^{1/3}}$. Assume that $r_{[h,i]}$ is defined. Whenever there are at least $t_{[h+1]}$ points in strata $\mathcal{X}_{[h+1,j]}$ for $j \in \{2i, 2i+1\}$, we define the value $r_{[h+1,j]}$ for $j \in \{2i, 2i+1\}$ (and j^- the other) as

$$\begin{aligned} r_{[h+1,j]} &= \left(\frac{w_{h+1}\hat{\sigma}_{[h+1,j]} + c\sqrt{A}\frac{w_{h+1}^{2/3}}{n^{1/3}}}{w_h\tilde{\sigma}_{[h,i]}} \right) r_{[h,i]} \quad (5) \\ &\times \mathbb{I} \left\{ w_{h+1}\hat{\sigma}_{[h+1,j^-]} - w_{h+1}\hat{\sigma}_{[h+1,j]} \geq 2c\sqrt{A}\frac{w_{h+1}^{2/3}}{n^{1/3}} \right\} \\ &+ \left(\frac{w_{h+1}\hat{\sigma}_{[h+1,j]} - c\sqrt{A}\frac{w_{h+1}^{2/3}}{n^{1/3}}}{w_h\tilde{\sigma}_{[h,i]}} \right) r_{[h,i]} \\ &\times \mathbb{I} \left\{ w_{h+1}\hat{\sigma}_{[h+1,j^-]} - w_{h+1}\hat{\sigma}_{[h+1,j]} \leq -2c\sqrt{A}\frac{w_{h+1}^{2/3}}{n^{1/3}} \right\} \\ &+ \min \left(\frac{w_{h+1} \min(\hat{\sigma}_{[h+1,j]}, \hat{\sigma}_{[h+1,j^-]}) + c\sqrt{A}\frac{w_{h+1}^{2/3}}{n^{1/3}}}{w_h\tilde{\sigma}_{[h,i]}} , \frac{1}{2} \right) r_{[h,i]} \\ &\times \mathbb{I} \left\{ |w_{h+1}\hat{\sigma}_{[h+1,j^-]} - w_{h+1}\hat{\sigma}_{[h+1,j]}| \leq 2c\sqrt{A}\frac{w_{h+1}^{2/3}}{n^{1/3}} \right\}, \end{aligned}$$

where $c = (8\tilde{\Sigma} + 1)\sqrt{A}$, $\tilde{\Sigma} = \hat{\sigma}_{[0,0]} + \frac{\sqrt{A}}{n^{1/3}}$. It is either a (proportional) upper, or a (proportional) lower confidence

bound on $w_{[h+1,j]}\sigma_{[h+1,j]}$. It is a (proportional) upper confidence bound for the stratum $[h+1, j]$ that has the smallest empirical standard deviation, and a (proportional) lower confidence bound for the other. If the quantities $w_{[h+1,j]}\hat{\sigma}_{[h+1,2i]}$ and $w_{[h+1,j]}\hat{\sigma}_{[h+1,2i+1]}$ are too close, we set the same value to both sub-strata. The quantities $r_{[h,i]}$ are key elements in algorithm MC-ULCB, and they account for the name of the algorithm (Monte Carlo Upper Lower Confidence Bound).

Additional to that, we define the technical quantities $H = \lfloor \frac{\log((3f_{\max})^3n)}{\log(2)} \rfloor + 1$, $B = 38\sqrt{2Ac}(1 + \frac{1}{\Sigma})$ and $C'_{\max} = \max(B, 14Hc\sqrt{A}) + 2\sqrt{A}$.

3.2. Sampling Schemes

The algorithm MC-ULCB that we will consider in the next Subsection works by updating a partition of the domain, refining it more where it seems necessary (i.e. where the algorithms detects that g or s have large variations). In order to do that, the algorithm needs to split some nodes $[h, i]$ in their children nodes. We thus need guarantees on the number of samples in each child node $[h+1, 2i]$ and $[h+1, 2i+1]$, when there are t samples in $[h, i]$. More precisely, we would like to have, up to rounding issues, $t/2$ samples in each child node.

The problem is that usual sampling procedures do not guarantee that. In particular, if one chooses the naive idea for sampling stratum $\mathcal{X}_{[h,i]}$, i.e. collect t samples independently at random according to $\nu_{\mathcal{X}_{[h,i]}}$, then there is no guarantees on the exact numbers of samples in $[h+1, 2i]$ and $[h+1, 2i+1]$. However, we would like that the sampling scheme that we use conserve the nice properties of sampling according to $\nu_{\mathcal{X}_{[h,i]}}$, i.e. that the empirical mean built on the samples remains an unbiased estimate of $\mu_{[h,i]}$ and that it has a variance smaller than or equal to $\sigma_{[h,i]}^2/t$.

This is one of the reasons why we need alternative sampling schemes

The Balanced Sampling Scheme We first describe what we call Balanced Sampling Scheme (BSS).

We design this sampling scheme in order to be able to divide at any time each stratum, so that at any time, the number of points in each sub-stratum is proportional to the measure of the sub-stratum (up to one sample of difference).

The proposed methodology is the following recursive procedure. Consider a stratum $\mathcal{X}_{[h,i]}$, indexed by node $[h, i]$ and that has already been sampled according to the BSS t times. It has two children in the hierarchical partitioning, namely $[h+1, 2i]$ and $[h+1, 2i+1]$. If they have been sampled a different number of times, e.g. $T_{[h+1,2i]} < T_{[h+1,2i+1]}$, we choose

the child that contains the smallest number of points, e.g. $[h + 1, 2i + 1]$, and apply BSS to this child. If the number of points in each of these nodes is equal, i.e. $T_{[h+1,2i]} = T_{[h+1,2i+1]}$, we choose uniformly at random one of these two children, and apply BSS to this child. Then we iterate the procedure in this node, until for some depth $h + l$ and node j , one has $T_{[h+l,j]} = 0$. Then when $T_{[h+l,j]} = 0$, sample randomly a point in stratum $\mathcal{X}_{[h+l,j]}$, according to $\nu_{\mathcal{X}_{[h+l,j]}}$. This provides the $(t + 1)$ th sample.

We provide in Figure 2 the pseudo-code of this recursive procedure. An immediate property is that

```

X = BSS([p, j])
if  $T_{[p+1,2j]} \neq T_{[p+1,2j+1]}$  then
    return BSS( arg min( $T_{[p+1,2j]}, T_{[p+1,2j+1]}$ ))
else if  $T_{[p+1,2j]} = T_{[p+1,2j+1]} > 0$  then
    return BSS([p + 1, 2j + B(1/2)])
else
    return  $X \sim \nu_{\mathcal{X}_{[p,j]}}$ 
endif
    
```

Figure 2. Recursive BSS procedure. $B(1/2)$ is a sample of the Bernoulli distribution of parameter $1/2$ (i.e. we sample at random among the two children strata). If stratum $[h, i]$ is sampled t times according to the BSS, any descendant stratum $[p, j]$ of $[h, i]$ is such that $T_{[p,j]} \geq \lfloor \frac{w_p}{w_h} t \rfloor \geq \frac{w_p}{w_h} t - 1$.

We also provide the following Lemma providing properties of an estimate of the empirical mean when sampling with the BSS.

Lemma 1 *Let $\mathcal{X}_{[h,i]}$ be a stratum where one samples t times according to the BSS. Then the empirical mean $\hat{\mu}_{[h,i]}$ of the samples is such that*

$$\mathbb{E}[\hat{\mu}_{[h,i]}] = \mu_{[h,i]}, \quad \text{and} \quad \mathbb{V}[\hat{\mu}_{[h,i]}] \leq \frac{\sigma_{[h,i]}^2}{t}.$$

The proof of this Lemma is in the Supplementary Material (Appendix B). This Lemma also holds for the children nodes of $[h, i]$ (for a descendant $[p, j]$, it holds with $\lfloor \frac{w_p}{w_h} t \rfloor$ samples, since the procedure is recursive).

A variant of the BSS: the BSS-A procedure
We now define a variant of the BSS: the BSS-A sampling scheme.

The reason why we need also this variant is that it is crucial, if two children of a node have obviously very different variances, to allocate more samples in the node that has higher variance. Indeed, the number of samples that one allocates to a node is directly linked to the amount of exploration that one can do of this node, and thus to the local refinement of the partitioning that one may consider. But it is also necessary to be careful and have an allocation that is more efficient than uniform allocation, as it is not sure that it is a good idea to split the parent-node. In order to

do that, we construct a scheme that uses upper confidence bounds for the less varying node, and lower confidence bounds for the most varying node: we use the $r_{[h,i]}$ that were defined for this purpose. We assume that these $r_{[h,i]}$ are defined in some sub-tree \mathcal{T}^e of the hierarchical partitioning, and undefined outside. Using such an allocation is naturally less efficient than the optimal oracle allocation, but however more efficient than uniform allocation. We illustrate this concept in Figure 3 and provide the pseudo-code in Figure 4.

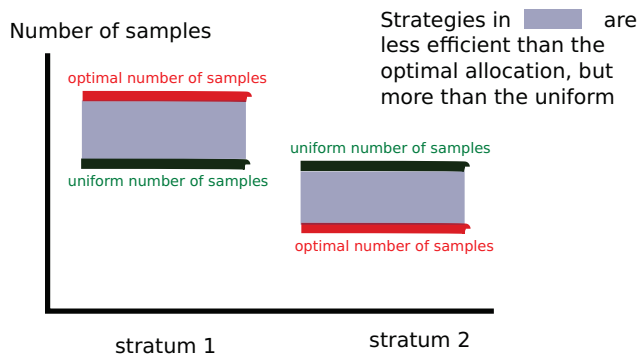


Figure 3. With high probability, the children of each node in \mathcal{N}_n are sampled a number of time that is in the gray zone by MC-ULCB.

```

X = BSS-A([p, j],  $\mathcal{T}^e$ )
if  $\{[p + 1, 2j], [p + 1, 2j + 1]\} \in \mathcal{T}^e$  then
    return BSS-A( arg min( $\frac{r_{[p+1,2j]}}{T_{[p+1,2j]}}$ ,  $\frac{r_{[p+1,2j+1]}}{T_{[p+1,2j+1]}}$ ),  $\mathcal{T}^e$ )
else
    return  $X = \text{BSS}([p, j])$ 
endif
    
```

Figure 4. Recursive BSS-A procedure.

3.3. Algorithm Monte-Carlo Upper-Lower Confidence Bound

We describe now the algorithm Monte-Carlo Upper-Lower Confidence Bound. It is decomposed in two main phases, a first Exploration Phase, and then an Exploitation Phase.

The **Exploration Phase** uses Upper and Lower Confidence bounds for allocating correctly the samples. During this phase, we update an Exploration partition, that we write \mathcal{N}_t^e , and that is included in the hierarchical partitioning. When, in a stratum $[h, i] \in \mathcal{N}_t^e$, there are more than t_h samples (also if the standard deviation of the stratum is large enough), we update \mathcal{N}_t^e by setting $\mathcal{N}_{t+1}^e = \mathcal{N}_t^e \cup [h + 1, 2i] \cup [h + 1, 2i + 1] \setminus [h, i]$: we divide $[h, i]$ in its two children strata, and compute the r corresponding to the children strata. The points are then allocated in the strata according to $\frac{r_{[h,i]}}{T_{[h,i],t}}$: a point is allocated in stratum $[h, i] \in \mathcal{N}_t^e$ if

$\frac{r_{[h,i]}^e}{T_{[h,i],t}^e} \geq \frac{4\hat{\Sigma}}{n}$. All the points are allocated inside each stratum $[h, i] \in \mathcal{N}_t^e$ according to the BSS procedure.

The Exploration Phase stops at time T , when every node $[h, i] \in \mathcal{N}_T^e$ is such that $\frac{r_{[h,i]}^e}{T_{[h,i],T+1}^e} < \frac{4\hat{\Sigma}}{n}$. We write \mathcal{T}_T^e the tree that is composed of all the nodes in \mathcal{N}_T^e and of their ancestors. The algorithm selects in this tree a partition, that we write \mathcal{N}_n , and that is an empirical minimiser (over all partitions in \mathcal{T}_T^e) of the upper bound on the regret of algorithm MC-UCB.

Finally, we perform the **Exploitation Phase** which is very similar to launching algorithm MC-UCB on \mathcal{N}_n . We pull the samples in the strata of \mathcal{N}_n according to the BSS-A sampling scheme (described in Figure 4). We compute the final estimate $\hat{\mu}_n$ of μ as a stratified estimate with respect to the deepest partition of \mathcal{T}_T^e , i.e. \mathcal{N}_T^e :

$$\hat{\mu}_n = \sum_{[h,i] \in \mathcal{N}_T^e} w_h \hat{\mu}_{[h,i],n}, \quad (6)$$

where $\hat{\mu}_{[h,i],n}$ is the empirical mean of all the samples in stratum $\mathcal{X}_{[h,i]}$.

We now provide the pseudo-code of algorithm MC-ULCB in Figure 5.

Input: f_{\max} , b and δ .
Initialization: Pull t_0 samples by BSS($[0, 0]$). Set $\mathcal{N}_t^e = \{[0, 0]\}$.
Exploration Phase:
while $\exists [h, i] \in \mathcal{N}_t^e : \frac{r_{[h,i]}^e}{T_{[h,i],t}^e} > \frac{4\hat{\Sigma}}{n}$ **do**
 Take a sample in BSS($[h, i]$).
 if $\exists [h, i] \in \mathcal{N}_t^e : \left\{ T_{[h,i],t}^e = 2t_{h+1}, w_h \hat{\sigma}_{[h,i],t} \geq 6Hc\sqrt{A} \frac{w_h^{2/3}}{n^{1/3}}, h < H \right\}$ **then**
 $\mathcal{N}_{t+1}^e = \mathcal{N}_t^e \cup [h+1, 2i] \cup [h+1, 2i+1] \setminus [h, i]$
 Compute $r_{[h+1,2i]}$ and $r_{[h+1,2i+1]}$
 end if
end while
 Select \mathcal{N}_n such that $\mathcal{N}_n = \arg \min_{\mathcal{N} \in \mathcal{T}_n^e} \left(\hat{\Sigma}_{\mathcal{N}} + (C'_{\max} - \sqrt{A}) \sum_{y \in \mathcal{N}} \frac{w_y^{2/3}}{n^{1/3}} \right)$
 $T = t$
Exploitation Phase:
for $t = T + 1, \dots, n$ **do**
 Compute $B_{[h,i],t} = \frac{w_h}{T_{[h,i],t-1}} \left(\hat{\sigma}_{[h,i]} + \sqrt{\frac{A}{n^{1/3}}} \right)$ for any $[h, i] \in \mathcal{N}_n$
 Choose a leaf $[h, i]_t$ such that $[h, i]_t = \arg \max_{[p,j] \in \mathcal{N}_n} B_{[p,j],t}$
 Pick a point according to BSS-A($[h, i]_t$)
end for
Output: $\hat{\mu}_n$

Figure 5. The pseudo-code of the Tree-MC-UCB algorithm. The empirical standard deviations and means $\hat{\sigma}_{[h,i]}$ and $\hat{\mu}_n$ and $\hat{\sigma}_{[h,i]}$ are computed using Equation 3, 6 and 4. The value of $r_{[h,i]}$ is computed using Equation 5. The BSS algorithm is described in Figure 2 and the BSS-A algorithm is described in Figure 4.

3.4. Main result

We are now going to provide the main result for the pseudo-risk of algorithm MC-ULCB.

Theorem 2 *Under Assumption 2 and 3 for the strata and 1 for the function F , the pseudo-risk of algorithm MC-ULCB is bounded with probability $1 - \delta$ as*

$$L_n(\mathcal{A}_{MC-ULCB}) \leq \sum_{[h,i] \in \mathcal{N}_n} \frac{(w_h \sigma_{[h,i]})^2}{T_{[h,i],n}} \leq \min_{\mathcal{N}} \left[\frac{\Sigma_{\mathcal{N}}^2}{n} + C'_{\max} \Sigma_{\mathcal{N}} \sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}} + C'^2_{\max} \left(\sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}} \right)^2 \right],$$

where \min means minimum over all partitions of the hierarchical partitioning, and $C'_{\max} \leq 320\sqrt{(1+3b+4f_{\max}) \log(4n^2(3f_{\max})^3/\delta)}(1/\sigma_{[0,0]} + 1)(8\sigma_{[0,0]} + 1) \log((3f_{\max})^3 n)$.

The proof of this result is in the Supplementary Material (Appendix D).

A first remark on this result is that even the first inequality (i.e. $L_n(\mathcal{A}_{MC-ULCB}) \leq \sum_{[h,i] \in \mathcal{N}_n} \frac{(w_h \sigma_{[h,i]})^2}{T_{[h,i],n}}$) is not trivial since the algorithm does not sample at random according to $\nu_{\mathcal{X}_{[h,i]}}$ in the strata $[h, i] \in \mathcal{N}_n$, but according the BSS-A. It was necessary to do that since in order to select wisely \mathcal{N}_n , one should have explored the tree \mathcal{T}_T^e , and thus it was necessary to allocate the points in order to allow splitting of the nodes and adequate exploration.

Assume that $\min_{\mathcal{N}} \Sigma_{\mathcal{N}}$ is lower bounded, e.g. the function F is noisy (i.e. the function s is not almost surely equal to 0). Then a second remark is that the second term in the final bound, namely $C'^2_{\max} \left(\sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}} \right)^2$, is negligible when compared

to the second term, namely $\sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}}$. Indeed, since $\sigma_{[0,0]}$ is bounded by Assumption 1 by f_{\max} ,

we know that $\min_{\mathcal{N}} \left[\frac{\Sigma_{\mathcal{N}}^2}{n} + C'_{\max} \Sigma_{\mathcal{N}} \sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}} \right]$

is smaller than $\frac{C'_{\max} f_{\max} + f_{\max}^2}{n}$, which implies that for one of the partitions \mathcal{N} that realises this minimum, we have $C'^2_{\max} \left(\sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}} \right)^2 \leq C'^2_{\max} \frac{(C'_{\max} f_{\max} + f_{\max}^2)^2}{n^2}$, which is negligible when compared to $n^{-4/3}$ and thus in particular $\sum_{[h,i] \in \mathcal{N}} \frac{w_h^{2/3}}{n^{4/3}}$.

3.5. Discussion

Algorithm MC-ULCB does almost as well as MC-UCB on the best partition: The result in Theorem 2 states that algorithm MC-ULCB selects adaptively a partition that is almost a minimiser of the upper bound on the pseudo-risk of algorithm MC-UCB. It then allocates almost optimally the samples

in this partition. Its upper bound on the regret is thus smaller, up to additional multiplicative term contained in C'_{\max} , than the upper bound on the regret of algorithm MC-UCB launched on an optimal partition of the hierarchical partitioning. The issue is that C'_{\max} is bigger than the constant C_{\min} for MC-UCB. More precisely, we have $C'_{\max} = C_{\min} \times C \log((3f_{\max})^3 n)$, where C is a constant depending of f_{\max} and b (see bound on C'_{\max} in Theorem 2). This additional dependency in $\log(n)$ is not an artifact of the proof and appears since we perform some model selection for selecting the partition \mathcal{N}_n . We do not know whether it is possible or not to get rid of it. Note however that a log factors already appears in the bound of MC-UCB, and that the question of whether it is or not needed remains open.

The final partition \mathcal{N}_n : Algorithm MC-ULCB refines more the partition \mathcal{N}_n in parts of the domain where splitting a stratum $[h, i]$ in a sub-partition $\mathcal{B}_{[h,i],\mathcal{N}}$ is such that $w_{[h,i]}\sigma_{[h,i]} - \sum_{x \in \mathcal{B}_{[h,i],\mathcal{N}}} w_x \sigma_x$ is large. Note that this corresponds, by definition of the $\sigma_{[h,i]}$, to parts of the domain where g and s have large variations. We do not refine the partition in regions of the domain where this is not the case, since it is more efficient to have also as few strata as possible.

The sampling schemes: The key-points in this paper are the sampling schemes. Indeed, we construct and use a sampling technique, the BSS, that is such that the samples are collected in a way that reminds low discrepancy sampling schemes⁶ on the domain, and provide an estimate such that its variance is smaller than the one of crude Monte-Carlo. We also build another sampling scheme, BSS-A. This sampling scheme ensures that, with high probability, if two children strata have very different variances, then the one with higher variance is more sampled. At the same time, it ensures that if finally the decision of splitting a stratum is not taken, then the allocation in the stratum is still better than or as efficient as random allocation according to ν restricted to the stratum.

Evaluation of the precision of the estimate and confidence intervals: An important question that one can ask here is on the possibility of constructing a confidence interval around the estimate that we obtain. What we would suggest in this case is to upper bound the pseudo-risk of the estimate by $(\sum_{x \in \mathcal{N}_n} (w_x \hat{\sigma}_x + w_x^{2/3}/n^{1/3}))^2/n$, and construct a confidence interval considering this as a bound on the

variance or the estimate, using e.g. Bennett's inequality. If e.g. the noise is symmetric, then the pseudo-risk equals the mean squared error, and the confidence interval is valid, and in particular asymptotically valid (see (Carpentier and Munos, 2011b)). Also it is less wide (up to a negligible term) than the smallest valid confidence interval on the best (oracle) stratified estimate on the hierarchical partitioning (and then in particular than the one for the crude MC estimate). Indeed, the oracle variance of such estimate is $(\inf_N \sum_{x \in \mathcal{N}} w_x \sigma_x)^2/n$ which is by definition of N_n larger or equal up to a negligible term to $(\sum_{x \in \mathcal{N}_n} w_x \sigma_x)^2/n$, and this equals up to a negligible term to the upper bound on the pseudo-risk we used to construct the confidence interval.

Conclusion

In this paper, we presented an algorithm, MC-ULCB, that aims at integrating a function in an efficient way.

MC-ULCB improves the performances of Deep-MC-UCB and returns an estimate whose pseudo-risk is smaller, up to a constant, than the minimal pseudo-risk of MC-UCB run on any partition of the hierarchical partitioning. The algorithm adapts the partition to the function and noise on it, i.e. it refines more the domain where g and s have large variations. We believe that this result is interesting since the class of hierarchical partitioning is very rich and can approximate many partition.

Acknowledgements: The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n 270327.

⁶Although the samples are chosen randomly, the sampling scheme is such that we know in a deterministic and exact way the number of samples in each not too small part of the domain.

References

- A. Carpentier and R. Munos. Finite-time analysis of stratified sampling for monte carlo. In *In Neural Information Processing Systems (NIPS)*, 2011a.
- A. Carpentier and R. Munos. Minimax number of strata for online stratified sampling given noisy samples. *Algorithmic Learning Theory*, 2012a.
- A. Carpentier and R. Munos. Adaptive stratified sampling for monte-carlo integration of differentiable functions. In *Advances in Neural Information Processing Systems 25*, 2012b.
- A. Carpentier and R. Munos. Finite-time analysis of stratified sampling for monte carlo. Technical report, INRIA-00636924, 2011b.
- Pierre Etoré and Benjamin Jourdain. Adaptive optimal allocation in stratified sampling methods. *Methodol. Comput. Appl. Probab.*, 12(3):335–360, September 2010.
- Pierre Etoré, Gersende Fort, Benjamin Jourdain, and Éric Moulines. On adaptive stratification. *Ann. Oper. Res.*, 2011. to appear.
- P. Glasserman, P. Heidelberger, and P. Shahabuddin. Asymptotically optimal importance sampling and stratification for pricing path-dependent options. *Mathematical Finance*, 9(2):117–152, 1999.
- V. Grover. Active learning and its application to heteroscedastic problems. *Department of Computing Science, Univ. of Alberta, MSc thesis*, 2009.
- R. Kawai. Asymptotically optimal allocation of stratified sampling with adaptive variance reduction by strata. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 20(2):1–17, 2010. ISSN 1049-3301.
- H. Niederreiter. *Quasi-Monte Carlo Methods*. Wiley Online Library, 2010.
- R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*. Wiley-interscience, 2008. ISBN 0470177942.