
1 Bandit View on Noisy Optimization

Jean-Yves Audibert

audibert@certis.enpc.fr

*Imagine, Université Paris Est; Willow, CNRS/ENS/INRIA
Paris, France*

Sébastien Bubeck

sebastien.bubeck@inria.fr

*Sequel Project, INRIA Lille - Nord Europe
Lille, France*

Rémi Munos

remi.munos@inria.fr

*Sequel Project, INRIA Lille - Nord Europe
Lille, France*

This chapter deals with the problem of making the best use of a finite number of noisy evaluations to optimize an unknown function. We are primarily concerned with the case where the function is defined over a finite set. In this discrete setting, we discuss various objectives for the learner, from optimizing the allocation of a given budget of evaluations to optimal stopping time problems with (ϵ, δ) -PAC guarantees. We also consider the so-called online optimization framework, where the result of an evaluation is associated to a reward, and the goal is to maximize the sum of obtained rewards. In this case, we extend the algorithms to continuous sets and (weakly) Lipschitzian functions (with respect to a prespecified metric).

1.1 Introduction

In this chapter, we investigate the problem of function optimization with a finite number of noisy evaluations. While at first, one may think that simple repeated sampling can overcome the difficulty introduced by noisy evaluations, it is far from being an optimal strategy. Indeed, to make the

best use of the evaluations, one may want to estimate more precisely the seemingly best options, while for bad options a rough estimate might be enough. This reasoning leads to non-trivial algorithms, which depend on the objective criterion that we set and how we define the budget constraint on the number of evaluations. The main mathematical tool that we use to build good strategies is a set of concentration inequalities that we briefly recall in the first section. Then in the second section, we discuss the fundamental case of discrete optimization under various budget constraints. Finally in the third section we consider the case where the optimization has to be performed online, in the sense that the value of an evaluation can be considered as a reward, and the goal of the learner is to maximize his cumulative rewards. In this case, we also consider the extension to continuous optimization.

Problem setup, notation. Consider a finite set of options $\{1, \dots, K\}$, also called actions or arms (in reference to the multi-armed bandit terminology). To each option $i \in \{1, \dots, K\}$ we associate a (reward) distribution ν_i on $[0, 1]$, with mean μ_i . Let i^* denote an optimal arm, that is $\mu_{i^*} = \max_{1 \leq j \leq K} \mu_j$. We denote by $\Delta_i = \mu_{i^*} - \mu_i$ the suboptimality gap of option i , and by $\Delta = \min_{i: \Delta_i > 0} \Delta_i$ the minimal positive gap. We assume that when one evaluates an option i , one receives a random variable drawn from the underlying probability distribution ν_i (independently from the previous draws). We investigate strategies that perform sequential evaluations of the options to find the one with highest mean. More precisely, at each time step $t \in \mathbb{N}$, a strategy chooses an option I_t to evaluate. We denote by $T_i(t)$ the number of times we evaluated option i up to time t , and by $\hat{X}_{i, T_i(t)}$ the empirical mean estimate of option i at time t (based on $T_i(t)$ i.i.d. random variables). In this chapter, we consider two different objectives for the strategy:

1. The learner possesses an evaluation budget, and once this budget is exhausted he has to select an option J as its candidate for being the best option. The performance of the learner is only evaluated through the quality of option J . This setting corresponds to the pure exploration multi-armed bandit setting, (Bubeck et al., 2009; Audibert et al., 2010). We study this problem under two different assumptions on the evaluation budget in Section 1.3.
2. The result of an evaluation is associated to a reward, and the learner wants to maximize his cumulative rewards. This setting corresponds to the classical multi-armed bandit setting, (Robbins, 1952; Lai and Robbins, 1985; Auer et al., 2002). We study this problem in Section 1.4.

1.2 Concentration inequalities

In this section, we state the fundamental concentration properties of sums of random variables. While we do not use directly the following theorems in this chapter (since we do not provide any proof), this concentration phenomenon is the corner stone of our reasoning, and a good understanding of the latter is necessary to get the insights behind our proposed algorithms.

We start with the celebrated Hoeffding-Azuma's inequality (Hoeffding, 1963) for sum of martingales differences. See for instance Williams (1991) for an introductory-level textbook on martingales and Lugosi (2004); Massart (2007) for lecture notes on concentration inequalities.

Theorem 1.1 (Hoeffding-Azuma's inequality for martingales). *Let $\mathcal{F}_1 \subset \dots \subset \mathcal{F}_n$ be a filtration, and X_1, \dots, X_n be real random variables such that X_t is \mathcal{F}_t -measurable, $\mathbb{E}(X_t | \mathcal{F}_{t-1}) = 0$ and $X_t \in [A_t, A_t + c_t]$ where A_t is a random variable \mathcal{F}_{t-1} -measurable and c_t is a positive constant. Then, for any $\epsilon > 0$, we have*

$$\mathbb{P}\left(\sum_{t=1}^n X_t \geq \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{t=1}^n c_t^2}\right), \quad (1.1)$$

or equivalently for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\sum_{t=1}^n X_t \leq \sqrt{\frac{\log(\delta^{-1})}{2} \sum_{t=1}^n c_t^2}. \quad (1.2)$$

In particular, when X_1, \dots, X_n are i.i.d. centered random variables taking their values in $[a, b]$ for some real numbers a and b , with probability at least $1 - \delta$, we have

$$\sum_{t=1}^n X_t \leq (b - a) \sqrt{\frac{n \log(\delta^{-1})}{2}}. \quad (1.3)$$

The next result is a refinement of the previous concentration inequality which takes into account the variance of the random variables. More precisely up to a second order term it replaces the range (squared) of the random variables by their variances.

Theorem 1.2 (Bernstein's inequality for martingales). *Let $\mathcal{F}_1 \subset \dots \subset \mathcal{F}_n$ be a filtration, and X_1, \dots, X_n real random variables such that X_t is \mathcal{F}_t -measurable, $\mathbb{E}(X_t | \mathcal{F}_{t-1}) = 0$, $|X_t| \leq b$ for some $b > 0$ and $\mathbb{E}(X_t^2 | \mathcal{F}_{t-1}) \leq v$*

for some $v > 0$. Then, for any $\epsilon > 0$, we have

$$\mathbb{P}\left(\sum_{t=1}^n X_t \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{2nv + 2b\epsilon/3}\right), \quad (1.4)$$

and for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\sum_{t=1}^n X_t \leq \sqrt{2nv \log(\delta^{-1})} + \frac{b \log(\delta^{-1})}{3}. \quad (1.5)$$

Inequalities (1.4) and (1.5) are two ways of expressing the concentration of the mean of i.i.d. random variables. They are almost equivalent to the extent that up to minor modification of the constants, one can go from (1.4) to (1.5) and conversely by a change of variables.

The next inequality was proved by Audibert et al. (2009). It allows to replace the true variance by its empirical estimate in Bernstein's bound.

Theorem 1.3 (Empirical Bernstein bound). *Let X_1, \dots, X_n be i.i.d. centered real random variables in $[a, b]$ for some $a, b \in \mathbb{R}$. Then for any $\delta > 0$ and $s \in \{1, \dots, n\}$, with probability at least $1 - \delta$, we have*

$$\sum_{t=1}^s X_t \leq \sqrt{2nV_s \log(3\delta^{-1})} + 3(b - a) \log(3\delta^{-1}),$$

where $V_s = \frac{1}{s} \sum_{t=1}^s (X_t - \frac{1}{s} \sum_{\ell=1}^s X_\ell)^2$.

Variants and refinement of this bound can be found in Maurer and Pontil (2009) and Audibert (2010).

1.3 Discrete optimization

In this section, we focus on strategies that use a finite budget of evaluations to find the best option. We consider two different (but related) assumptions on this budget:

- There is a fixed budget of n evaluations (Bubeck et al., 2009; Audibert et al., 2010). The value of n can be known or unknown by the learner. When it is unknown, the learner has thus to design an anytime strategy, that is a policy with good theoretical guarantees whatever the budget is.
- The strategy must stop as soon as possible with the guarantee that an ϵ -optimal option has been found with probability at least $1 - \delta$, where ϵ and δ are fixed before the procedure starts (Maron and Moore, 1993; Domingo et al., 2002; Dagum et al., 2000; Even-Dar et al., 2006; Mnih et al., 2008).

Let $A_1 = \{1, \dots, K\}$, $\overline{\log}(K) = \frac{1}{2} + \sum_{i=2}^K \frac{1}{i}$, $n_0 = 0$ and for $k \in \{1, \dots, K-1\}$,

$$n_k = \left\lceil \frac{1}{\overline{\log}(K)} \frac{n-K}{K+1-k} \right\rceil.$$

For each phase $k = 1, 2, \dots, K-1$:

- (1) For each $i \in A_k$, select option i for $n_k - n_{k-1}$ evaluations.
- (2) Let $A_{k+1} = A_k \setminus \arg \min_{i \in A_k} \hat{X}_{i, n_k}$ (we only remove one element from A_k , if there is a tie, select randomly the option to dismiss among the worst options).

Recommend the unique element of A_K .

Figure 1.1: SR (Successive Rejects) algorithm.

1.3.1 Fixed Budget

In this section, the number of evaluations is fixed, and the goal is to make the best use of this budget. We propose a simple strategy, yet almost optimal in a strong sense, see Theorem 1.4 below. The algorithm, called SR (Successive Rejects) is described precisely in Figure 1.1. Informally it proceeds as follows. First the algorithm divides the budget (i.e., the n evaluations) in $K-1$ phases. At the end of each phase, the algorithm dismisses the option with the lowest empirical mean. During the next phase, it evaluates equally often all the options which have not been dismissed yet. The recommended arm J is the last surviving option. The length of the phases are carefully chosen to obtain an optimal (up to a logarithmic factor) convergence rate. More precisely, one option is evaluated $n_1 = \lceil \frac{1}{\overline{\log}(K)} \frac{n-K}{K} \rceil$ times, one $n_2 = \lceil \frac{1}{\overline{\log}(K)} \frac{n-K}{K-1} \rceil$ times, ..., and two options are evaluated $n_{K-1} = \lceil \frac{1}{\overline{\log}(K)} \frac{n-K}{2} \rceil$ times. SR does not exceed the budget of n evaluations, since, from the definition $\overline{\log}(K) = \frac{1}{2} + \sum_{i=2}^K \frac{1}{i}$, we have

$$n_1 + \dots + n_{K-1} + n_{K-1} \leq K + \frac{n-K}{\overline{\log}(K)} \left(\frac{1}{2} + \sum_{k=1}^{K-1} \frac{1}{K+1-k} \right) = n.$$

Theorem 1.4 (Successive Rejects). *Assume that there is a unique arm i^* with maximal mean and let $H = \frac{1}{\Delta} + \sum_{i \neq i^*} \frac{1}{\Delta_i}$. Then the probability of error of SR satisfies*

$$\mathbb{P}(J \neq i^*) \leq \frac{K(K-1)}{2} \exp\left(-\frac{n-K}{\log(2K)H}\right). \quad (1.6)$$

Moreover if ν_1, \dots, ν_K are Bernoulli distributions with parameters in $[p, 1 - p]$, $p \in (0, 1/2)$, then for any strategy, there exists a permutation $\sigma : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$ such that the probability of error of the strategy on the problem defined by $\tilde{\nu}_1 = \nu_{\sigma(1)}, \dots, \tilde{\nu}_K = \nu_{\sigma(K)}$ satisfies

$$\mathbb{P}(J \neq i^*) \geq \exp\left(-\frac{(5 + o(1))n \log(2K)}{p(1-p)H}\right), \quad (1.7)$$

where the $o(1)$ term depends only on K and n and goes to 0 when n goes to infinity.

Interpretation of Theorem 1.4. Essentially, equation (1.6) indicates that if the number of evaluations is of order of $H \log^2 K$ then SR finds the best option with high probability. On the other hand, equation (1.7) shows that it is statistically impossible to find the best option with less than (of order of) $H/\log K$ evaluations. Thus H is a good measure of the *hardness* of the task, it characterizes the order of magnitude of the number of evaluations required to find the best option with a reasonable probability.

Closing the logarithmic gap between the upper and lower bounds in Theorem 1.4 is an open problem. Audibert et al. (2010) exhibit an algorithm which requires only (of order of) $H \log n$ evaluations to find the best option with high probability. However this algorithm needs to know the value of H to tune its parameters. One can overcome this difficulty by trying to estimate H online, this leads to the algorithm Adaptive UCB-E that we describe precisely in Figure 1.2. We do not give any further details about this algorithm and refer the interested reader to (Audibert et al., 2010), but we simply point out that in our numerical simulations, Adaptive UCB-E outperformed SR.

Anytime versions of SR and Adaptive UCB-E. Both algorithms that we propose heavily depend on the knowledge of the number of evaluations n . However in many natural cases this number is only implicitly defined, for instance through CPU time. Thus it is important to also have strategies which do not need to know in advance the time horizon. One simple and famous trick for this purpose is the doubling trick. The idea is to introduce meta-phases, $s = 1, 2, \dots$, such that from the evaluations $t = 2^{s-1} + 1$ to $t = 2^s$, one runs a new instance of the algorithm with n replaced by 2^{s-1} . While it is often assumed that the new instance of the algorithm does not use the samples obtained in the previous phases, here we do not need to make this assumption. For instance the anytime version of SR would work as follows. At time 2^s there is only one surviving option. Then at time $2^s + 1$

Parameter: exploration rate $c > 0$.

Definitions: For $k \in \{1, \dots, K-1\}$, let $n_k = \lceil \frac{1}{\log(K)} \frac{n-K}{K+1-k} \rceil$, $t_0 = 0$, $t_1 = Kn_1$, and for $k > 1$, $t_k = n_1 + \dots + n_{k-1} + (K-k+1)n_k$. For $i \in \{1, \dots, K\}$ and $a > 0$, let $B_{i,s}(a) = \widehat{X}_{i,s} + \sqrt{\frac{a}{s}}$ for $s \geq 1$ and $B_{i,0} = +\infty$.

Algorithm: For each phase $k = 0, 1, \dots, K-1$:
Let $\widehat{H}_k = K$ if $k = 0$, and otherwise

$$\widehat{H}_k = \max_{K-k+1 \leq i \leq K} i \widehat{\Delta}_{\langle i \rangle}^{-2},$$

where $\widehat{\Delta}_i = (\max_{1 \leq j \leq K} \widehat{X}_{j, T_j(t_k)}) - \widehat{X}_{i, T_i(t_k)}$ and $\langle i \rangle$ is an ordering such that $\widehat{\Delta}_{\langle 1 \rangle} \leq \dots \leq \widehat{\Delta}_{\langle K \rangle}$.

For $t = t_k + 1, \dots, t_{k+1}$:

Evaluate $I_t \in \operatorname{argmax}_{i \in \{1, \dots, K\}} B_{i, T_i(t-1)}(cn/\widehat{H}_k)$.

Recommendation: Let $J \in \operatorname{argmax}_{i \in \{1, \dots, K\}} \widehat{X}_{i, T_i(n)}$.

Figure 1.2: Adaptive UCB-E (Upper Confidence Bound Exploration)

we “revive” all the options and run SR with n replaced by 2^{s+1} (to define the length of the phases of SR), however the empirical mean of each option is computed over the whole run of the algorithm, starting with $t = 1$.

1.3.2 Hoeffding and Bernstein Races

Racing algorithms aim to reduce the computational burden of performing tasks such as model selection using a hold-out set by discarding poor models quickly (Maron and Moore, 1993; Ortiz and Kaelbling, 2000). A racing algorithm either terminates when it runs out of time (i.e. at the end of the n -th round) or when it can say that with probability at least $1 - \delta$, it has found the best option, i.e. an option $i^* \in \operatorname{argmax}_{i \in \{1, \dots, K\}} \mu_i$. The goal is to stop as soon as possible, and the time constraint n is just here to stop the algorithm when the two best options have (almost) equal mean rewards.

The Hoeffding race introduced by Maron and Moore (1993) is an algorithm based on discarding options which are likely to have smaller mean than the optimal one until only one option remains. Precisely, for each time step and each option i , $\delta/(nK)$ -confidence intervals are constructed for the mean μ_i . Options with upper confidence bound smaller than the lower confidence bound of another option are discarded. The algorithm samples one by one all the options that have not been discarded yet. The process is detailed in Figure 1.3. The correctness of this algorithm is proved by Maron and Moore (1993), while its sample complexity is given by the following theorem (Even-

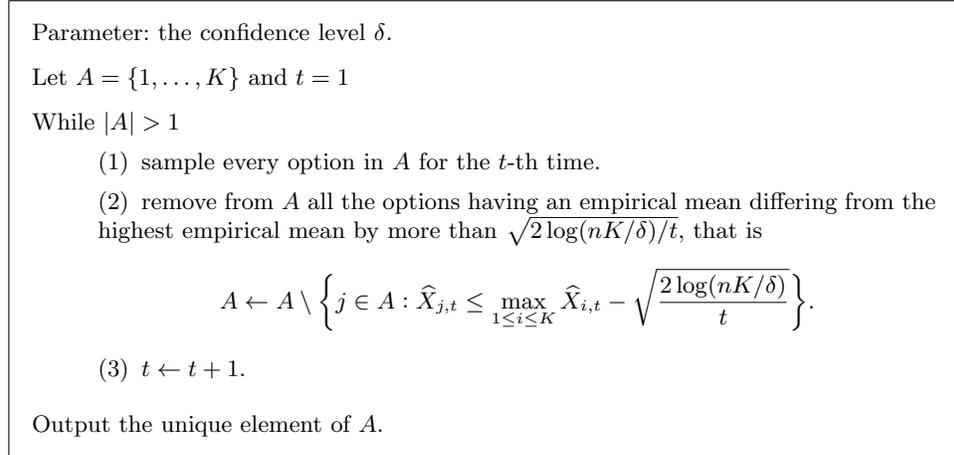


Figure 1.3: Hoeffding race

Dar et al., 2006; Mnih et al., 2008).

Theorem 1.5 (Hoeffding race). *With probability at least $1 - \delta$, the optimal option is not discarded, and the non-discarded option(s) (which can be multiple when the algorithm runs out of time) satisfy(ies)*

$$\Delta_i = O\left(\sqrt{\frac{\log(nK/\delta)}{n/K}}\right).$$

Besides, if there is a unique optimal arm i^* , with probability at least $1 - \delta$, the Hoeffding race stops after at most $O\left(\sum_{i \neq i^*} \frac{1}{\Delta_i^2} \log\left(\frac{nK}{\delta}\right)\right)$ time steps.

Empirical and theoretical studies show that replacing the Hoeffding inequality by the empirical Bernstein bound to build the confidence intervals generally leads to significant improvements. The algorithm based on the empirical Bernstein bound is described in Figure 1.4. The following theorem provides its theoretical guarantee while Table 1.1 shows the percentage of work saved by each method ($1 -$ number of samples taken by method divided by nK), as well as the number of options remaining after termination (see Mnih et al. (2008) for a more detailed description of the experiments).

Theorem 1.6 (Bernstein race). *Let σ_i denote the standard deviation of ν_i . With probability at least $1 - \delta$, the optimal option is not discarded, and the non-discarded option(s) (which can be multiple when the algorithm runs out*

of time) satisfy(ies)

$$\Delta_i = O\left((\sigma_i + \sigma_{i^*})\sqrt{\frac{\log(nK/\delta)}{n/K}} + \frac{\log(nK/\delta)}{n/K}\right).$$

Besides, if there is a unique optimal arm i^* , with probability at least $1 - \delta$, the Bernstein race stops after at most $O\left(\sum_{i \neq i^*} \frac{\sigma_i^2 + \sigma_{i^*}^2 + \Delta_i}{\Delta_i^2} \log\left(\frac{nK}{\delta}\right)\right)$ time steps.

Parameter: the confidence level δ .

Let $A = \{1, \dots, K\}$ and $t = 1$

While $|A| > 1$

- (1) sample every option in A for the t -th time.
- (2) remove from A suboptimal options:

$$A \leftarrow A \setminus \left\{ j \in A : \widehat{X}_{j,t} + \sqrt{\frac{2V_{j,t} \log(nK/\delta)}{t}} + 6 \frac{\log(nK/\delta)}{t} \leq \max_{1 \leq i \leq K} \left(\widehat{X}_{i,t} - \sqrt{\frac{2V_{i,t} \log(nK/\delta)}{t}} \right) \right\},$$

where $V_{i,t} = \frac{1}{t} \sum_{s=1}^t (X_{i,s} - \widehat{X}_{i,t})^2$ is the empirical variance of option i .

- (3) $t \leftarrow t + 1$.

Output the unique element of A .

Figure 1.4: Bernstein race

Table 1.1: Percentage of work saved / number of options left after termination.

Data set	Hoeffding	Empirical Bernstein
SARCOS	0.0% / 11	44.9% / 4
Coverttype2	14.9% / 8	29.3% / 5
Local	6.0% / 9	33.1% / 6

1.3.3 Optimal Stopping Times

The next section takes a step back since it considers the single option case (that is, when $K = 1$). The additive and multiplicative stopping time problems are there tackled. Section 1.3.3.2 then deals with the multiple options case for the additive stopping time problem.

1.3.3.1 For a single option

Algorithms described in the previous section rely on either the Hoeffding or the (empirical) Bernstein inequality and a probabilistic union bound corresponding to both the different options and the different time steps. Maximal inequalities based on a martingale argument due to Doob (1953) (see also Freedman (1975) for maximal inequalities more similar to the one below) allow to reduce the impact on the confidence levels of the union bound across time steps. Precisely, one can write the following version of the empirical Bernstein inequality, which holds uniformly over time.

Theorem 1.7. *Let X_1, \dots, X_n be $n \geq 1$ i.i.d. random variables taking their values in $[a, b]$. Let $\mu = \mathbb{E}X_1$ be their common expected value. For any $1 \leq t \leq n$, introduce the empirical mean \hat{X}_t and variance V_t defined respectively by*

$$\hat{X}_t = \frac{\sum_{i=1}^t X_i}{t} \quad \text{and} \quad V_t = \frac{\sum_{i=1}^t (X_i - \hat{X}_t)^2}{t}.$$

For any $x > 0$, with probability at least

$$1 - 3 \inf_{1 < \alpha \leq 3} \min\left(\frac{\log n}{\log \alpha}, n\right) e^{-x/\alpha}, \quad (1.8)$$

the following inequality holds simultaneously for any $t \in \{1, 2, \dots, n\}$:

$$|\hat{X}_t - \mu| \leq \sqrt{\frac{2V_t x}{t}} + \frac{3(b-a)x}{t}. \quad (1.9)$$

This theorem allows to address the additive stopping time problem in which the learner stops sampling an unknown distribution ν supported in $[a, b]$ as soon as it can output an estimate $\hat{\mu}$ of the mean μ of ν with additive error at most ϵ with probability at least $1 - \delta$, that is

$$\mathbb{P}(|\hat{\mu} - \mu| \leq \epsilon) \geq 1 - \delta, \quad (1.10)$$

with the time constraint that the learner is not allowed to sample more than n times. Indeed, from Theorem 1.7, it suffices to stop sampling at time t such that the right-hand side of (1.9) is below ϵ where x is set such that (1.8)

equals $1 - \delta$. Besides, it can be shown that the sampling complexity is in expectation

$$O\left(\left(\log(\delta^{-1}) + \log(\log(3n))\right) \max\left(\frac{\sigma^2}{\epsilon^2}, \frac{b-a}{\epsilon}\right)\right),$$

where σ^2 is the variance of the sampling distribution. This is optimal up to the $\log \log$ term.

In the multiplicative stopping time problem, the learner stops sampling an unknown distribution ν supported in $[a, b]$ as soon as it can output an estimate $\hat{\mu}$ of the mean μ of ν with relative error at most ϵ with probability at least $1 - \delta$, that is

$$\mathbb{P}(|\hat{\mu} - \mu| \leq \epsilon|\mu|) \geq 1 - \delta, \tag{1.11}$$

with the time constraint that the learner is not allowed to sample more than n times. The multiplicative stopping time problem is similar to the additive one, except when μ is close to 0 (but nonzero). Considering relative errors introduces an asymmetry between the left and right bound of the confidence intervals, which requires more involved algorithms to get better practical performances. The state-of-the-art method to handle the task is the geometric empirical Bernstein stopping proposed by Mnih et al. (2008) and detailed in Figure 1.5. A slightly refined version is given in Audibert (2010).

It uses a geometric grid and parameters ensuring that the event $\mathcal{E} = \{|\hat{X}_t - \mu| \leq c_t, t \geq t_1\}$ occurs with probability at least $1 - \delta$. It operates by maintaining a lower bound, LB, and an upper bound, UB, on the absolute value of the mean of the random variable being sampled, terminates when $(1 + \epsilon)\text{LB} < (1 - \epsilon)\text{UB}$, and returns the mean estimate $\hat{\mu} = \text{sign}(\hat{X}_t) \frac{(1+\epsilon)\text{LB} + (1-\epsilon)\text{UB}}{2}$. Mnih et al. (2008) proved that the output satisfies (1.11) and that the expected stopping time of the policy is

$$O\left(\left(\log\left(\frac{1}{\delta}\right) + \log\left(\log\frac{3}{\epsilon|\mu|}\right)\right) \max\left(\frac{\sigma^2}{\epsilon^2\mu^2}, \frac{b-a}{\epsilon|\mu|}\right)\right).$$

Up to the $\log \log$ term, this is optimal from the work of Dagum et al. (2000).

1.3.3.2 For multiple options

Let us go back to the case where we consider $K > 1$ options. A natural variant of the best option identification problems addressed in Sections 1.3.1 and 1.3.2 is to find, with high probability, a near optimal option while not sampling for a too long time. Precisely, the learner wants to stop sampling

Parameters: $q > 0$, $t_1 \geq 1$ and $\alpha > 1$ defining the geometric grid $t_k = \lceil \alpha t_{k-1} \rceil$. (Good default choice: $q = 0.1$, $t_1 = 20$ and $\alpha = 1.1$.)

Initialization:
 $c = \frac{3}{\delta t_1^q (1 - \alpha^{-q})}$
 $\text{LB} \leftarrow 0$
 $\text{UB} \leftarrow \infty$

For $t = 1, \dots, t_1 - 1$,
 sample X_t from ν
End For

For $k = 1, 2, \dots$,
 For $t = t_k, \dots, t_{k+1} - 1$,
 sample X_t from ν
 compute $\ell_t = \frac{t_{k+1} - t_k}{t^2} \log(ct_k^q)$ and $c_t = \sqrt{2\ell_t V_t} + 3(b - a)\ell_t$
 $\text{LB} \leftarrow \max(\text{LB}, |\hat{X}_t| - c_t)$
 $\text{UB} \leftarrow \min(\text{UB}, |\hat{X}_t| + c_t)$
 If $(1 + \epsilon)\text{LB} < (1 - \epsilon)\text{UB}$, Then
 stop simulating X and return the mean estimate $\text{sign}(\hat{X}_t) \frac{(1 + \epsilon)\text{LB} + (1 - \epsilon)\text{UB}}{2}$
 End If
 End For
End For

Figure 1.5: Geometric empirical Bernstein stopping rule.

as soon as he can say that with probability at least $1 - \delta$, he has identified an option i with $\mu_i \geq \max_{1 \leq j \leq K} \mu_j - \epsilon$. An algorithm solving this problem will be called a (ϵ, δ) -correct policy. A simple way to get such a policy is to adapt the Hoeffding or Bernstein races (Figures 1.3 and 1.4) by adding an ϵ in the right-hand side of the inequality defining the removal step. It can be easily shown that this strategy is (ϵ, δ) -correct and has an expected sampling time of $O\left(\frac{K}{\epsilon^2} \log\left(\frac{nK}{\delta}\right)\right)$. This is minimax optimal up to the $\log(nK)$ term in view of the following lower bound due to Mannor and Tsitsiklis (2004).

Theorem 1.8 (Additive optimal sampling lower bound). *There exists positive constants c_1, c_2 such that for any $K \geq 2$, $0 < \delta < 1/250$, $0 < \epsilon < 1/8$, and any (ϵ, δ) -correct policy, there exists distributions ν_1, \dots, ν_K on $[0, 1]$ such that the average stopping time of the policy is greater than $c_1 \frac{K}{\epsilon^2} \log\left(\frac{c_2}{\delta}\right)$.*

Even-Dar et al. (2006) proposes a policy called median elimination and detailed in Figure 1.6 with a sampling complexity matching the previous lower bound according to the following sampling complexity result.

Theorem 1.9 (Median elimination). *The median elimination algorithm is (ϵ, δ) -correct and stops after at most $O\left(\frac{K}{\epsilon^2} \log\left(\frac{2}{\delta}\right)\right)$.*

Parameters: $\epsilon > 0, \delta > 0$.

Let $A = \{1, \dots, K\}$, $\tilde{\epsilon} = \epsilon/4$ and $\tilde{\delta} = \delta/2$.

While $|A| > 1$

- (1) sample $\lfloor \frac{4}{\tilde{\epsilon}^2} \log(3/\tilde{\delta}) \rfloor$ times every option in A .
- (2) remove from A suboptimal options:

$$A \leftarrow A \setminus \{j \in A : \hat{X}_{j,t} \text{ is smaller than the median of } (\hat{X}_{i,t})_{i \in A}\},$$
- (3) $\tilde{\epsilon} \leftarrow \frac{3}{4}\tilde{\epsilon}$ and $\tilde{\delta} \leftarrow \frac{1}{2}\tilde{\delta}$.

Output the unique element of A .

Figure 1.6: Median elimination

1.4 Online optimization

In this section we consider a different setting than the one presented in Section 1.3. Here we make the assumption that the result of an evaluation is associated to a reward, and the objective is to maximize the sum of obtained rewards. This notion induces an explicit trade-off between exploration and exploitation: At each time step the strategy has to balance between trying to obtain more information about the options, and selecting the option which seems to yield (in expectation) the highest rewards. As we shall see in Section 1.4.1, good strategies perform both exploration and exploitation at the same time.

This framework is known as the multi-armed bandit problem. It was introduced in Robbins (1952). In the last decade there has been a flurry of activity around this type of problems, with many different extensions. In this section we concentrate on the basic version where there is a finite number of options, as well as on the extension to an arbitrary set of options with a Lipschitz assumption on the mapping from options to expected rewards. A more extensive review of the existing literature (as well as the proofs of the results of Section 1.4.1) can be found in Chapter 2 of Bubeck (2010).

1.4.1 Discrete Case

We propose three different strategies for the case of a finite number of options. We describe these algorithms in Figure 1.7. They are all based on

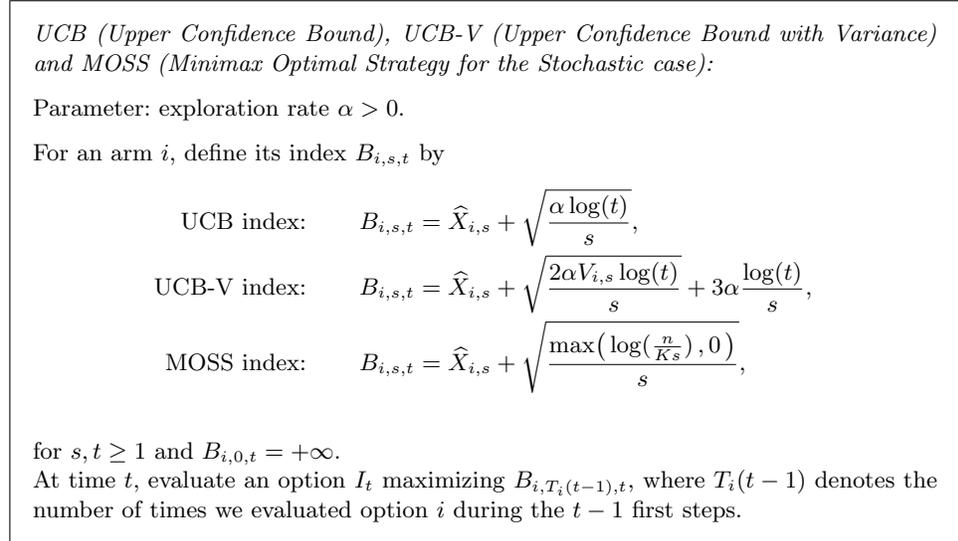


Figure 1.7: Upper confidence bound based policies.

the same underlying principle, namely the optimism in face of uncertainty. More precisely these methods assign an upper confidence bound on the mean reward of each option (which holds with high probability), and then select the option with the highest bound.

We now review the theoretical performances of the proposed strategies, and briefly discuss the implication of the different results. In particular, as we shall see, none of these strategies is uniformly (over all possible K -tuple of distributions) better (in the sense that it would have a larger expected sum of rewards) than the others.

To assess a strategy we use the expected cumulative regret defined as:

$$R_n = n \max_{1 \leq i \leq K} \mu_i - \sum_{t=1}^n \mathbb{E} \mu_{I_t}.$$

That is, R_n represents the difference in expected reward between the optimal strategy (which always selects the best option) and the strategy we used.

UCB, (Auer et al., 2002). This strategy relies on the basic Hoeffding's inequality (1.3) to build the upper confidence bound. This leads to a simple and natural algorithm, yet almost optimal. More precisely, the distribution dependent upper bound (1.12) has the optimal logarithmic rate in n , but not the optimal distribution dependent constant (see Theorem 1.13 below for the corresponding lower bound). On the other hand, the distribution free

upper bound (1.13) is optimal up to a logarithmic term (see Theorem 1.14 below for the corresponding lower bound). The two other strategies, UCB-V and MOSS, are designed to improve on these weaknesses.

Theorem 1.10 (Upper Confidence Bound algorithm). *UCB with $\alpha > 1/2$ satisfies:*

$$R_n \leq \sum_{i:\Delta_i>0} \frac{4\alpha}{\Delta_i} \log(n) + \Delta_i \left(1 + \frac{4}{\log(\alpha + 1/2)} \left(\frac{\alpha + 1/2}{\alpha - 1/2} \right)^2 \right), \quad (1.12)$$

and

$$R_n \leq \sqrt{nK \left(4\alpha \log n + 1 + \frac{4}{\log(\alpha + 1/2)} \left(\frac{\alpha + 1/2}{\alpha - 1/2} \right)^2 \right)}. \quad (1.13)$$

UCB-V, (Audibert et al., 2009). Here the confidence intervals are derived from an empirical version of Bernstein's inequality (see Theorem 1.3). This leads to an improvement in the distribution dependent rate, where basically one can replace the range of the distributions by their variances.

Theorem 1.11 (Upper Confidence Bound with Variance algorithm). *UCB-V with $\alpha > 1$ satisfies:*¹

$$R_n \leq 8\alpha \sum_{i:\Delta_i>0} \left(\frac{\sigma_i^2}{\Delta_i} + 2 \right) \log(n) + \Delta_i \left(2 + \frac{12}{\log(\alpha + 1)} \left(\frac{\alpha + 1}{\alpha - 1} \right)^2 \right). \quad (1.14)$$

MOSS, (Audibert and Bubeck, 2009). In this second modification of UCB, one combines the Hoeffding's type confidence intervals by using a tight peeling device. This leads to a minimax strategy, in the sense that the distribution free upper bound (1.16) is optimal up to a numerical constant. On the other hand, the distribution dependent bound (1.15) can be slightly worse than the one for UCB. Note also that on the contrary to UCB and UCB-V, MOSS needs to know in advance the number of evaluations. Again, one can overcome this difficulty with the doubling trick.

Theorem 1.12 (Minimax Optimal Strategy for the Stochastic case). *MOSS*

1. In the context of UCB-V it is interesting to see the influence of the range of the distributions. Precisely, if the support of all distributions ν_i are included in $[0, b]$, and if one uses the upper confidence bound sequence $B_{i,s,t} = \hat{X}_{i,s} + \sqrt{2\alpha V_{i,s} \log(t)/s} + 3b\alpha \frac{\log(t)}{s}$, then one can easily prove that the leading constant in the bound becomes $\frac{\sigma_i^2}{\Delta_i} + 2b$, which can be much smaller than the b^2/Δ_i factor characterizing the regret bound of UCB.

satisfies

$$R_n \leq \frac{23K}{\Delta} \log \left(\max \left(\frac{110n\Delta^2}{K}, 10^4 \right) \right), \quad (1.15)$$

and

$$R_n \leq 25\sqrt{nK}. \quad (1.16)$$

Lower bounds, (Lai and Robbins, 1985; Auer et al., 2003). For sake of completeness, we state here the two main lower bounds for multi-armed bandits. In Theorem 1.13, we use the Kullback-Leibler divergence between two Bernoulli distributions of parameters $p, q \in (0, 1)$, defined as:

$$\text{KL}(p, q) = p \log \left(\frac{p}{q} \right) + (1 - p) \log \left(\frac{1 - p}{1 - q} \right).$$

A useful inequality to compare the lower bound of Theorem 1.13 with (1.12) and (1.14) is the following:

$$2(p - q)^2 \leq \text{KL}(p, q) \leq \frac{(p - q)^2}{q(1 - q)}.$$

Theorem 1.13 (Distribution-dependent lower bound). *Let us consider a strategy such that for any set of K distributions, any arm i such that $\Delta_i > 0$ and any $a > 0$, we have $\mathbb{E}T_i(n) = o(n^a)$. Then if ν_1, \dots, ν_K are Bernoulli distributions, all different from a Dirac distribution at 1, the following holds true:*

$$\liminf_{n \rightarrow +\infty} \frac{R_n}{\log n} \geq \sum_{i: \Delta_i > 0} \frac{\Delta_i}{\text{KL}(\mu_i, \max_{1 \leq j \leq K} \mu_j)}. \quad (1.17)$$

An extension of the previous theorem can be found in Burnetas and Katehakis (1996).

Theorem 1.14 (Distribution-free lower bound). *Let \sup represents the supremum taken over all set of K distributions on $[0, 1]$ and \inf the infimum taken over all strategies, then the following holds true:*

$$\inf \sup R_n \geq \frac{1}{20} \sqrt{nK}. \quad (1.18)$$

1.4.2 Continuous Case

In many natural examples, the number of options is extremely large, potentially infinite. One particularly important and ubiquitous case is when the set of options is identified by a finite number of continuous-valued pa-

rameters. Unfortunately this type of problems can be arbitrarily difficult without further assumptions. One standard way to constraint the problem is to make a smoothness assumption on the mapping from options to expected reward (the mean-payoff function). In this section we present the approach proposed in (Bubeck et al., 2008), where there is essentially a weak compactness assumption on the set of options, and a weak Lipschitz assumption on the mean-payoff. We make these assumptions more precise in Section 1.4.3. Then Section 1.4.4 details the algorithm called HOO (Hierarchical Optimistic Optimization), which is based on the recent successful tree-optimization algorithms, (Kocsis and Szepesvari, 2006; Coquelin and Munos, 2007). Finally, Section 1.4.5 provides the theoretical guarantees that one can derive for HOO. The latter can be informally summed up as follows: if one knows the local smoothness of the mean-payoff function around its maximum, then with n evaluations it is possible to find an option which is (of order of) $1/\sqrt{n}$ -optimal (no matter what is the ambient dimension).

1.4.3 Assumptions, notation

Let \mathcal{X} denote the set of options, f the mean-payoff function and $f^* = \sup_{x \in \mathcal{X}} f(x)$ the supremum of f over \mathcal{X} . Recall that when one evaluates a point $x \in \mathcal{X}$, one receives an independent random variable in $[0, 1]$ with expectation $f(x)$. Let X_t be the t^{th} point that one chooses to evaluate.

As we said, one needs to make some restriction on the set of possible mean-payoff functions. We shall do this by resorting to some (weakly) Lipschitz condition. However, somewhat unconventionally, we shall use dissimilarity functions rather than metric distances, allowing us to deal with function classes of highly different smoothness orders in a unified manner. Formally a *dissimilarity* ℓ over \mathcal{X} is a non-negative mapping $\ell : \mathcal{X}^2 \rightarrow \mathbb{R}$ satisfying $\ell(x, x) = 0$ for all $x \in \mathcal{X}$. The weakly Lipschitz assumption on the mean-payoff requires that for all $x, y \in \mathcal{X}$,

$$f^* - f(y) \leq f^* - f(x) + \max\{f^* - f(x), \ell(x, y)\}. \quad (1.19)$$

The choice of this terminology follows from the fact that if f is 1-Lipschitz w.r.t. ℓ , i.e., for all $x, y \in \mathcal{X}$, one has $|f(x) - f(y)| \leq \ell(x, y)$, then it is also weakly Lipschitz w.r.t. ℓ . On the other hand, weak Lipschitzness is a milder requirement. It implies local (one-sided) 1-Lipschitzness at any global maximum (if one exists) x^* (i.e. such that $f(x^*) = f^*$) since in that case the criterion (1.19) rewrites to $f(x^*) - f(y) \leq \ell(x^*, y)$. In the vicinity of other option x , the constraint is milder as the option x gets worse (as $f^* - f(x)$

increases) since the condition (1.19) rewrites to

$$\forall y \in \mathcal{X}, \quad f(x) - f(y) \leq \max\{f^* - f(x), \ell(x, y)\}.$$

In fact, it is possible to relax (1.19) and require it only to hold locally at the global maximum (or the set of maxima if there are several). We refer the interested reader to (Bubeck et al., 2010) for further details.

We also make a mild assumption on the set \mathcal{X} which can be viewed as some sort of compactness w.r.t. ℓ . More precisely we assume that there exists a sequence $(\mathcal{P}_{h,i})_{h \geq 0, 1 \leq i \leq 2^h}$ of subsets of \mathcal{X} satisfying:

- $\mathcal{P}_{0,1} = \mathcal{X}$, and for all $h \geq 0, 1 \leq i \leq 2^h$, $\mathcal{P}_{h,i} = \mathcal{P}_{h+1,2i-1} \cup \mathcal{P}_{h+1,2i}$.
- There exists $\nu_1, \nu_2 > 0$ and $\rho \in (0, 1)$ such that each $\mathcal{P}_{h,i}$ is included in a ball of radius $\nu_1 \rho^h$ (w.r.t. ℓ) and contains a ball of radius $\nu_2 \rho^h$. Moreover, for a given h , the balls of radius $\nu_2 \rho^h$ are all disjoint.

Intuitively, for a given h , the sets $(\mathcal{P}_{h,i})_{1 \leq i \leq 2^h}$ represent a covering of \mathcal{X} at “scale” h .

The proposed algorithm takes this sequence of subsets and the real numbers ν_1, ρ as inputs. Moreover the sequence $(\mathcal{P}_{h,i})$ will be represented as an infinite binary tree, where the nodes are indexed by pairs of integers (h, i) , such that the nodes $(h+1, 2i-1)$ and $(h+1, 2i)$ are the children of (h, i) . The subset $\mathcal{P}_{h,i}$ is associated with node (h, i) .

1.4.4 The Hierarchical Optimistic Optimization (HOO) strategy

The HOO strategy (cf. Algorithm 1.1) incrementally builds an estimate of the mean-payoff function f over \mathcal{X} . The core idea is to estimate f precisely around its maxima, while estimating it loosely in other parts of the space \mathcal{X} . To implement this idea, HOO maintains the binary tree described in Section 1.4.3, whose nodes are associated with subsets of \mathcal{X} such that the regions associated with nodes deeper in the tree (further away from the root) represent increasingly smaller subsets of \mathcal{X} . The tree is built in an incremental manner. At each node of the tree, HOO stores some statistics based on the information received in previous evaluations. In particular, HOO keeps track of the number of times a node was traversed up to round n and the corresponding empirical average of the rewards received so far. Based on these, HOO assigns an optimistic estimate (denoted by B) to the maximum mean-payoff associated with each node. These estimates are then used to select the next node to “play”. This is done by traversing the tree, beginning from the root, and always following the node with the highest B -value (cf. lines 4–14 of Algorithm 1.1). Once a node is selected, a point in

the region associated with it is chosen (line 16) and is evaluated. Based on the point selected and the received reward, the tree is updated (lines 18–33).

Note that the total running time up to the n^{th} evaluation is quadratic in n . However it is possible to slightly modify the algorithm to obtain a running time of order $O(n \log n)$. The details can be found in (Bubeck et al., 2010).

1.4.5 Regret bound for HOO

In this section, we show that the regret of HOO depends on how fast the volumes of the set \mathcal{X}_ϵ of ϵ -optimal options shrink as $\epsilon \rightarrow 0$. We formalize this notion with the near-optimality dimension of the mean-payoff function. We start by recalling the definition of packing numbers.

Definition 1.15 (Packing number). *The ϵ -packing number $\mathcal{N}(\mathcal{X}, \ell, \epsilon)$ of \mathcal{X} w.r.t. the dissimilarity ℓ is the largest integer k such that there exists k disjoint ℓ -open balls with radius ϵ contained in \mathcal{X} .*

We now define the c -near-optimality dimension, which characterizes the size of the sets $\mathcal{X}_{c\epsilon}$ as a function of ϵ . It can be seen as some growth rate in ϵ of the metric entropy (measured in terms of ℓ and with packing numbers rather than covering numbers) of the set of $c\epsilon$ -optimal options.

Definition 1.16 (Near-optimality dimension). *For $c > 0$, the c -near-optimality dimension of f w.r.t. ℓ equals*

$$\max \left\{ 0, \limsup_{\epsilon \rightarrow 0} \frac{\log \mathcal{N}(\mathcal{X}_{c\epsilon}, \ell, \epsilon)}{\log(\epsilon^{-1})} \right\}.$$

Theorem 1.17 (Hierarchical Optimistic Optimization). *Let d be the $4\nu_1/\nu_2$ -near-optimality dimension of the mean-payoff function f w.r.t. ℓ . Then, for all $d' > d$, there exists a constant γ such that for all $n \geq 1$, HOO satisfies*

$$R_n = nf^* - \mathbb{E} \sum_{t=1}^n f(X_t) \leq \gamma n^{(d'+1)/(d'+2)} (\log n)^{1/(d'+2)}.$$

To put in perspective this result, we present the following example. Equip $\mathcal{X} = [0, 1]^D$ with a norm $\|\cdot\|$ and assume that the mean-payoff function f satisfies the Hölder-type property at any global maximum x^* of f (these maxima being in addition assumed to be in finite number):

$$f(x^*) - f(x) = \Theta(\|x - x^*\|^\alpha) \quad \text{as } x \rightarrow x^*,$$

Algorithm 1.1 The HOO strategy

Parameters: Two real numbers $\nu_1 > 0$ and $\rho \in (0, 1)$, a sequence $(\mathcal{P}_{h,i})_{h \geq 0, 1 \leq i \leq 2^h}$ of subsets of \mathcal{X} .

Auxiliary function $\text{LEAF}(\mathcal{T})$: outputs a leaf of \mathcal{T} .

Initialization: $\mathcal{T} = \{(0, 1)\}$ and $B_{1,2} = B_{2,2} = +\infty$.

```

1: for  $n = 1, 2, \dots$  do    % Strategy HOO in round  $n \geq 1$ 
2:    $(h, i) \leftarrow (0, 1)$     % Start at the root
3:    $P \leftarrow \{(h, i)\}$     %  $P$  stores the path traversed in the tree
4:   while  $(h, i) \in \mathcal{T}$  do    % Search the tree  $\mathcal{T}$ 
5:     if  $B_{h+1,2i-1} > B_{h+1,2i}$  then    % Select the "more promising" child
6:        $(h, i) \leftarrow (h + 1, 2i - 1)$ 
7:     else if  $B_{h+1,2i-1} < B_{h+1,2i}$  then
8:        $(h, i) \leftarrow (h + 1, 2i)$ 
9:     else    % Tie-breaking rule
10:       $Z \sim \text{Ber}(0.5)$     % e.g., choose a child at random
11:       $(h, i) \leftarrow (h + 1, 2i - Z)$ 
12:    end if
13:     $P \leftarrow P \cup \{(h, i)\}$ 
14:  end while
15:   $(H, I) \leftarrow (h, i)$     % The selected node
16:  Choose option  $x$  in  $\mathcal{P}_{H,I}$  and evaluate it    % Arbitrary selection of an option
17:  Receive corresponding reward  $Y$ 
18:   $\mathcal{T} \leftarrow \mathcal{T} \cup \{(H, I)\}$     % Extend the tree
19:  for all  $(h, i) \in P$  do    % Update the statistics  $T$  and  $\hat{\mu}$  stored in the path
20:     $T_{h,i} \leftarrow T_{h,i} + 1$     % Increment the counter of node  $(h, i)$ 
21:     $\hat{\mu}_{h,i} \leftarrow (1 - 1/T_{h,i})\hat{\mu}_{h,i} + Y/T_{h,i}$     % Update the mean  $\hat{\mu}_{h,i}$  of node  $(h, i)$ 
22:  end for
23:  for all  $(h, i) \in \mathcal{T}$  do    % Update the statistics  $U$  stored in the tree
24:     $U_{h,i} \leftarrow \hat{\mu}_{h,i} + \sqrt{(2 \log n)/T_{h,i}} + \nu_1 \rho^h$     % Update the  $U$ -value of node  $(h, i)$ 
25:  end for
26:   $B_{H+1,2I-1} \leftarrow +\infty$     %  $B$ -values of the children of the new leaf
27:   $B_{H+1,2I} \leftarrow +\infty$ 
28:   $\mathcal{T}' \leftarrow \mathcal{T}$     % Local copy of the current tree  $\mathcal{T}$ 
29:  while  $\mathcal{T}' \neq \{(0, 1)\}$  do    % Backward computation of the  $B$ -values
30:     $(h, i) \leftarrow \text{LEAF}(\mathcal{T}')$     % Take any remaining leaf
31:     $B_{h,i} \leftarrow \min\{U_{h,i}, \max\{B_{h+1,2i-1}, B_{h+1,2i}\}\}$     % Backward computation
32:     $\mathcal{T}' \leftarrow \mathcal{T}' \setminus \{(h, i)\}$     % Drop updated leaf  $(h, i)$ 
33:  end while
34: end for

```

for some smoothness order $\alpha \in [0, \infty)$. This means that there exist $c_1, c_2, \delta > 0$ such that for all x satisfying $\|x - x^*\| \leq \delta$,

$$c_2 \|x - x^*\|^\alpha \leq f(x^*) - f(x) \leq c_1 \|x - x^*\|^\alpha.$$

In particular, one can check that f is locally weakly Lipschitz for the dissimilarity defined by $\ell_{c,\beta}(x, y) = c\|x - y\|^\beta$, where $\beta \leq \alpha$ (and $c \geq c_1$ when $\beta = \alpha$) (see (Bubeck et al., 2010) for a precise definition of locally weakly Lipschitz). We further assume that HOO is run with parameters ν_1 and ρ and a tree of dyadic partitions such that the assumptions of Section 1.4.3 are satisfied. The following statements can then be formulated on the regret of HOO.

- **Known smoothness:** if we know the true smoothness of f around its maxima, then we set $\beta = \alpha$ and $c \geq c_1$. This choice $\ell_{c_1, \alpha}$ of a dissimilarity is such that f is locally weakly Lipschitz with respect to it and the near-optimality dimension is $d = 0$. Theorem 1.17 thus implies that the expected regret of HOO is $\tilde{O}(\sqrt{n})$, i.e., *the rate of the bound is independent of the dimension D* .
- **Smoothness underestimated:** here, we assume that the true smoothness of f around its maxima is unknown and that it is underestimated by choosing $\beta < \alpha$ (and some c). Then f is still locally weakly Lipschitz with respect to the dissimilarity $\ell_{c,\beta}$ and the near-optimality dimension is $d = D(1/\beta - 1/\alpha)$; the regret of HOO is $\tilde{O}(n^{(d+1)/(d+2)})$.
- **Smoothness overestimated:** now, if the true smoothness is overestimated by choosing $\beta > \alpha$ or $\alpha = \beta$ and $c < c_1$, then the assumption of weak Lipschitzness is violated and we are unable to provide any guarantee on the behavior of HOO. The latter, when used with an overestimated smoothness parameter, may lack exploration and exploit too heavily from the beginning. As a consequence, it may get stuck in some local optimum of f , missing the global one(s) for a very long time (possibly indefinitely). Such a behavior is illustrated in the example provided in Coquelin and Munos (2007) and showing the possible problematic behavior of the closely related algorithm UCT of Kocsis and Szepesvari (2006). UCT is an example of an algorithm overestimating the smoothness of the function; this is because the B -values of UCT are defined similarly to the ones of the HOO algorithm but without the additional third term in the definition of the U -values. In such cases, the corresponding B -values do not provide high probability upper-bounds on the supremum of f over the corresponding domains and the resulting algorithms do not implement the idea of “optimism in the face of uncertainty” anymore.

References

- J.-Y. Audibert. PAC-Bayesian aggregation and multi-armed bandits, 2010. Habilitation thesis.
- J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *22nd annual conference on learning theory*, 2009.
- J.-Y. Audibert, R. Munos, and Cs. Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410:1876–1902, 2009.
- J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *23rd annual Conference on Learning Theory*, 2010.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning Journal*, 47(2-3):235–256, 2002.
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.
- S. Bubeck. *Bandits Games and Clustering Foundations*. PhD thesis, Université Lille 1, 2010.
- S. Bubeck, R. Munos, G. Stoltz, and Cs. Szepesvari. Online optimization in \mathcal{X} -armed bandits. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 22*, pages 201–208, 2008.
- S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Proc. of the 20th International Conference on Algorithmic Learning Theory*, 2009.
- S. Bubeck, R. Munos, G. Stoltz, and Cs. Szepesvari. \mathcal{X} -armed bandits. Arxiv preprint 1001.4475, 2010.
- A.N. Burnetas and M.N. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.
- P.-A. Coquelin and R. Munos. Bandit algorithms for tree search. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 67–74, 2007.
- Paul Dagum, Richard Karp, Michael Luby, and Sheldon Ross. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, 2000.

- C. Domingo, R. Gavaldà, and O. Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery*, 6(2):131–152, 2002.
- J.L. Doob. *Stochastic processes*. New York, 1953.
- E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.
- D.A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, 1975.
- W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.
- L. Kocsis and Cs. Szepesvari. Bandit based Monte-carlo planning. In *Proceedings of the 15th European Conference on Machine Learning*, pages 282–293, 2006.
- T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.
- G. Lugosi. Concentration-of-measure inequalities. *Lecture notes*, 2004.
- S. Mannor and J. N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5: 623–648, 2004.
- O. Maron and A. W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems*, pages 59–66, 1993.
- P. Massart. *Concentration inequalities and model selection: Ecole d’Eté de Probabilités de Saint-Flour XXXIII-2003*. Springer, 2007.
- A. Maurer and M. Pontil. Empirical bernstein bounds and sample-variance penalization. In *22th annual Conference on Learning Theory*, 2009.
- V. Mnih, Cs. Szepesvári, and J.-Y. Audibert. Empirical bernstein stopping. In *Proceedings of the International Conference on Machine Learning*, volume 307, pages 672–679, 2008.
- L. E. Ortiz and L. P. Kaelbling. Sampling methods for action selection in influence diagrams. In *Proceedings of the national conference on Artificial Intelligence*, pages 378–385, 2000.
- H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.
- D. Williams. *Probability with martingales*. Cambridge University Press, 1991.