Information Compexity in Bandit Subset Selection
Emilie Kaufmann and Shivaram Kalyanakrishnan

In *JMLR Workshop and Conference Proceedings (Conference on Learning Theory, 2013)*, 30:228–251, 2013.

JMLR: Workshop and Conference Proceedings vol 30 (2013) 1–24

Information Complexity in Bandit Subset Selection

Emilie Kaufmann KAUFMANN@TELECOM-PARISTECH.FR
Institut Mines-Telecom; Telecom ParisTech

Shivaram Kalyanakrishnan SHIVARAM@YAHOO-INC.COM
Yahoo! Labs Bangalore

Abstract

We consider the problem of efficiently exploring the arms of a stochastic bandit to identify the best subset of a specified size. Under the PAC and the fixed-budget formulations, we derive improved bounds by using KL-divergence-based confidence intervals. Whereas the application of a similar idea in the regret setting has yielded bounds in terms of the KL-divergence between the arms, our bounds in the pure-exploration setting involve the "Chernoff information" between the arms. In addition to introducing this novel quantity to the bandits literature, we contribute a comparison between strategies based on uniform and adaptive sampling for pure-exploration problems, finding evidence in favor of the latter.

Keywords: Stochastic multi-armed bandits, subset selection, KL-divergence.

1. Introduction

We consider a stochastic bandit model with a finite number of arms $K \geq 2$. Each arm a corresponds to a Bernoulli distribution with mean p_a; the arms are numbered such that $p_1 \geq p_2 \geq ... \geq p_K$. Each draw of arm a yields a reward drawn from an unknown distribution $\mathcal{B}(p_a)$. In the classical "regret" setting, an agent seeks to sample arms sequentially in order to maximize its cumulative reward, or equivalently, to minimize its regret. This setting was originally motivated by clinical trials (Thompson, 1933) wherein the number of subjects cured is to be maximized through the judicious allocation of competing treatments. By contrast, the "pure exploration" setting models an *off-line* regime in which the rewards accrued while learning are immaterial; rather, the agent has to identify an optimal set of m arms ($1 \leq m < K$) at the end of its learning (or *exploration*) phase. Such a setting would naturally suit a company that conducts a dedicated testing phase for its products to determine which m to launch into the market. Bubeck et al. (2011, see Section 1) present an informative comparison between the regret and pure-exploration settings.

In this paper, we consider the pure-exploration problem of finding the m best arms, introduced by Kalyanakrishnan and Stone (2010) as "Explore-m". This problem, which generalizes the single-arm-selection problem studied by Even-Dar et al. (2006) (Explore-1), is as follows. For some fixed tolerance $\epsilon \in [0, 1]$, let $\mathcal{S}^*_{m,\epsilon}$ be the set of all (ϵ, m)-optimal arms: that is, the set of arms a such that $p_a \geq p_m - \epsilon$. Observe that the set of m best arms, $\mathcal{S}^*_m = \{1, 2, \ldots, m\}$, is necessarily a subset of $\mathcal{S}^*_{m,\epsilon}$. For a given mistake probability $\delta \in]0, 1]$, our goal is to design an algorithm that after using a finite (but possibly random) number of samples $\mathcal{N}$ returns $\mathcal{S}_\delta$, a set of m arms satisfying $\mathbb{P}(\mathcal{S}_\delta \subset \mathcal{S}^*_{m,\epsilon}) \geq 1 - \delta$. We desire $\mathcal{N}$ to be small in expectation.

Contrasting with the PAC formulation described above, an alternative goal in the pure-exploration setting would be to fix a maximum number of samples, n, for learning, and to find a set $\mathcal{S}_n$ of m arms after n rounds such that $e_n := \mathbb{P}(\mathcal{S}_n \not\subset \mathcal{S}^*_{m,\epsilon})$ is minimal. This setting was proposed by Audibert et al. (2010) for $m = 1$ (and $\epsilon = 0$) and generalized by Bubeck et al. (2013) to arbitrary values of m. We denote this alternative problem Explore-m-FB (Explore-m with *fixed budget*). It is interesting to note that indeed these two problems are related: Gabillon et al. (2012) point out that knowing the problem complexity allows algorithms for Explore-m to be converted to algorithms for Explore-m-FB, and vice versa.

In the regret setting, a recent line of research has yielded algorithms that are essentially optimal. While the regret bound for the UCB algorithm of Auer et al. (2002) is optimal in its logarithmic dependence on the horizon, its accompanying problem-specific constant does not match the lower bound provided by Lai and Robbins (1985). Garivier and Cappé (2011) and Maillard et al. (2011) show that by replacing UCB's Hoeffding's inequality-based bounds with upper bounds based on Kullback-Leibler divergence, the constant, too, becomes optimal (see also Cappé et al. (2013)).

The primary contribution of this paper is a set of similarly-improved bounds for the pure-exploration setting. We show improvements both for Explore-m and for Explore-m-FB by replacing Hoeffding-based bounds with KL-divergence-based bounds in corresponding algorithms. Interestingly, our analysis sheds light on potential differences between the pure-exploration and regret settings: the improved sample-complexity bounds we obtain here involve the *Chernoff information* between the arms, and not KL-divergence as in the regret setting.

Algorithms for pure-exploration broadly fall into two categories: algorithms based on *uniform sampling and eliminations* (Even-Dar et al., 2006; Heidrich-Meisner and Igel, 2009; Bubeck et al., 2013), and fully-sequential algorithms based on *adaptive sampling* (Kalyanakrishnan et al., 2012; Gabillon et al., 2012). The second contribution of this paper is a comparison between these contrasting approaches, through the study of two generic algorithms using confidence intervals, Racing and LUCB. We consider both "Hoeffding" and "KL" versions of these algorithms: in each case our theoretical and experimental results point to the superiority of the adaptive sampling heuristic.

This paper is organized as follows. We discuss the complexity of Explore-m(-FB) in Section 2, introducing Chernoff information as a relevant quantity therein. We present generic versions of the Racing and LUCB algorithms in Section 3, proceeding to describe two specific instances, KL-Racing and KL-LUCB, for which we propose a PAC guarantee and a sample-complexity analysis in Section 4. Section 5 presents corroborative results from numerical experiments.

2. Complexity measure for the Explore-m problem

While existing algorithms for Explore-m have an expected sample complexity bounded by $O(H_\epsilon \log(H_\epsilon/\delta))$, the SAR algorithm of Bubeck et al. (2013) for Explore-m-FB (with $\epsilon = 0$, which we only allow with the extra assumption that $p_m > p_{m+1}$) satisfies $\mathbb{P}(\mathcal{S}_n \not\subset \mathcal{S}^*_m) \leq$

$C\exp\left(-n/(C'\log(K)H_0)\right)$, where C and C' are some constants and

$$H_\epsilon = \sum_{a\in\{1,2,\dots K\}} \frac{1}{\max(\Delta_a^2, (\frac{\epsilon}{2})^2)}, \quad \text{with} \quad \Delta_a = \begin{cases} p_a - p_{m+1} & \text{for } a \in \mathcal{S}_m^*, \\ p_m - p_a & \text{for } a \in (\mathcal{S}_m^*)^c. \end{cases}$$

In the regret setting, the following lower bound is known from Lai and Robbins (1985). If $N_a(n)$ denotes the number of draws of arm a and R_n the regret of some algorithm up to time n, then: if $\lim_{n\to\infty} R_n = o(n^\alpha)$ for every $\alpha > 0$ and every bandit problem, then

$$\liminf_{n\to\infty} \frac{\mathbb{E}[N_a(n)]}{\log(n)} \geq \frac{1}{d(p_a, p_1)},$$

with d, the Kullback-Leibler divergence between two Bernoulli distributions, given by

$$d(x,y) = KL\left(\mathcal{B}(x), \mathcal{B}(y)\right) = x\log\left(\frac{x}{y}\right) + (1-x)\log\left(\frac{1-x}{1-y}\right).$$

While $\mathbb{E}[N_a(n)]$ is only bounded by $O(\log(n)/\Delta_a^2)$ for the UCB algorithm (Auer et al., 2002), it is indeed bounded by $\log(n)/d(p_a, p_1)$ plus some constant for KL-UCB (Cappé et al., 2013), showing that this algorithm is optimal with respect to Lai and Robbins' bound.

Note that Lai and Robbins' result holds for *any* bandit problem; however, the current lower bound of Kalyanakrishnan et al. (2012) for Explore-m is a *worst-case result* stating that for every PAC algorithm, *there exists* a bandit problem on which $\mathbb{E}[\mathcal{N}] \geq CH_\epsilon \log(m/\delta)$ (with C of order 10^{-5}). For $m = 1$, Mannor and Tsitsiklis (2004) derive lower bounds holding for any problem: their bounds involve a sum of square gaps $1/\Delta_a^2$ over a set of arms not too far from the optimal arm, but that does not necessarily contain all the arms. In the fixed-budget setting, for $m = 1$, Audibert et al. (2010) state that for every algorithm and every bandit problem with parameters in $[p, 1-p]$, there exists a permutation of the arms such that $\mathbb{P}(\mathcal{S}_n \not\subset \mathcal{S}_m^*) \geq \exp(-n/C'H_0))$ with $C' = p(1-p)/(5+o(1))$. In short, there is only a worst-case result for $m > 1$, and for $m = 1$ all the results involve squared-gaps and some constants. Thus, these existing lower bounds do not preclude algorithms from achieving upper bounds for Explore-m(-FB) in terms of quantities smaller than H, possibly involving information-theoretic terms.

In this paper, we derive upper bounds for Explore-m(-FB) in terms of Chernoff information, a quantity closely related to KL-divergence. The Chernoff information $d^*(x,y)$ between two Bernoulli distributions $\mathcal{B}(x)$ and $\mathcal{B}(y)$ is defined by

$$d^*(x,y) = d(z^*, x) = d(z^*, y) \quad \text{where } z^* \text{ is the unique } z \text{ such that } \quad d(z,x) = d(z,y).$$

Chernoff information is a relevant quantity in testing problems (Cover and Thomas, 2006). Let $X_1, X_2, \dots, X_n$ be n i.i.d. samples and $H_1 : X_i \sim \mathcal{B}(x)$ versus $H_2 : X_i \sim \mathcal{B}(y)$ be two alternative hypotheses. For a test ϕ, let $\alpha_n(\phi) = \mathbb{P}_1(\phi = 2)$ and $\beta_n = \mathbb{P}_2(\phi = 1)$ be respectively the type I and type II error. Chernoff's Theorem states that when the objective is to minimize both type I and type II error, the best achievable exponent is

$$d^*(x,y) = \lim_{n\to\infty} -\frac{1}{n}\log\min_\phi \max(\alpha_n(\phi), \beta_n(\phi)).$$

Hence, for small δ, $\frac{1}{d^*(p_a,p_m)}\log(\frac{1}{\delta})$ (resp. $\frac{1}{d^*(p_a,p_{m+1})}\log(\frac{1}{\delta})$) represents the minimal number of samples needed to discriminate between arm a and arm m (resp. arm a and arm $m+1$) with an error probability smaller than δ, in the simpler case where the parameters p_a and p_m are known. This reasoning motivates our first conjecture of a complexity term:

$$H_\epsilon^{\text{target}} := \sum_{a\in\mathcal{S}_m^*} \frac{1}{\max(d^*(p_a,p_{m+1}),\frac{\epsilon^2}{2})} + \sum_{a\in(S_m^*)^c} \frac{1}{\max(d^*(p_a,p_m),\frac{\epsilon^2}{2})}. \tag{1}$$

The complexity term H_ϵ^* we derive in Section 4 is a tight upper bound on $H_\epsilon^{\text{target}}$.

3. Two classes of algorithms based on Confidence Intervals

To define an algorithm for Explore-m, one needs to specify a *sampling strategy* (choosing which arms to draw at each round of the algorithm), a *stopping strategy* (when to stop) and a *recommendation strategy* (choosing a subset of arms in the end). Virtually all the algorithms proposed to date for pure-exploration problems can be classified according to their sampling strategy: *uniform sampling algorithms* maintain a set of remaining arms, and sample all these remaining arms at each round, whereas *adaptive sampling algorithms* sample at each round one or two well-chosen arms.

Just as upper confidence bounds have been used successfully in the regret setting, most existing algorithms for Explore-m have used both upper and lower confidence bounds on the means of the arms. We state here a generic version of a uniform sampling algorithm, Racing, and a generic version of an adaptive sampling algorithm, LUCB. To describe these contrasting heuristics, we use generic confidence intervals, denoted by $\mathcal{I}_a(t) = [L_a(t), U_a(t)]$, where t is the round of the algorithm, $L_a(t)$ and $U_a(t)$ are the lower and upper confidence bounds on the mean of arm a. Let $N_a(t)$ denote the number of draws, and $S_a(t)$ the sum of the rewards gathered from arm a up to time t. Let $\hat{p}_a(t) = \frac{S_a(t)}{N_a(t)}$ be the corresponding empirical mean reward, and let $\hat{p}_{a,u}$ be the empirical mean of u i.i.d. rewards from arm a. Additionally, let $J(t)$ be the set of m arms with the highest empirical means at time t (for the Racing algorithm, $J(t)$ only includes $m' \leq m$ arms if $m - m'$ have already been selected). Also, l_t and u_t are two 'critical' arms from $J(t)$ and $J(t)^c$ that are likely to be misclassified:

$$u_t = \underset{j\notin J(t)}{\text{argmax}}\ U_j(t) \quad \text{and} \quad l_t = \underset{j\in J(t)}{\text{argmin}}\ L_j(t). \tag{2}$$

The Racing algorithm The idea of Racing dates back to Maron and Moore (1997), who introduced it in the context of model selection for finding the (single) best model. The idea of using both accepts and rejects was then used by Heidrich-Meisner and Igel (2009) in a setting like Explore-m, applied within the context of reinforcement learning. These authors do not formally analyze the algorithm's sample complexity, as we do here. The Racing algorithm, stated precisely as Algorithm 1, samples at each round t all the remaining arms, and updates the confidence bounds. Then a decision is made to possibly select the empirical best arm if its lower confidence bound (LCB) is larger than the upper confidence bounds (UCBs) of all arms in $J(t)^c$, or to discard the empirical worst arm if its UCB is smaller than the LCBs of all arms in $J(t)$. The successive elimination algorithm (Even-Dar et al., 2006) for Explore-1 is a specification of Algorithm 1 using Hoeffding bounds.

Algorithm 1 Racing algorithm

Require: $\epsilon \geq 0$ (tolerance level), U, L (confidence bounds)
$\mathcal{R} = \{1, ..., K\}$ set of remaining arms. $S = \emptyset$ set of selected arms.
$\mathcal{D} = \emptyset$ set of discarded arms. $t = 1$ (current round of the algorithm)
while $|\mathcal{S}| < m$ and $|\mathcal{D}| < K - m$ **do**
Sample all the arms in $\mathcal{R}$ update confidence intervals
Compute $J(t)$ the set of empirical $m - |\mathcal{S}|$ best arms and $J(t)^c = \mathcal{R} \backslash J(t)$
Compute u_t and l_t according to (2)
Compute a_B (resp. a_W) the empirical best (resp. worst) arm in $\mathcal{R}$
if $(U_{u_t}(t) - L_{a_B}(t) < \epsilon) \cup (U_{a_W}(t) - L_{l_t}(t) < \epsilon)$ **then**
$a = \underset{\{a_B, a_W\}}{\text{argmax}} \left((U_{u_t}(t) - L_{a_B}(t)) \mathbb{1}_{U_{u_t}(t) - L_{a_B}(t) < \epsilon}; (U_{a_W}(t) - L_{l_t}(t)) \mathbb{1}_{U_{a_W}(t) - L_{l_t}(t) < \epsilon} \right)$
Remove arm a: $\mathcal{R} = \mathcal{R} \backslash \{a\}$
If $a = a_B$ select a: $\mathcal{S} = \mathcal{S} \cup \{a\}$, else discard a: $\mathcal{D} = \mathcal{D} \cup \{a\}$
end if
t=t+1
end while
return $\mathcal{S}$ if $|\mathcal{S}| = m$, $\mathcal{S} \cup \mathcal{R}$ otherwise

The LUCB algorithm A general version of the LUCB algorithm proposed by Kalyanakrishnan et al. (2012) is stated in Algorithm 2, using generic confidence bounds U and L, while the original LUCB uses Hoeffding confidence regions. Unlike Racing, this algorithm does not sample the arms uniformly; rather, it draws at each round the two critical arms u_t and l_t. This *sampling strategy* is associated with the natural *stopping criterion* $(B(t) < \epsilon)$ where $B(t) := U_{u_t}(t) - L_{l_t}(t)$. The UGapEc algorithm of Gabillon et al. (2012) is also an adaptive sampling algorithm, that is very close to LUCB: it uses an alternative definition of $J(t)$ using confidence bounds on the simple regret, and a correspondingly different stopping criterion $B(t)$. But as LUCB, it also samples the corresponding critical arms u_t or l_t.

KL-Racing and KL-LUCB The two algorithms mentioned above both use generic upper and lower confidence bounds on the mean of each arm, and one has the intuition that the

Algorithm 2 LUCB algorithm

Require: $\epsilon \geq 0$ (tolerance level), U, L (confidence bounds)
$t = 1$ (number of stage of the algorithm), $B(1) = \infty$ (stopping index)
for a=1...K **do**
Sample arm a, compute confidence bounds $U_a(1)$, $L_a(1)$
end for
while $B(t) > \epsilon$ **do**
Draw arm u_t and l_t. $t = t + 1$.
Update confidence bounds, set $J(t)$ and arms u_t, l_t
$B(t) = U_{u_t}(t) - L_{l_t}(t)$
end while
return $J(t)$.

smaller these confidence regions are, the smaller the sample complexity of these algorithms will be. Most of the previous algorithms use Hoeffding bounds, of the form

$$U_a(t) = \hat{p}_a(t) + \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}} \quad \text{and} \quad L_a(t) = \hat{p}_a(t) - \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}} \tag{3}$$

for some *exploration rate* $\beta(t,\delta)$. Previous work (Mnih et al., 2008; Heidrich-Meisner and Igel, 2009; Gabillon et al., 2012) has also considered the use of empirical Bernstein bounds, that can be tighter. In this paper, we introduce the use of confidence regions based on KL-divergence for Explore-m, inspired by recent improvements in the regret setting (Cappé et al., 2013). We define, for some exploration rate $\beta(t,\delta)$,

$$u_a(t) := \max\left\{q \in [\hat{p}_a(t), 1] : N_a(t) d(\hat{p}_a(t), q) \leq \beta(t,\delta)\right\}, \text{ and} \tag{4}$$

$$l_a(t) := \min\left\{q \in [0, \hat{p}_a(t)] : N_a(t) d(\hat{p}_a(t), q) \leq \beta(t,\delta)\right\}. \tag{5}$$

Pinsker's inequality ($d(x,y) \geq 2(x-y)^2$) shows that KL-confidence regions are always smaller than those obtained with Hoeffding bounds, while they share the same coverage probability (see Lemma 4 in Appendix A):

$$\hat{p}_a(t) - \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}} \leq l_a(t) \quad \text{and} \quad u_a(t) \leq \hat{p}_a(t) + \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}}. \tag{6}$$

We define, for a given function β, the **KL-Racing** and **KL-LUCB** algorithms with exploration rate β as the instances of Racing and LUCB, respectively, that use $u_a(t)$ and $l_a(t)$ as confidence bounds. Section 4 provides conditions on β for both algorithms to be PAC and sample complexity bounds under these conditions. In our theoretical and experimental analysis to follow, we address the "KL versus Hoeffding" and "uniform versus adaptive sampling" questions.

Other algorithms and fixed budget setting Apart from the Halving algorithm of Even-Dar et al. (2006) for Explore-1 (and its adaptation to Explore-m by Kalyanakrishnan and Stone (2010)) for which the upper bound scales as K/ϵ^2, Racing and LUCB capture (to the best of our knowledge) all existing algorithms for Explore-m. In the fixed-budget setting, Bubeck et al. (2013) propose the Successive Accepts and Rejects (SAR) algorithm for Explore-m-FB, generalizing the Successive Reject algorithm of Audibert et al. (2010) for Explore-1-FB. In this algorithm, arms are sampled uniformly in each of the $K-1$ phases with predetermined length, and at the end of each phase, the empirical best arm is selected or the empirical worst discarded, according to the empirical gap with $J(t)^c$ or $J(t)$ respectively (a criterion than cannot be formulated with confidence intervals). Some adaptive sampling algorithms do exist for this setting too, namely UCB-E of Audibert et al. (2010) for $m = 1$, or UGapEb of Gabillon et al. (2012). We propose here another adaptive algorithm for the fixed-budget setting, KL-LUCB-E, studied in Appendix F, derived from KL-LUCB by choosing the exploration rate β as a function of n.

4. Analysis of KL-Racing and KL-LUCB

Theorem 1, whose proof can be found in Appendix A, gives choices of β such that KL-Racing and KL-LUCB are correct with probability at least δ (δ-PAC). These choices of

β lead to the same guarantees as for their Hoeffding counterpart, (Hoeffding)-Racing and LUCB.

Theorem 1 *The (KL)-Racing algorithm using* $\beta(t,\delta) = \log\left(\frac{k_1 K t^\alpha}{\delta}\right)$, *with* $\alpha > 1$ *and* $k_1 > 1 + \frac{1}{\alpha-1}$; *and the (KL)-LUCB algorithm using* $\beta(t,\delta) = \log\left(\frac{k_1 K t^\alpha}{\delta}\right) + \log\log\left(\frac{k_1 K t^\alpha}{\delta}\right)$, *with* $\alpha > 1$ *and* $k_1 > 2e + 1 + \frac{e}{\alpha-1} + \frac{e+1}{(\alpha-1)^2}$, *are correct with probability at least* $1-\delta$.

Theorem 2 gives an upper bound on the sample complexity for KL-Racing involving Chernoff information that holds in high probability. The bound we give for KL-LUCB in Theorem 3 is smaller and holds in expectation. It involves, for $c \in [p_{m+1}, p_m]$

$$H^*_{\epsilon,c} := \sum_{a \in \{1,\dots,K\}} \frac{1}{\max(d^*(p_a,c), \epsilon^2/2)} \quad \text{and} \quad H^*_\epsilon := \min_{c \in [p_{m+1}, p_m]} H^*_{\epsilon,c}.$$

In the remainder of the paper, the parameter $c \in [p_{m+1}, p_m]$ that we introduce in our analysis will be assumed to be in $]0,1[$, excluding the case $p_m = p_{m+1} = 0$ or 1.

4.1. A concentration result involving Chernoff information

Our analysis of KL-Racing and KL-LUCB share the need to bound the probability that some constant c belongs to the interval $\mathcal{I}_a(t)$ after this arm has already been sufficiently sampled. Deriving such a result for intervals based on KL-divergence brings up Chernoff information:

Lemma 1 *Let* $T \geq 1$ *be an integer. Let* $\delta > 0$, $\gamma > 0$ *and* $c \in]0,1[$ *be such that* $p_a \neq c$.

$$\sum_{t=1}^{T} \mathbb{P}\left(a = u_t \vee a = l_t, N_a(t) > \left\lceil \frac{\gamma}{d^*(p_a,c)} \right\rceil, N_a(t) d(\hat{p}_a(t), c) \leq \gamma\right) \leq \frac{\exp(-\gamma)}{d^*(p_a,c)}.$$

Sketch of the Proof Some functions based on KL-divergence need to be defined in order to state an optimal concentration result involving KL-divergence in Lemma 2.

Definition 1 *Let* $C_1 > 1$, $(y,c) \in]0,1[^2$, $y \neq c$. *Let* $s_{C_1}(y,c)$ *be the implicit function:*

$$d(s_{C_1}(y,c), c) = \frac{d(y,c)}{C_1} \text{ and } s_{C_1}(y,c) \in \big(y,c\big),$$

where (y,c) *denotes the interval* $[y,c]$ *if* $y < c$, *and* $[c,y]$ *otherwise. We define* F_{C_1} *as:*

$$F_{C_1}(y,c) = \frac{C_1 d(s_{C_1}(y,c), y)}{d(y,c)}.$$

Lemma 2 *Let* $C_1 > 1$, $\gamma > 0$ *and* $c \in]0,1[$ *such that* $p_a \neq c$. *For any integer* T,

$$\sum_{u=\lceil \frac{C_1 \gamma}{d(p_a,c)} \rceil + 1}^{T} \mathbb{P}\left(u d(\hat{p}_{a,u}, c) \leq \gamma\right) \leq \frac{\exp\left(-F_{C_1}(p_a,c)\gamma\right)}{d\left(s_{C_1}(p_a,c), p_a\right)}. \tag{7}$$

The sum in Lemma 2 is bounded tightly in the recent analysis of KL-UCB by Cappé et al. (2013, Appendix A.2) for the value $C_1 = 1$. However, the related bound shows no exponential decay in γ, unlike the one we prove for $C_1 > 1$ in Appendix C. Whereas it was used to bound an expectation for KL-UCB, we need to bound a probability for KL-LUCB and thus need this exponential decay. This technical difference ushers in the bifurcation between Chernoff information and KL-divergence. Indeed, $F_{C_1}(p_a, c)$, that is the optimal rate in the exponential (see Appendix C), depends on the problem and to be able to later choose an exploration rate that does not, we have to choose C_1 such that $F_{C_1}(p_a, c) = 1$. As we can see below, there is a unique constant $C_1(p_a, c)$ satisfying $F_{C_1(p_a,c)}(p_a, c) = 1$ and it is related to Chernoff information:

$$\begin{aligned} F_{C_1}(p_a, c) = 1 \quad &\Leftrightarrow \quad d(s_{C_1}(p_a, c), p_a) = \frac{d(p_a, c)}{C_1} \Leftrightarrow d(s_{C_1}(p_a, c), p_a) = d(s_{C_1}(p_a, c), c) \\ &\Leftrightarrow \quad s_{C_1}(p_a, c) \text{ is the unique } z \text{ satisfying } d(z, p_a) = d(z, c). \end{aligned}$$

Hence, $C_1(p_a, c)$ can be rephrased using **Chernoff information** which is precisely defined for two Bernoulli by $d^*(p_a, c) = d(z^*, c) = d(z^*, p_a)$. One gets

$$C_1(p_a, c) = d(p_a, c)/d^*(p_a, c). \tag{8}$$

As detailed in Appendix C, invoking Lemma 2 with this particular value of C_1 leads to Lemma 1.

4.2. Sample Complexity results and discussion

We gather here our two main results on the sample complexity of KL-Racing (Theorem 2, proof in Appendix B) and KL-LUCB (Theorem 3), proved in Section 4.3.

Theorem 2 *Let $c \in [p_{m+1}, p_m]$. Let $\beta(t, \delta) = \log\left(\frac{k_1 K t^\alpha}{\delta}\right)$, with $\alpha > 1$ and $k_1 > 1 + \frac{1}{\alpha - 1}$. The number of samples $\mathcal{N}$ used in KL-Racing with $\epsilon = 0$ is such that*

$$\mathbb{P}\left(\mathcal{N} \leq \max_{a \in \{1,\dots,K\}} \frac{K}{d^*(p_a, c)} \log\left(\frac{k_1 K (H^*_{\epsilon,c})^\alpha}{\delta}\right) + K, \mathcal{S}_\delta = \mathcal{S}^*_m\right) \geq 1 - 2\delta.$$

Theorem 3 *Let $c \in [p_m, p_{m+1}]$, $\epsilon \geq 0$. Let $\beta(t, \delta) = \log\left(\frac{k_1 K t^\alpha}{\delta}\right) + \log\log\left(\frac{k_1 K t^\alpha}{\delta}\right)$ with $k_1 > 2e + 1 + \frac{e}{\alpha - 1} + \frac{e+1}{(\alpha-1)^2}$. Then with $\alpha > 1$, KL-LUCB is δ-PAC and*

$$\mathbb{P}\left(\tau \leq C_0(\alpha) H^*_{\epsilon,c} \log\left(\frac{k_1 K (H^*_{\epsilon,c})^\alpha}{\delta}\right)\right) \geq 1 - 2\delta.$$

Moreover, for $\alpha > 2$, we have the following upper bound on $\mathbb{E}[\tau]$:

$$\mathbb{E}[\tau] \leq 2 C_0(\alpha) H^*_{\epsilon,c} \log\left(\frac{k_1 K (2H^*_{\epsilon,c})^\alpha}{\delta}\right) + K_\alpha,$$

with

$$\begin{aligned} K_\alpha := \quad & \frac{\delta}{k_1 K(\alpha - 2)} + \frac{\delta 2^{\alpha-1}}{k_1(\alpha - 2)} + \frac{2^{\alpha-1}\delta}{k_1}\left(1 + \frac{1}{e} + \log 2^\alpha + \log(1 + \alpha \log 2)\right) \frac{1}{(\alpha - 2)^2}, \\ C_0(\alpha) \quad & \text{such that } C_0(\alpha) \geq \alpha \log(C_0(\alpha)) + 1 + \frac{\alpha}{e}. \end{aligned}$$

Improvements and lower bound The result of Theorem 2 is only a first bound involving Chernoff information and might be improved to involve a sum over the different arms rather than a supremum. Note that a sample complexity analysis for Hoeffding-Racing and KL-Racing can also be derived by adapting that of the successive elimination algorithm of Even-Dar et al. (2006), leading to this result, that scales in the gaps Δ_a (see Section 2):

$$\mathbb{P}\left(\mathcal{N} = \mathrm{O}\sum_{a\in\{1,2,\dots,K\}} \frac{1}{\Delta_a^2}\log\left(\frac{K}{\Delta_a\delta}\right), \mathcal{S}_\delta = \mathcal{S}_m^*\right) \geq 1-\delta. \tag{9}$$

Still, the bound in Theorem 2 can be smaller than (9) on some problems, and it does not involve some big or unspecified constant multiplied by the complexity term. Yet, this bound gives no information on the sample complexity when the algorithm makes a mistake (as Kalyanakrishnan et al. (2012) note for successive elimination); and only holds for $\epsilon = 0$.

For KL-LUCB, as Theorem 3 is true for all $c \in [p_{m+1}, p_m]$, we can show a more elegant result on the *expectation* of $\mathcal{N}$ involving the smaller quantity H_ϵ^* mentioned above, for *every* $\epsilon \geq 0$. For KL-LUCB with parameters $2 < \alpha \leq 2.2$ and $2e + 1 + \frac{e}{\alpha-1} + \frac{e+1}{(\alpha-1)^2} < k_1 \leq 13$,

$$\mathbb{E}[\mathcal{N}] \leq 24H_\epsilon^* \log\left(\frac{13(H_\epsilon^*)^{2.2}}{\delta}\right) + \frac{18\delta}{k_1(\alpha-2)^2} \quad \text{with} \quad H_\epsilon^* = \min_{c\in[p_{m+1};p_m]} H_{\epsilon,c}^*.$$

We believe that this finite-time upper bound on the expected sample complexity is the first of its kind involving KL-divergence (through Chernoff information). Pinsker's inequality shows that $d^*(x,y) \geq (x-y)^2/2$ and gives a relationship with the complexity term H_ϵ (see Section 2) derived by Kalyanakrishnan et al. (2012): $H_\epsilon^* \leq 8H_\epsilon$. Although H_ϵ^* cannot be shown to be strictly smaller than H_ϵ on every problem, the explicit bound in (4.2) still improves over that of Kalyanakrishnan et al. (2012) in terms of the hidden constants. Also, the theoretical guarantees in Theorem 3 hold for smaller exploration rates, which appear to lower the sample complexity in practice.

Observe that H_ϵ^* is larger than $H_\epsilon^{\text{target}}$, conjectured as the "true" problem complexity in (1). We believe that the parameter c is an artifact of our proof, but are currently unable to eliminate it. It is worth noting that on problems in which p_m and p_{m+1} are not too close to 0 or 1, our Chernoff information-based bound is comparable with a KL-divergence-based bound: numerical evaluations of the function $C(p_a, c)$ (see (8)) indeed show that for $p_m, p_{m+1} \in [0.001, 0.999]$,

$$H_\epsilon^{\text{target}} \leq H_\epsilon^* \leq H^*_{\epsilon,\frac{p_m+p_{m+1}}{2}} \leq 9\sum_{a\in\{1,\dots,K\}} \frac{1}{\max\left(d\left(p_a, (p_m+p_{m+1})/2\right), \frac{\epsilon^2}{2}\right)}.$$

It is an open problem to show a problem-dependent lower bound for Explore-m, and to investigate if it involves Chernoff information terms $d^*(x,y)$ or KL-divergence terms $d(x,y)$.

Generalization Although we have only considered Bernoulli distributions in this paper, note that KL-LUCB (and our proofs) can be extended to rewards in the exponential family (as shown by Garivier and Cappé (2011) for KL-UCB) by using an appropriate d function.

4.3. Proof of Theorem 3

Theorem 3 easily follows from these two inequalities that holds for any exploration rate:

$$\text{for } \alpha > 1, T \geq T_1^*, \ \mathbb{P}(\tau \geq T) \leq H^*_{\epsilon,c} e^{-\beta(T,\delta)} + \sum_{t=1}^{\infty} (\beta(t,\delta)\log(t)+1) e^{-\beta(t,\delta)} \tag{10}$$

$$\text{for } \alpha > 2, T \geq T_2^*, \ \mathbb{P}(\tau \geq T) \leq H^*_{\epsilon,c} e^{-\beta(T,\delta)} + \frac{KT}{2} (\beta(T,\delta)\log(T)+1) e^{-\beta(T/2,\delta)}, \tag{11}$$

with

$$T_1^* = \min\{T : H^*_{c,\epsilon}\beta(T,\delta) < T\} \quad \text{and} \quad T_2^* = \min\{T : 2H^*_{c,\epsilon}\beta(T,\delta) < T\},$$

and from the bound on T_1^* and T_2^* given in Appendix E:

$$T_1^* \leq C_0(\alpha) H^*_{\epsilon,c} \log\left(\frac{k_1 K (H^*_{\epsilon,c})^\alpha}{\delta}\right) \text{ and } T_2^* \leq 2C_0(\alpha) H^*_{\epsilon,c} \log\left(\frac{k_1 K (2H^*_{\epsilon,c})^\alpha}{\delta}\right),$$

with $C_0(\alpha)$ as specified in Theorem 3.

We now show (11). For $c \in [p_{m+1}, p_m]$, if the confidence intervals of arms in $J(t)$ and in $J(t)^c$ are separated by c, the algorithm necessarily has to stop. This simple idea is expressed in Proposition 1, with a proof provided in Appendix D. To state it, we need to define the event

$$W_t = \bigcap_{a \in \mathcal{S}_m^*} (u_a(t) > p_a) \bigcap_{b \in (\mathcal{S}_m^*)^c} (l_b(t) < p_b).$$

Proposition 1 *If $U_{u_t} - L_{l_t} > \epsilon$ and W_t holds, then either $k = l_t$ or $k = u_t$ satisfies*

$$c \in \mathcal{I}_k(t) \quad \textit{and} \quad \tilde{\beta}_k(t) > \frac{\epsilon}{2},$$

where we define $\tilde{\beta}_a(t) := \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}}$.

The remainder of this proof borrows from Kalyanakrishnan et al. (2012, see Lemma 5). Let T be some fixed time and τ the stopping time of the algorithm. Our goal is to find an event on which $\min(\tau, T) < T$; that is, the algorithm must have stopped after T rounds. Writing $\overline{T} = \lceil \frac{T}{2} \rceil$, we upper bound $\min(\tau, T)$:

$$\begin{aligned}
\min(\tau,T) &= \overline{T} + \sum_{t=\overline{T}}^{T} \mathbb{1}_{(U_{u_t}(t) - L_{l_t}(t) > \epsilon)} \mathbb{1}_{W_t} + \sum_{t=\overline{T}}^{T} \mathbb{1}_{W_t^c} \\
&\leq \overline{T} + \sum_{t=\overline{T}}^{T} \left[\mathbb{1}_{(c \in \mathcal{I}_{l_t}(t))} \mathbb{1}_{(\tilde{\beta}_{l_t}(t) > \frac{\epsilon}{2})} + \mathbb{1}_{(c \in \mathcal{I}_{u_t}(t))} \mathbb{1}_{(\tilde{\beta}_{u_t}(t) > \frac{\epsilon}{2})} \right] + \sum_{t=\overline{T}}^{T} \mathbb{1}_{W_t^c} \\
&\leq \overline{T} + \sum_{t=\overline{T}}^{T} \sum_{a \in \{1,2,\ldots,K\}} \mathbb{1}_{(a=l_t) \vee (a=u_t)} \mathbb{1}_{(c \in \mathcal{I}_a(t))} \mathbb{1}_{(\tilde{\beta}_a(t) > \epsilon/2)} + \sum_{t=\overline{T}}^{T} \mathbb{1}_{W_t^c},
\end{aligned} \tag{12}$$

where the first inequality comes from Proposition 1. First, one has that $\tilde{\beta}_a(t) > \epsilon/2 \Leftrightarrow N_a(t) < \beta(t,\delta)/(\epsilon^2/2)$. We then split the first sum in the RHS of (12) depending on whether arm a belongs to the set $\mathcal{A}_\epsilon = \{a \in \{1,2,\ldots,K\} : d^*(p_a, c) < \epsilon^2/2\}$.

$$\begin{aligned}\min(\tau, T) &\leq \overline{T} + \sum_{a\in\mathcal{A}_\epsilon}\sum_{t=\overline{T}}^{T} \mathbb{1}_{(a=l_t\vee u_t)}\mathbb{1}_{\left(N_a(t)<\frac{\beta(t,\delta)}{\epsilon^2/2}\right)} + \sum_{a\in\mathcal{A}_\epsilon^c}\sum_{t=\overline{T}}^{T} \mathbb{1}_{(a=l_t\vee u_t)}\mathbb{1}_{(c\in\mathcal{I}_a(t))} + \mathbb{1}_{W^c_{\overline{T},T}} \\ &\leq \overline{T} + \sum_{a\in\mathcal{A}_\epsilon} \frac{\beta(T,\delta)}{\epsilon^2/2} + \sum_{a\in\mathcal{A}_\epsilon^c}\sum_{t=\overline{T}}^{T} \mathbb{1}_{(a=l_t)\vee(a=u_t)}\mathbb{1}_{N_a(t)\leq\left\lceil\frac{\beta(T,\delta)}{d^*(p_a,c)}\right\rceil} + R_T\end{aligned}$$

where

$$R_T := \sum_{a\in\mathcal{A}_\epsilon^c}\sum_{t=\overline{T}}^{T} \mathbb{1}_{(a=l_t)\vee(a=u_t)}\mathbb{1}_{(c\in\mathcal{I}_a(t))}\mathbb{1}_{N_a(t)>\left\lceil\frac{\beta(T,\delta)}{d^*(p_a,c)}\right\rceil} + \sum_{t=\overline{T}}^{T}\mathbb{1}_{W_t^c}.$$

This yields: $\min(\tau, T) \leq \overline{T} + H^*_{\epsilon,c}\beta(T,\delta) + R_T$. Introducing $T_2^* = \min\{T : 2H^*_{c,\epsilon}\beta(T,\delta) < T\}$, we get that for $T > T_2^*$, on the event $(R_T = 0)$, $\min(\tau, T) < T$, the algorithm must have stopped before T. Hence, for such T,

$$\mathbb{P}(\tau \geq T) \leq \underbrace{\mathbb{P}\left(\exists a \in \mathcal{A}_\epsilon^c, t \leq T : a = l_t \vee u_t, N_a(t) > \frac{\beta(T,\delta)}{d^*(p_a,c)}, c \in \mathcal{I}_a(t)\right)}_{A} + \underbrace{\mathbb{P}\left(\bigcup_{\overline{T}}^{T} W_t^c\right)}_{B}.$$

We upper-bound term A using Lemma 1 (with $\gamma = \beta(T,\delta)$), writing

$$\begin{aligned}A &\leq \sum_{a\in\mathcal{A}_\epsilon^c}\sum_{t=1}^{T} \mathbb{P}\left(a = l_t \vee a = u_t, N_a(t) > \left\lceil\frac{\beta(T,\delta)}{d^*(p_a,c)}\right\rceil, c \in \mathcal{I}_a(t)\right) \\ &\leq \sum_{a\in\mathcal{A}_\epsilon^c}\sum_{t=1}^{T} \mathbb{P}\left(a = l_t \vee a = u_t, N_a(t) > \left\lceil\frac{\beta(T,\delta)}{d^*(p_a,c)}\right\rceil, N_a(t)d(\hat{p}_a(t), c) \leq \beta(T,\delta)\right) \\ &\leq \sum_{a\in\mathcal{A}_\epsilon^c} \frac{1}{d^*(p_a,c)} \exp\left(-\beta(T,\delta)\right) \leq H^*_{\epsilon,c}\exp\left(-\beta(T,\delta)\right),\end{aligned}$$

and we upper-bound term B using Lemma 4 (Appendix A):

$$B \leq K\sum_{t=\overline{T}}^{T} e\beta(t,\delta)(\log t + 1)\exp(-\beta(t,\delta)) \leq K\frac{T}{2}\beta(T,\delta)(\log T + 1)\exp\left(-\beta\left(T/2,\delta\right)\right).$$

This proves (11). The proof of (10) follows along the same lines, except we do not introduce $\overline{T}$ and replace it by zero in the above equations. The introduction of $\overline{T}$ to show (10) is necessary to be able to upper-bound $\mathbb{P}(\tau \geq T)$ by the general term of a convergent series.

5. Numerical experiments

On the basis of our theoretical analysis from the preceding sections, could we expect the "KL-ized" versions of our algorithms to perform better in practice? Does being "fully sequential" make our adaptive sampling algorithms more efficient than uniform sampling algorithms *in practice*? In this section, we present numerical experiments that answer both these questions in the affirmative.

In our experiments, in addition to (KL-)LUCB and (KL-)Racing, we include (KL-)LSC, an adaptive sampling algorithm akin to (KL)-LUCB. This algorithm uses the same stopping criterion as (KL)-LUCB, but rather than sample arms u_t and l_t at stage t, (KL)-LSC samples the **l**east-**s**ampled arm from $J(t)$ (or $J(t)^c$) that **c**ollides (overlaps by at least ϵ) with some arm in $J(t)^c$ ($J(t)$). To ensure that all algorithms are provably PAC, we run them with the following parameters: (KL-)LUCB and (KL-)LSC with $\alpha = 1.1$, $k_1 = 405.5$, and (KL-)Racing with $\alpha = 1.1$, $k_1 = 11.1$. Results are summarized in Figure 1.

As a first order of business, we consider bandit instances with $K = 10, 20, \ldots, 60$ arms; we generate 1000 random instances for each setting of K, with each arm's mean drawn uniformly at random from $[0, 1]$. We set $m = \frac{K}{5}$, $\epsilon = 0.1$, $\delta = 0.1$. The expected sample complexity of each algorithm on the bandit instances for each K are plotted in Figure 1(a). Indeed we observe for each K that (1) the KL-ized version of each algorithm enjoys a lower sample complexity, and (2) (KL)-LUCB outperforms (KL-)LSC, which outperforms (KL-)Racing. The differences in sample complexity consistently increase with K.

These trends, aggregated from multiple bandit instances, indeed hold for nearly every individual bandit instance therein. In fact, we find that KL-izing has a more pronounced effect on bandit instances with means close to 0 or 1. For illustration, consider instance B_1 ($K = 15$; $p_1 = \frac{1}{2}$; $p_a = \frac{1}{2} - \frac{a}{40}$ for $a = 2, 3, \ldots, K$), an instance used by Bubeck et al. (2013, see Experiment 5). Figure 1(b) compares the runs of LUCB and KL-LUCB both on B_1 (with $m = 3$, $\epsilon = 0.04$, $\delta = 0.1$), and a "scaled-down" version B_2 (with $m = 3$, $\epsilon = 0.02$, $\delta = 0.1$) in which each arm's mean is half that of the corresponding arm's in B_1 (and thus closer to 0). While LUCB and KL-LUCB both incur a higher sample complexity on the harder B_2, the latter's relative economy is clearly visible in the graph—an advantage that could benefit applications such as optimizing click-through rates of on-line advertisements.

How conservative are the stopping criteria of our PAC algorithms? In our third experiment, we halt these algorithms at intervals of 1000 samples, and at each stage record the probability that the set $J(t)$ of m empirical best arms that would be returned at that stage is non-optimal. Results from this experiment, again on B_1 (with $m = 3$, $\epsilon = 0.04$, $\delta = 0.1$), are plotted in Figure 1(c). Notice that (KL)-LUCB indeed drives down the mistake probability much faster than its competitors. Yet, even if all the algorithms have an empirical mistake probability smaller than δ after 5,000 samples, they only stop after at least 20,000 episodes, leaving us to conclude that our formal bounds are rather conservative. On the low-reward instance B_2 (with $m = 3$, $\epsilon = 0.02$, $\delta = 0.1$), we observe that KL-LUCB indeed reduces the mistake probability more quickly than LUCB, indicating a superior sampling strategy. This difference is between LUCB and KL-LUCB is *not* apparent on B_1 in Figure 1(c).

We test KL-LUCB-$log(t)$, a version of KL-LUCB with an exploration rate of $log(t)$ (which yields no provable guarantees) as a candidate for *Explore-m-FB*. On B_1 (with $n = 4000$), we compare this algorithm with KL-LUCB-E, discussed in Appendix F, which has a provably-optimal exploration rate involving the problem complexity ($H^*_\epsilon \approx 13659$). Quite surprisingly, we find that KL-LUCB-$log(t)$ significantly outdoes KL-LUCB-E for every setting of m from 1 to 14. KL-LUCB-$log(t)$ also outperforms the SAR algorithm of Bubeck et al. (2013), yielding yet another result in favor of adaptive sampling. A *tuned version* of KL-LUCB-E (using an exploration rate of $\frac{n}{2\times 180}$) performs virtually identical to KL-LUCB-$log(t)$, and is not shown in the figure.

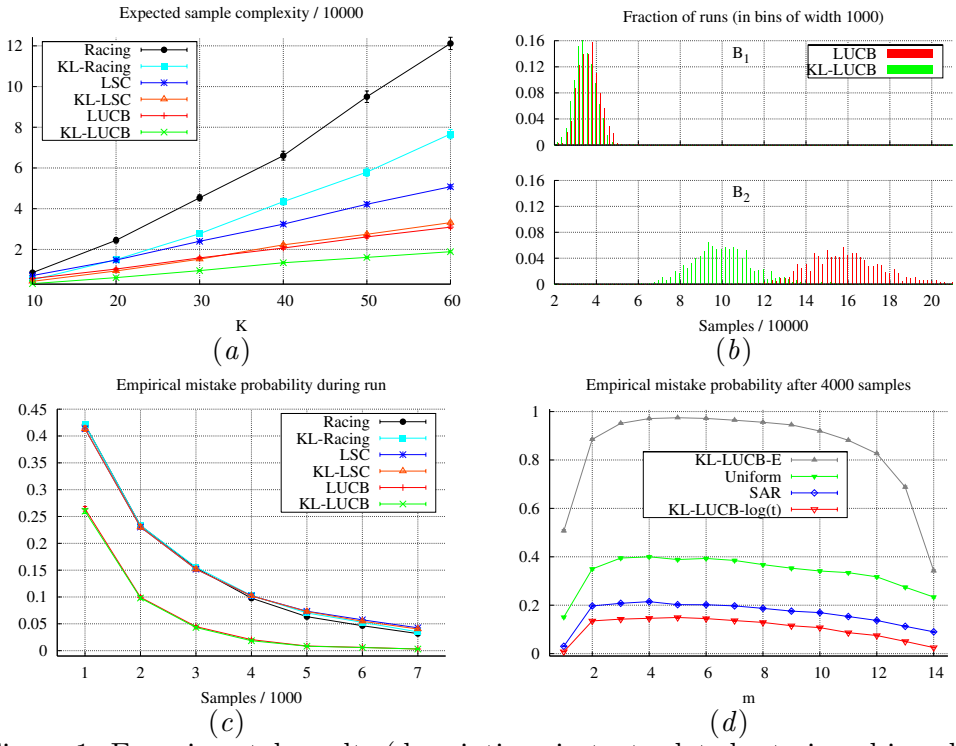

Figure 1: Experimental results (descriptions in text; plots best viewed in color).

6. Conclusion

This paper presents a successful translation of recent improvements for bandit problems in the regret setting to the pure-exploration setting. Incorporating confidence intervals based on KL-divergence into the Racing and LUCB algorithms, which capture almost every algorithm previously used for Explore-m, we introduce the KL-LUCB and KL-Racing algorithms, which improve both in theory and in practice over their Hoeffding counterparts. Our experiments also provide the novel insight that adaptive sampling might be superior to uniform sampling even for Explore-m-FB.

For KL-LUCB, we provide the first finite-time upper bound on the expected sample complexity involving Chernoff information. Is there a fundamental difference between the regret and pure-exploration settings that would justify a different complexity measure, albeit one still based on KL-divergence? A problem-dependent lower bound on the expected sample complexity of any PAC algorithm for Explore-m could answer this question, and is left as an interesting open question. As another gap between regret and pure-exploration, one might consider that no counterpart of the Thompson Sampling algorithm, recently shown to be optimal in the regret setting (Kaufmann et al., 2012) as well as practically very efficient, has yet been found for Explore-m(-FB).

Acknowledgments

We thank Aurélien Garivier and Olivier Cappé for many helpful discussions and valuable advice. We also thank anonymous reviewers whose comments have improved the paper.

References

J-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *Conference on Learning Theory (COLT)*, 2010.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely armed and continuous armed bandits. *Theoretical Computer Science 412, 1832-1852*, 412:1832–1852, 2011.

S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning (ICML)*. To appear, 2013.

O. Cappé, A. Garivier, O-A. Maillard, R. Munos, and G. Stoltz. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *to appear in Annals of Statistics*, 2013.

T. Cover and J. Thomas. *Elements of Information Theory (2nd Edition)*. Wiley, 2006.

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.

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Neural Information and Signal Processing (NIPS)*, 2012.

A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Conference on Learning Theory (COLT)*, 2011.

V. Heidrich-Meisner and C. Igel. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *International Conference on Learning Theorey (ICML)*, 2009.

S. Kalyanakrishnan. *Learning Methods for Sequential Decision Making with Imperfect Representations*. PhD thesis, Departement of Computer Science, The University of Texas at Austin, 2011.

S. Kalyanakrishnan and P. Stone. Efficient selection in multiple bandit arms: Theory and practice. In *International Conference on Machine Learning (ICML)*, 2010.

S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2012.

E. Kaufmann, N. Korda, and R. Munos. Thompson sampling : an asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory (ALT)*, 2012.

T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

O-A. Maillard, R. Munos, and G. Stoltz. A finite-time analysis of multi-armed bandits problems with Kullback-Leibler divergences. In *Conference On Learning Theory (COLT)*, 2011.

S. Mannor and J. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, pages 623–648, 2004.

O. Maron and A. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1-5):113–131, 1997.

V. Mnih, C. Szepesvári, and J-Y. Audibert. Empirical Bernstein stopping. In *International Conference on Machine Learning (ICML)*, 2008.

W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Appendix A. PAC guarantees for (KL)-Racing and (KL)-LUCB

Proof of Theorem 1 The PAC guarantees for (KL)-Racing and (KL)-LUCB follow from Lemma 3 that states that generic Racing and LUCB algorithm are correct on some event W. Then the concentration inequalities in Lemma 4 lead to a bound on $\mathbb{P}(W^c)$ for each algorithm that depend on $\beta(t,\delta)$. Finally it is easy to check that choices of $\beta(t,\delta)$ in Theorem 1 lead to $\mathbb{P}(W^c) \leq \delta$.

Lemma 3 *Racing and LUCB are correct on the event*

$$W = \bigcap_{t\in\mathbb{N}} \bigcap_{a\in\mathcal{S}_m^*} (U_a(t) > p_a) \bigcap_{b\in(\mathcal{S}_m^*)^c} (L_b(t) < p_b).$$

where U and L denote the generic confidence bounds used by these two algorithms.

Proof for Racing If Racing is not correct, there exists some first round t on which either an arm in $(\mathcal{S}_{m,\epsilon}^*)^c$ is selected (first situation), or an arm in $\mathcal{S}_m^*$ is dismissed (second situation). Before t, all the arms in the set of selected arms $\mathcal{S}$ are in $\mathcal{S}_{m,\epsilon}^*$, and all the arms in set of discarded arms $\mathcal{D}$ are in $(\mathcal{S}^*)^c$. In the first situation, let b be the arm in $(\mathcal{S}_{m,\epsilon}^*)^c$ selected : for all arms a in $J(t)^c$, one has $U_a(t) - L_b(t) < \epsilon$. Among these arms, at least one must be in $\mathcal{S}_m^*$. So there exists $a \in \mathcal{S}_m^*$ and $b \in (\mathcal{S}_\epsilon^*)^c$ such that $U_a(t) < L_b(t) + \epsilon$. The second situation leads to the same conclusion. Hence if the algorithm fails, the following event holds:

$$\begin{aligned}
&\bigcup_{t\in\mathbb{N}} (\exists a \in \mathcal{S}_m^*, \exists b \in (\mathcal{S}_{m,\epsilon}^*)^c : \; U_a(t) - L_b(t) < \epsilon) \\
&\subset \bigcup_{t\in\mathbb{N}} \left(\exists a \in \mathcal{S}_m^*, \exists b \in (\mathcal{S}_{m,\epsilon}^*)^c : \; (U_a(t) < p_a) \cup (L_b(t) > p_a - \epsilon > p_b)\right) \\
&\subset \bigcup_{t\in\mathbb{N}} \bigcup_{a\in\mathcal{S}_m^*} (U_a(t) < p_a) \bigcup_{b\in(\mathcal{S}_{m,\epsilon}^*)^c} (L_b(t) > p_b) \subset W^c.
\end{aligned}$$

Proof for LUCB If LUCB is not correct, there exists some stopping time τ, arm a in $\mathcal{S}_m^*$ and an arm b in $(\mathcal{S}_{m,\epsilon}^*)^c$ such that $a \in J(\tau)$ and $b \in J(\tau)^c$. As the stopping condition holds, on has $U_a(t) - L_b(t) < \epsilon$. Using the same reasoning as above, if the algorithm fails, the following event holds:

$$\bigcup_{t\in\mathbb{N}} \bigcup_{a\in\mathcal{S}_m^*} (U_a(t) < p_a) \bigcup_{b\in(\mathcal{S}_{m,\epsilon}^*)^c} (L_b(t) > p_b) \subset W^c.$$

□

Lemma 4 *Let $u_a(t)$ and $l_a(t)$ be the KL-bounds defined in (4) and (5). Let $U_a(t)$ and $L_a(t)$ be the Hoeffding bounds defined in (3). For any algorithm and any arm a,*

$$\mathbb{P}(u_a(t) < p_a) = \mathbb{P}(l_a(t) > p_a) \leq e(\beta(t,\delta)\log(t) + 1)\exp(-\beta(t,\delta)). \quad (13)$$

$$\mathbb{P}(U_a(t) < p_a) = \mathbb{P}(L_a(t) > p_a) \leq e(\beta(t,\delta)\log(t) + 1)\exp(-\beta(t,\delta)). \quad (14)$$

For (KL)-Racing, for an arm still in the race:

$$\mathbb{P}(u_a(t) < p_a) = \mathbb{P}(l_a(t) > p_a) \leq \exp(-\beta(t,\delta)). \quad (15)$$

$$\mathbb{P}(U_a(t) < p_a) = \mathbb{P}(L_a(t) > p_a) \leq \exp(-\beta(t,\delta)). \quad (16)$$

Proof of Lemma 4 Hoeffding's inequality (resp. Chernoff's inequality) is enough to derive (15) (resp. (16)), since for an arm still in the race, $N_a(t) = t$, and $\hat{p}_a(t) = \hat{p}_{a,t}$ (no union bound over $N_a(t)$ is needed). For a more general algorithm (including LUCB), sampling is not uniform, and the self-normalized inequality proved by Garivier and Cappé (2011, see Theorem 10) leads to the bound (13), which is tighter than what we get by applying Chernoff inequality and a union bound over $N_a(t)$. (14) can be shown using the same technique as in their proof.

□

Appendix B. Sample complexity analysis of KL-Racing

Let $c \in [p_{m+1}, p_m]$ and $T = \lceil H_{0,c} \rceil$. Let W the event defined in Lemma 3 (in Appendix A) on which the algorithm is correct, and

$$\tilde{W}_T = \bigcap_{t \leq T} \bigcap_{a \in \mathcal{A}} \left(N_a(t) > \left\lceil \frac{\beta(T,\delta)}{d^*(p_a, c)} \right\rceil \Rightarrow c \notin \mathcal{I}_a(t) \right).$$

On W, KL-Racing has only selected good arms and dismissed bad arms before time t. Let $a \in \mathcal{S}_m^*$ and t_a be the number of samples of a used by the algorithm. A sufficient condition for a to be selected at time t is that $l_a(t) > c$ and $u_b(t) < c$ for all arms $b \in (\mathcal{S}_m^*)^c$ still in the race. On $\tilde{W}_T$ this happens when $t \geq \max\left(\frac{\beta(T,\delta)}{d^*(p_a,c)}, \frac{\beta(T,\delta)}{d^*(p_{m+1},c)} \right) + 1$. Hence,

$$t_a \leq \max_a \frac{1}{d^*(p_a, c)} \beta(T, \delta) + 1.$$

The same holds for $b \in (\mathcal{S}_m^*)^c$. Therefore on $W \cap \tilde{W}_T$,

$$\mathcal{N} \leq \max_{a \in \{1,\dots,K\}} \frac{K}{d^*(p_a, c)} \log\left(\frac{k_1 K (H_{0,c}^*)^\alpha}{\delta} \right) + K.$$

From its definition, $\mathbb{P}(W^c) \leq \delta$ and

$$\begin{aligned}
\mathbb{P}(\tilde{W}_T^c) &\leq \sum_{t \leq T} \sum_{a \in \mathcal{A}} \mathbb{P}\left(N_a(t) > \left\lceil \frac{\beta(T,\delta)}{d^*(p_a,c)} \right\rceil, N_a(t) d(\hat{p}_a(t), c) \leq \beta(T,\delta) \right) \\
&\leq \sum_{a \in \mathcal{A}} \sum_{t = \left\lceil \frac{\beta(t,\delta)}{d^*(p_a,c)} \right\rceil + 1} \mathbb{P}\left(t d(\hat{p}_{a,t}, c) \leq \beta(T,\delta) \right) \\
&\overset{(7)}{\leq} \sum_{a \in \mathcal{A}} \frac{e^{-\beta(T,\delta)}}{d^*(p_a,c)} \leq \frac{H_{0,c}^*}{k_1 K T^\alpha} \delta \overset{(T \geq H_{0,c})}{\leq} \delta.
\end{aligned}$$

Appendix C. Proof of concentration Lemmas

The probability we have to bound in order to prove Lemma 2 is

$$P := \sum_{u = \lceil C_1 \gamma / d(p_a,c) \rceil + 1}^{T} \mathbb{P}\left(u d(\hat{p}_{a,u}, c) \leq \gamma \right).$$

This sum also arises in the analysis of the KL-UCB algorithm and is precisely upper-bounded by Cappé et al. (2013, see Appendix A.2), for the choice $C_1 = 1$. However, as we want to bound a probability and not an expectation, the bound in Cappé et al. (2013) is not tight enough, and we adapt the method proposed to the choice $C_1 > 1$. Introducing

$$d^+(x,c) = d(x,c)\mathbb{1}_{(x<c)} \quad \text{and} \quad d^-(x,c) = d(x,c)\mathbb{1}_{(x>c)},$$

we use:

$$\begin{aligned} P &\leq \sum_{u=n_1(a,c,\gamma)+1}^{T} \mathbb{P}\left(ud^+(\hat{p}_{a,u},c) \leq \gamma\right) \quad \text{for} \quad p_a < c, \text{ and} \\ P &\leq \sum_{u=n_1(a,c,\gamma)+1}^{T} \mathbb{P}\left(ud^-(\hat{p}_{a,u},c) \leq \gamma\right) \quad \text{for} \quad p_a > c, \end{aligned}$$

with $n_1(a,c,\gamma) = \left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil$. We now introduce notation that will be useful in the rest of the proof. The two mappings

$$\begin{array}{rccc} d^+ : & [0,c] & \longrightarrow & [0,d(0,c)] \\ & x & \mapsto & d(x,c) \end{array} \qquad \begin{array}{rccc} d^- : & [c,1] & \longrightarrow & [0,d(1,c)] \\ & x & \mapsto & d(x,c) \end{array}$$

are bijective and monotone. Then, for $\alpha \in [0, d(p_a,c)]$, the quantity $s^*_\alpha(p_a,c)$ is well-defined by:

$$d(s^*_\alpha(p_a,c),c) = \alpha \quad \text{and} \quad s^*_\alpha(p_a,c) \in (p_a,c).$$

With this new notation, one has, for $a \in (\mathcal{S}^*_m)^c$:

$$\mathbb{P}\left(ud^+(\hat{p}_{a,u},c) \leq \gamma)\right) = \mathbb{P}\left(d^+(\hat{p}_{a,u},c) \leq \frac{\gamma}{u}\right) = \mathbb{P}\left(\hat{p}_{a,u} \geq s^*_{\frac{\gamma}{u}}(p_a,c)\right).$$

And for $a \in \mathcal{S}^*_m$:

$$\mathbb{P}\left(ud^-(\hat{p}_{a,u},c) \leq \gamma)\right) = \mathbb{P}\left(\hat{p}_{a,u} \leq s^*_{\frac{\gamma}{u}}(p_a,c)\right).$$

Using Chernoff's concentration inequality and a comparison with an integral yields in both cases:

$$P \leq \sum_{u=n_1(a,c,\gamma)+1}^{T} \exp\left(-ud\left(s^*_{\frac{\gamma}{u}}(p_a,c),p_a\right)\right) \leq \int_{n_1(a,c,\gamma)}^{\infty} \exp\left(-ud\left(s^*_{\frac{\gamma}{u}}(p_a,c),p_a\right)\right) du.$$

With the change of variable $u = \gamma v$, one has:

$$P \leq \underbrace{\gamma \int_{\frac{C_1}{d(p_a,c)}}^{\infty} \exp\left(-\gamma v d\left(s^*_{\frac{1}{v}}(p_a,c),p_a\right)\right) dv}_{m_\gamma}. \tag{17}$$

An asymptotic equivalent This last integral takes the form

$$\int_{\frac{C_1}{d(p_a,c)}}^{\infty} \exp\left(-\gamma\phi(v)\right) \quad \text{with} \quad \phi(v) = vd\left(s^*_{\frac{1}{v}}(p_a,c), p_a\right)$$

and ϕ is increasing. We can use the Laplace method for approximating the integral when γ goes to infinity.

$$\phi'(v) = d\left(s^*_{\frac{1}{v}}(p_a,c), p_a\right) - \frac{1}{v}\frac{d'\left(s^*_{\frac{1}{v}}(p_a,c), p_a\right)}{d'\left(s^*_{\frac{1}{v}}(p_a,c), c\right)} \geq 0.$$

And $\phi'\left(\frac{C_1}{d(p_a,c)}\right) = 0$ iff $C_1 = 1$. If $C_1 > 1$ the following equivalent holds:

$$\int_{\frac{C_1}{d(p_a,c)}}^{\infty} \exp\left(-\gamma\phi(v)\right) \underset{\gamma\to\infty}{\sim} \frac{\exp\left(-\gamma\phi\left(\frac{C_1}{d(p_a,c)}\right)\right)}{\gamma\phi'\left(\frac{C_1}{d(p_a,c)}\right)}.$$

Noting that $s^*_{\frac{d(p_a,c)}{C_1}}(p_a,c) = s_{C_1}(p_a,c)$, we get

$$m_\gamma \underset{\gamma\to\infty}{\sim} \frac{\exp(-\gamma F_{C_1}(p_a,c))}{\phi'\left(\frac{C_1}{d(p_a,c)}\right)} \quad \text{with} \quad F_{C_1}(p_a,c) = \frac{C_1 d\left(s_{C_1}(p_a,c), p_a\right)}{d(p_a,c)}.$$

And $\phi'\left(\frac{C_1}{d(p_a,c)}\right)$ can be written as

$$\phi'\left(\frac{C_1}{d(p_a,c)}\right) = \frac{d(p_a,c)}{C_1}\left(F_{C_1}(p_a,c) - \frac{d'(s_{C_1}(p_a,c),p_a)}{d'(s_{C_1}(p_a,c),c)}\right).$$

This asymptotic equivalent shows that, starting from (17), we cannot improve the constant $F_{C_1}(p_a,c)$ in the exponential with a bigger (and maybe non problem-dependent) one. If $C_1 = 1$ the same reasoning holds, but the Laplace equivalent is different and leads to:

$$m_\gamma \underset{\gamma\to\infty}{\sim} \sqrt{\gamma}\sqrt{\frac{\pi}{-2\phi''\left(\frac{1}{d(p_a,c)}\right)}},$$

which is a trivial upper bound for a probability.

An 'optimal' bound of the probability We now give a non-asymptotic upper bound of (17) involving the optimal rate $F_{C_1}(p_a,c)$ in the exponential. If $v \geq \frac{C_1}{d(p_a,c)}$, $s^*_{\frac{1}{v}}(p_a,c) \geq s^*_{\frac{d(p_a,c)}{C_1}}(p_a,c)$ and we can use this bound in the integral in (17) to get:

$$P \quad \leq \quad \int_{\frac{C_1}{d(p_a,c)}}^{\infty} \exp\left(-ud\left(s_{C_1}(p_a,c), p_a\right)\right) du = \frac{\exp\left(-F_{C_1}(p_a,c)\gamma\right)}{d\left(s_{C_1}(p_a,c), p_a\right)}.$$

Proof of Lemma 1 For a given value of C_1, the following quantity can be upper bounded by P, using a trick shown in Garivier and Cappé (2011, see Lemma 7).

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{P}\left(a = u_t \vee a = l_t, N_a(t) > \left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil, N_a(t) d(\hat{p}_a(t), c) \leq \gamma\right) \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{a=u_t \vee a=l_t} \mathbb{1}_{N_a(t) > \left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil} \mathbb{1}_{N_a(t) d(\hat{p}_a(t),c) \leq \gamma}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \sum_{u=\left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil + 1}^{t} \mathbb{1}_{a=u_t \vee a=l_t} \mathbb{1}_{N_a(t)=u} \mathbb{1}_{ud(\hat{p}_{a,u},c) \leq \gamma}\right] \\
&\leq \mathbb{E}\left[\sum_{u=\left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil + 1}^{T} \mathbb{1}_{ud(\hat{p}_{a,u},c) \leq \gamma} \sum_{t=u}^{T} \mathbb{1}_{a=u_t \vee l_t} \mathbb{1}_{N_a(t)=u}\right] \leq \mathbb{E}\left[\sum_{u=\left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil + 1}^{T} \mathbb{1}_{ud(\hat{p}_{a,u},c) \leq \gamma}\right] \\
&= \sum_{u=\left\lceil \frac{C_1\gamma}{d(p_a,c)} \right\rceil + 1}^{T} \mathbb{P}\left(ud(\hat{p}_{a,u}, c) \leq \gamma\right).
\end{aligned}$$

Using Lemma 2 to upper bound P with the choice of $C_1 = d^*(p_a, c)/d(p_a, c)$ lead to the inequality of Lemma 1.

Appendix D. Proof of Proposition 1

We first show that at time t, if the stopping condition does not hold ($U_{u_t} - L_{l_t} > \epsilon$) and the event W_t holds, then either $c \in \mathcal{I}_{u_t}(t)$ or $c \in \mathcal{I}_{l_t}(t)$. This comes from a straightforward adaptation of the beginning of the proof of Lemma 2 from Kalyanakrishnan et al. (2012). Then we also observe that if $U_{u_t} - L_{l_t} > \epsilon$, the two intervals $\mathcal{I}_{u_t}(t)$ and $\mathcal{I}_{l_t}(t)$ cannot be too small simultaneously. Indeed, Pinsker's inequality (6) and the fact that $\hat{p}_{u_t}(t) < \hat{p}_{l_t}(t)$ leads to

$$\tilde{\beta}_{u_t}(t) + \tilde{\beta}_{l_t}(t) > \epsilon \quad \text{with} \quad \tilde{\beta}_a(t) := \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}}. \tag{18}$$

Hence either $\tilde{\beta}_{u_t}(t) > \frac{\epsilon}{2}$ or $\tilde{\beta}_{u_t}(t) > \frac{\epsilon}{2}$. It remains to show that one of $k = l_t$ and $k = u_t$ such that $c \in \mathcal{I}_k(t)$ also satisfies this second condition. This part is the Proof uses properties of KL-divergence, and *cannot* directly be adapted from Kalyanakrishnan et al. (2012).

It remains to show that if $U_{u_t}(t) - L_{l_t}(t) > \epsilon$, then the four statements below hold.

$$c \in \mathcal{I}_{u_t}(t) \text{ and } c > U_{l_t}(t) \quad \Rightarrow \quad \tilde{\beta}_{u_t}(t) > \frac{\epsilon}{2}. \tag{19}$$

$$c \in \mathcal{I}_{u_t}(t) \text{ and } c < L_{l_t}(t) \quad \Rightarrow \quad \tilde{\beta}_{u_t}(t) > \frac{\epsilon}{2}. \tag{20}$$

$$c \in \mathcal{I}_{l_t}(t) \text{ and } c > U_{u_t}(t) \quad \Rightarrow \quad \tilde{\beta}_{l_t}(t) > \frac{\epsilon}{2}. \tag{21}$$

$$c \in \mathcal{I}_{l_t}(t) \text{ and } c < L_{l_t}(t) \quad \Rightarrow \quad \tilde{\beta}_{l_t}(t) > \frac{\epsilon}{2}. \tag{22}$$

To prove (19), note that if $c \in \mathcal{I}_{u_t}(t)$ and $c > U_{l_t}(t)$, one has

$$d(\hat{p}_{u_t}(t), c) \leq 2\tilde{\beta}_{u_t}(t)^2 \quad \text{and} \quad d(\hat{p}_{l_t}(t), c) \geq 2\tilde{\beta}_{l_t}(t)^2.$$

Moreover, as $c > U_{l_t}$, $c > \hat{p}_{l_t}(t) > \hat{p}_{u_t}(t)$ holds, and therefore $d(\hat{p}_{l_t}(t), c) \leq d(\hat{p}_{u_t}(t), c)$. Hence,

$$2\tilde{\beta}_{l_t}(t)^2 \leq d(\hat{p}_{l_t}(t), c) \leq d(\hat{p}_{u_t}(t), c) \leq 2\tilde{\beta}_{u_t}(t)^2 \quad \text{and} \quad \tilde{\beta}_{l_t}(t) \leq \tilde{\beta}_{u_t}(t)$$

This together with $\tilde{\beta}_{l_t}(t) + \tilde{\beta}_{u_t}(t) > \epsilon$ leads to $\tilde{\beta}_{u_t}(t) > \frac{\epsilon}{2}$ and proves statement (19). The proof of statement (21) use identical arguments.

The proof of statement (20) goes as follows :

$$\begin{aligned}
&(U_{u_t}(t) - L_{l_t}(t) > \epsilon) \cap (L_{u_t} < c) \cap (c < L_{l_t}(t)) \\
&\Rightarrow (U_{u_t}(t) > c + \epsilon) \cap (L_{u_t} < c) \\
&\Rightarrow (\hat{p}_{u_t}(t) + \tilde{\beta}_{u_t}(t) > c + \epsilon) \cap (\hat{p}_{u_t}(t) - \tilde{\beta}_{u_t}(t) < c) \\
&\Rightarrow 2\tilde{\beta}_{u_t}(t) > \epsilon.
\end{aligned}$$

And the proof of statement (22) is similar.

Appendix E. Upper bound on T1

T_1^* (resp. T_2^*) is the unique solution of the equation

$$x = \frac{1}{\beta} \log\left(\frac{t^\alpha}{\eta}\right) \tag{23}$$

with $\beta = \frac{1}{H^*}$ for T_1^* (resp. $\beta = \frac{1}{2H^*}$ for T_2^*) and $\eta = \frac{\delta}{k_1 K}$. The bounds given in Section 4.2 come from the following lemma.

Lemma 5 *Let x^* be the solution of (23) with β and η satisfying $\eta < \frac{1}{e}$, $\beta < 1$. Then*

$$\frac{1}{\beta} \log\left(\frac{1}{\beta^\alpha \eta}\right) \leq x^* \leq \frac{C_0}{\beta} \log\left(\frac{1}{\beta^\alpha \eta}\right)$$

with C_0 such that $C_0 \geq \alpha \log(C_0) + 1 + \frac{\alpha}{e}$.

Proof x^* is upper-bounded by any x such that $\frac{1}{\beta} \log\left(\frac{x^\alpha}{\eta}\right) < x$. We look for such x of the form $x = \frac{C_0}{\beta} \log\left(\frac{1}{\beta^\alpha \eta}\right)$. One has:

$$\begin{aligned}
\frac{1}{\beta} \log\left(\frac{x^\alpha}{\eta}\right) &= \frac{1}{\beta} \log\left(\frac{C_0^\alpha \left(\log\left(\frac{1}{\beta^\alpha \eta}\right)\right)^\alpha}{\beta^\alpha \eta}\right) = \frac{1}{\beta}\left[\alpha \log C_0 + \log \frac{1}{\beta^\alpha \eta} + \alpha \log\log \frac{1}{\beta^\alpha \eta}\right] \\
&\leq \frac{1}{\beta}\left[\alpha \log(C_0) + \left(1 + \frac{\alpha}{e}\right) \log\left(\frac{1}{\beta^\alpha \eta}\right)\right] \\
&\leq \frac{1}{\beta}\left[\alpha \log(C_0) + \left(1 + \frac{\alpha}{e}\right)\right] \log\left(\frac{1}{\beta^\alpha \eta}\right)
\end{aligned}$$

where the first inequalities uses that $\forall x, x + \alpha \log(x) \leq (1 + \frac{\alpha}{e})x$ and the second inequality holds because $\beta^\alpha \eta < \frac{1}{e}$. Therefore, choosing C_0 such that

$$\alpha \log(C_0) + \left(1 + \frac{\alpha}{e}\right) < C_0$$

yields the inequality $\frac{1}{\beta} \log\left(\frac{x^\alpha}{\eta}\right) < x$. The lower bound comes from the fact that the sequence defined by

$$\begin{aligned} t_0 &= 1, \text{ and} \\ t_{n+1} &= \frac{1}{\beta} \log\left(\frac{t_n^\alpha}{\eta}\right) \end{aligned}$$

is increasing ($\beta < 1$ and $\eta < \frac{1}{e}$ imply that $t_0 \leq t_1$) and converges to x^*. Hence,

$$x^* \geq t_2 = \frac{1}{\beta} \log\left(\frac{1}{\eta \beta^\alpha}\right) + \frac{\alpha}{\beta} \log\log\left(\frac{1}{\eta}\right) \geq \frac{1}{\beta} \log\left(\frac{1}{\eta \beta^\alpha}\right).$$

□

Appendix F. KL-LUCB for the Fixed Budget Setting

Gabillon et al. (2012) and Kalyanakrishnan (2011, Section 5.4) suggest methods to turn an algorithm $\mathcal{A}$ for Explore-m into an algorithm for Explore-m-FB when the complexity of the problem is known. Their two ideas have in common to use an exploration rate depending on n and on the complexity (and no longer on t and δ) and to use the sampling strategy of algorithm $\mathcal{A}$ with this exploration rate during n rounds. The idea of Gabillon et al. (2012) is to store the sets $J(t)$ and the value of the stopping criteria $B(t)$ of $\mathcal{A}$ at all rounds $t = 1, 2, \ldots, n$ and to recommend the set $J(t)$ associated with the smallest value of $B(t)$ (i.e., the output is the set for which $\mathcal{A}$ was the closest from stopping). The idea of Kalyanakrishnan (2011) is to output the result of $\mathcal{A}$ if it has stopped before n, and any other set otherwise (e.g. the empirical m best arms). Here we focus on this last idea to define KL-LUCB-E, but the bound on e_n given in Theorem 4 also holds for the approach suggested by Gabillon et al. (2012).

Definition 2 *Let $\mathcal{A}_b$ denote the KL-LUCB algorithm with $\beta(t, \delta) = b$ and τ_b be the corresponding stopping time. The KL-LUCB-E algorithm with parameter b runs $\mathcal{A}_b$ up to at least round $n/2$. If $\tau_b < n/2$, it recommends the output of $\mathcal{A}_b$, else it recommends the empirical m best arms at round $n/2$ of $\mathcal{A}_b$.*

Theorem 4 *The KL-LUCB-E algorithm with parameter $b \leq \frac{n}{2H^*_{\epsilon,c}}$ uses at least n samples of the arms and satisfies*

$$e_n := \mathbb{P}(\mathcal{S}_n \not\subset \mathcal{S}^*_{m,\epsilon}) \leq (Kn + H^*_{\epsilon,c}) \exp(-b).$$

Especially for $b = \frac{n}{2H^*_{\epsilon,c}}$, one gets

$$e_n \leq \left(\frac{Kn}{2} + H^*_{\epsilon,c}\right) \exp\left(-\frac{n}{2H^*_{\epsilon,c}}\right). \tag{24}$$

KL-LUCB-E can be seen as a generalization to the Explore-m problem of the UCB-E algorithm proposed by Audibert et al. (2010) for Explore-1, and the corresponding upper bound involves Chernoff information. Of course, KL-LUCB-E requires the knowledge of the complexity of the problem and is therefore not very interesting. A variant of KL-LUCB-E estimating the complexity on-line could overcome this problem, as Audibert et al. (2010) do to turn UCB-E into an efficient algorithm. Even when KL-LUCB-E is given the complexity, our experiments report that it is not very efficient. This can be partly explained by the fact that on difficult problems (where $H^*_{\epsilon,c}$ is large) (24) is a non trivial bound (smaller than 1) only for very big values of n. Thus there are no guarantees for KL-LUCB-E to be practically efficient for reasonable values of n.

The SAR algorithm of Bubeck et al. (2013), based on uniform sampling is more reasonable since it does not require the complexity of the problem. However, as demonstrated in our numerical experiments in Section 5, smart sampling based on a "standard" exploration rate such as $\log(t)$ can indeed outperform SAR *in practice*, even if it lacks a theoretical error bound. Thus, adaptive sampling appears to be a relevant approach even for Explore-m-FB.

Proof of Theorem 4 We introduce $n' = \frac{n}{2}$ and write

$$\begin{aligned} e_n &= \mathbb{P}(\mathcal{S}_n \nsubseteq \mathcal{S}^*_{m,\epsilon} | \tau_b \leq n') \mathbb{P}(\tau_b \leq n') + \mathbb{P}(\mathcal{S}_n \nsubseteq \mathcal{S}^*_{m,\epsilon} | \tau_b > n') \mathbb{P}(\tau_b > n') \\ &\leq \mathbb{P}(\mathcal{S}_n \nsubseteq \mathcal{S}^*_{m,\epsilon} | \tau_b \leq n') + \mathbb{P}(\tau_b > n'). \end{aligned}$$

If $\tau_b \leq n'$, $\mathcal{S}_n \nsubseteq \mathcal{S}^*_{m,\epsilon}$ implies that $\mathcal{A}_b$ must return a wrong set on $[0, n']$ and as already seen, that the event W_n^c holds, where

$$W_n = \bigcap_{t \leq n'} \bigcap_{a \in \mathcal{S}^*_{m,\epsilon}} (u_a(t) > p_a) \bigcap_{b \in (\mathcal{S}^*_{m,\epsilon})^c} (l_b(t) < p_b).$$

To bound $\mathbb{P}(\tau_b > n')$ we use the same technique as in the proof of Theorem 3 (using the same notations)

$$\min(\tau_b, n') \leq H^*_{\epsilon,c} b + \underbrace{\sum_{a \in \mathcal{A}^c_\epsilon} \sum_{t=1}^{n'} \mathbb{1}_{(a=l_t) \vee (a=u_t)} \mathbb{1}_{N_a(t) d(\hat{p}_a(t), c) \leq b} \mathbb{1}_{N_a(t) > \frac{b}{d^*(p_a, c)}} + \mathbb{1}_{W_n^c}}_{R_{n'}}.$$

Hence, for $b \leq \frac{n'}{H^*_{\epsilon,c}}$, on $(R_{n'} = 0)$ one has $\min(\tau_b, n') \leq n'$ and $\mathcal{A}_b$ has stopped after n' rounds.

$$\begin{aligned} \mathbb{P}(\tau_b > n') &\leq \mathbb{P}(W_n^c) + \sum_{a \in \mathcal{A}^c_\epsilon} \sum_{t=1}^{n'} \mathbb{P}\left(a = l_t \vee u_t, N_a(t) > \frac{b}{d^*(p_a, c)}, N_a(t) d(\hat{p}_a(t), c) \leq b \right) \\ &\leq \mathbb{P}(W_n^c) + H^*_{\epsilon,c} \exp(-b). \end{aligned}$$

where the inequality follows from Lemma 1. We have

$$e_n \leq H^*_{\epsilon,c} \exp(-b) + 2\mathbb{P}(W_n^c).$$

To bound $\mathbb{P}(W_n^c)$, we use that for all a,

$$\begin{aligned}
\mathbb{P}(\exists t \leq n' : u_a(t) < p_a) &= \mathbb{P}(\exists t \leq n' : N_a(t) d^+(\hat{p}_a(t), p_a) \geq b) \\
&= \mathbb{P}(\exists t \leq n', \exists s \leq t : s d^+(\hat{p}_{a,s}, p_a) \geq b) \\
&= \mathbb{P}(\exists s \leq n' : s d^+(\hat{p}_{a,s}, p_a) \geq b) \\
&\leq \sum_{s=1}^{n'} \mathbb{P}(s d^+(\hat{p}_{a,s}, p_a) \geq b) \leq n' \exp(-b),
\end{aligned}$$

where the last inequality is a consequence of Chernoff inequality. Symmetrically,

$$\mathbb{P}(\exists t \leq n' : l_b(t) > p_b) \leq n' \exp(-b)$$

also holds and finally, using an union bound, we get for $b \leq \frac{n}{2H^*_{\epsilon,c}}$,

$$e_n \leq H^*_{\epsilon,c} \exp(-b) + \frac{Kn}{2} \exp(-b).$$

□