Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

PAC Identification of a Bandit Arm Relative to a Reward Quantile

Arghya Roy Chaudhuri, Shivaram Kalyanakrishnan
Department of Computer Science and Engineering
Indian Institute of Technology Bombay, Mumbai 400076, India
{arghya, shivaram}@cse.iitb.ac.in

Abstract

We propose a PAC formulation for identifying an arm in an n-armed bandit whose mean is within a fixed tolerance of the m^{th} highest mean. This setup generalises a previous formulation with $m = 1$, and differs from yet another one which requires m such arms to be identified. The key implication of our proposed approach is the ability to derive upper bounds on the sample complexity that depend on n/m in place of n. Consequently, even when the number of arms is *infinite*, we only need a finite number of samples to identify an arm that compares favourably with a fixed reward quantile. This facility makes our approach attractive to applications such as drug discovery, wherein the number of arms (molecular configurations) may run into a few thousands.

We present sampling algorithms for both the finite- and infinite-armed cases, and validate their efficiency through theoretical and experimental analysis.We also present a lower bound on the worst case sample complexity of PAC algorithms for our problem, which matches our upper bound up to a logarithmic factor.

1 Introduction

The Multi-armed bandit (Berry and Fristedt 1985) is a well-studied abstraction of decision making under uncertainty. Each *arm* of a bandit may be *pulled* in order to obtain a reward. The experimenter, based on his/her objective, must decide which arm to pull at every round. He/she may consult the preceding history of pulls and rewards in order to make this decision. Bandits were originally introduced as an abstraction of clinical drug testing (Robbins 1952), with each arm corresponding to a drug, a pull its administration to a particular patient, and the reward the success of the treatment. Subsequently, the formalism has also found application in simulation optimisation (Amaran et al. 2015), game-playing (Goschin et al. 2012), and on-line advertising (Li et al. 2010).

We consider the *stochastic* multi-armed bandit (Auer, Cesa-Bianchi, and Fischer 2002; Robbins 1952), wherein the rewards from each arm come as i.i.d. samples from a corresponding distribution (by contrast, the rewards may come from an *arbitrary* sequence in the *adversarial* setting (Auer et al. 2003)). The predominant body of work in the stochastic setting addresses *regret minimisation*, which amounts to maximising the expected sum of rewards from the pulls of the arms. Naturally a successful strategy here must *explore* the arms sufficiently to infer their mean rewards, but also *exploit* profitable arms enough to reap high rewards.

More recently, applications such as product testing (Audibert, Bubeck, and Munos 2010)—wherein there is a dedicated experimentation period *before* the product is launched—have also motivated the study of the *pure exploration* setting. In this setting, the rewards accrued while experimenting are immaterial; the objective is to either (1) minimise the number of pulls required to identify an arm that satisfies a reward constraint (Even-Dar, Mannor, and Mansour 2002; Kalyanakrishnan et al. 2012; Mannor et al. 2004), or to (2) maximise the expected reward of an arm that is returned after a specified number of pulls (Audibert, Bubeck, and Munos 2010; Carpentier and Valko 2015). Our contributions, which we introduce below, fall in the first category.

Background. Let us consider an arbitrary instance of an n-armed bandit, $n \geq 2$, with a set of arms $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$. Each pull of an arm returns a reward drawn i.i.d. from a corresponding distribution; for simplicity we assume that reward distributions have their support in $[0, 1]$. Let the mean reward of arm $a \in A$ be μ_a. Without loss of generality, assume $\mu_{a_i} \geq \mu_{a_j}$ if $i < j$, for $i, j \in \{1, 2, \ldots, n\}$. The means of the bandit arms are not known to the experimenter, who must implement an *algorithm*. Based on the historical sequence of pulls and rewards, an algorithm either (1) terminates and returns an answer, or (2) specifies an arm to be pulled next. Algorithms can be either deterministic or randomised.

Given a "tolerance" $\epsilon \in [0, 1]$, and given $m \in \{1, 2, \ldots, n\}$, we define an arm $a \in A$ to be (ϵ, m)-optimal if $\mu_a \geq \mu_{a_m} - \epsilon$. The earliest treatment of the pure exploration setting, by Even-dar *et al.* (2002), considers the design of algorithms, which, for a given mistake probability $\delta \in (0, 1]$, must return an $(\epsilon, 1)$-optimal arm with probability at least $1 - \delta$. This *Probably Approximately Correct (PAC)* formulation to identify a near-optimal arm was subsequently generalised by Kalyanakrishnan and Stone (2010) to the problem of finding a near-optimal *subset* of arms, as defined below.

Definition 1 (SUBSET) *An instance of* SUBSET *is fixed by an n-armed bandit instance, $n \geq 2$, with a set of arms $\mathcal{A}$; $m \in \{1, 2, \ldots, n\}$; $\epsilon \in (0, 1]$; and $\delta \in (0, 1]$. An algorithm $\mathcal{L}$ is said to solve* SUBSET *if for every instance $(n, \mathcal{A}, m, \epsilon, \delta)$, $\mathcal{L}$ terminates with probability 1, and returns m (ϵ, m)-optimal arms with probability at least $1 - \delta$.*

Kalyanakrishnan and Stone (2010) generalise the Median-Elimination algorithm introduced by Even-dar *et al.* (2002) to obtain an algorithm that solves SUBSET while taking at most $O((n/\epsilon^2)\log(m/\delta))$ pulls for every instance $(n, \mathcal{A}, m, \epsilon, \delta)$. Kalyanakrishnan *et al.* (2012) also show that there exist problem instances $(n, \mathcal{A}, m, \epsilon, \delta)$ for which any algorithm that solves SUBSET must incur $\Omega((n/\epsilon^2)\log(m/\delta))$ pulls, establishing the tightness of the upper bound up to a constant factor.

Problem statement. In this paper, we propose a problem that is related to SUBSET, but which is *easier* to solve. Under this problem—denoted Q-F, for reasons explained below—it is only required that a *single* (ϵ, m)-optimal arm be returned.

Definition 2 (Q-F) *An instance of* Q-F *is fixed by an n-armed bandit instance, $n \geq 2$, with a set of arms $\mathcal{A}$; $m \in \{1, 2, \ldots, n\}$; $\epsilon \in (0, 1]$; and $\delta \in (0, 1]$. An algorithm $\mathcal{L}$ is said to solve* Q-F *if for every instance $(n, \mathcal{A}, m, \epsilon, \delta)$, $\mathcal{L}$ terminates with probability 1, and returns an (ϵ, m)-optimal arm with probability at least $1 - \delta$.*

Q-F caters to the situation most-commonly encountered in practice, in which it suffices to identify an arm that compares favourably with a specified reward quantile, say 90%. It would be unnecessarily expensive, in such cases, to try and (1) identify an $(\epsilon, 1)$-optimal arm (which is defined with respect to the the 100% quantile) (Even-Dar, Mannor, and Mansour 2002), or to (2) identify an entire $(n/10)$-sized subset of arms (Kalyanakrishnan and Stone 2010). In principle, one can control the trade-off between *sample efficiency* (the number of pulls) and *quality* (mean reward) in SUBSET by varying ϵ, the tolerance threshold. However, a close inspection of SUBSET unearths a more fundamental limitation. Notice that the lower bound for SUBSET has a linear dependence on the number of arms; clearly SUBSET is infeasible to solve when the number of arms is large (potentially even infinite). On the other hand, it is quite common to encounter applications with a large number of arms. For example, arms might correspond to strategies for playing games (Goschin et al. 2012), locations for mining minerals (Berry et al. 1997), or molecular configurations for drug design (Berry et al. 1997). In all these cases, the number of arms is at least a few thousands. SUBSET is no longer a good modeling choice. Q-F, however, continues to offer a viable way to identify a good arm.

Essentially, Q-F requires the identification of an arm whose reward is within ϵ of the $(1 - \rho)$-quantile, for $\rho = m/n \in (0, 1]$. Interestingly, we can bound the number of samples needed to do so in terms of ρ, but *independent* of n. The strategy to escape the dependence on n is randomisation. Observe that if $\tilde{O}(1/\rho)$ arms are selected uniformly at random from $\mathcal{A}$, at least one must exceed the $(1-\rho)$-quantile with high probability. This arm can be isolated (modulo ϵ) by sampling each selected arm $\tilde{O}(1/\epsilon^2)$ times, yielding an overall bound of $\tilde{O}(1/\rho\epsilon^2)$ samples. In fact, this strategy does not need full access to the set of arms $\mathcal{A}$; rather, it only needs to be able to draw arms from $\mathcal{A}$ at random. Thus, the strategy may be extended to bandit instances wherein $\mathcal{A}$ is not finite, as long as arms can be drawn from $\mathcal{A}$ based on a probability distribution $P_{\mathcal{A}}$. Formally, define an arm $a \in A$ to be $[\epsilon, \rho]$-optimal, $\epsilon, \rho \in (0, 1]$, if $P_{a' \sim_{P_{\mathcal{A}}} \mathcal{A}}\{\mu_a \geq \mu_{a'} - \epsilon\} \geq 1 - \rho$. Whereas Q-F optimises with respect to a reward Quantile in a Finite armed bandit, we may extend the same idea to bandit instances whose arms can be drawn at random based on a Probability distribution.

Definition 3 (Q-P) *An instance of* Q-P *is fixed by a bandit instance with a set of arms $\mathcal{A}$; a probability distribution $P_{\mathcal{A}}$ over $\mathcal{A}$; $\rho \in (0, 1]$; $\epsilon \in (0, 1]$; and $\delta \in (0, 1]$. An algorithm $\mathcal{L}$ is said to solve* Q-P *if and only if for every instance $(\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta)$, $\mathcal{L}$ terminates with probability 1, and returns an $[\epsilon, \rho]$-optimal arm with probability at least $1 - \delta$.*

Contributions. In this paper, we propose efficient algorithms for Q-F and Q-P.

1. First, we present algorithms and corresponding upper bounds that apply to every instance of Q-F and Q-P (Section 3). Here we also present a lower bound on the worst case sample complexity of the PAC algorithms for Q-F which is also extensible to Q-P.
2. Although the upper bounds in Section 3 apply even to the hardest bandit instances (in which the arms all have the same mean), in practice one will likely encounter instances with some separation between arms. In Section 4, we describe a fully sequential PAC algorithm for Q-F that achieves a tighter instance-specific bound than those that carry over from SUBSET (Even-Dar, Mannor, and Mansour 2002; Kalyanakrishnan et al. 2012). Our proof incorporates ideas from the analysis of the LUCB algorithm (Kalyanakrishnan et al. 2012) for SUBSET.
3. In Section 4, we extend our ideas to derive a fully-sequential algorithm for algorithm Q-P. The algorithm is parameterised by the number of arms it can keep in memory. Although a constant-size memory suffices for correctness, we present experimental results (in Section 5) to demonstrate the practical gains of using larger memory sizes.

Before presenting the technical contributions of the paper, we briefly contrast them with related work.

2 Related Work

Whereas infinite-armed bandits have been considered before in the literature, most previous studies make structural assumptions about the arms and/or the rewards. Starting with Agrawal's "continuum-armed bandit" (Agrawal 1995), there is a significant body of literature devoted to the setting wherein the arms are embedded in a metric space, with respect to which the rewards are continuous (Kleinberg 2005; Auer, Ortner, and Szepesvári 2007; Kleinberg, Slivkins, and Upfal 2008). A separate line of research, while making no assumption about the relationship between arms, assumes

specific properties for the reward distribution: for example that it is uniform (Berry et al. 1997), or that there is a concentration of mass around the optimal reward (Wang, Audibert, and Munos 2008; Carpentier and Valko 2015). Among these it is worth mentioning that whereas all the other efforts listed above focus on regret minimisation, Carpentier and Valko (2015) aim to minimise *simple regret*: a pure exploration setting in which the number of pulls (rather than the tolerance and mistake probability) is fixed beforehand.

It is an empirical question whether for a given application of infinite-armed bandits—of which there are several (Berry et al. 1997)—one can find features to generalise well across the arms, or if indeed, the set of rewards possesses some structure. We make no assumptions in either respect.Thus, our approach is not preferable when useful inductive biases are available, but preferable when they are not.

The line of work most closely related to ours is that of Goschin *et al.* (2012). In their PAC formulation, the objective is to identify an arm whose mean reward exceeds a specified threshold r_0. At the heart of both this formulation and ours is the "good" probability mass lying above r_0. We believe it is more natural to fix this mass (equivalently, the quantile) rather than the actual reward, which may not be easy to guess. Users generally do not have a very good estimate of the highest or near-highest reward if the reward distribution across the arms is fat-tailed. In such cases, it could be inefficient to find an arm with satisfactory expected reward. Note that Q-F and Q-P are well-defined for every bandit instance as long as $\epsilon > 0$; our algorithm in Section 4 can easily be adapted to work even when $\epsilon = 0$ on bandit instances with a non-zero separation of means. We view our work as a more practical cousin of that of Goschin *et al.* (2012). For example, we provide fully sequential algorithms for Q-F and Q-P, whereas Goschin *et al.* primarily focus on worst-case bounds.

As mentioned in Section 1, our work draws on previous PAC formulations (Even-Dar, Mannor, and Mansour 2002; Mannor et al. 2004; Kalyanakrishnan and Stone 2010; Kalyanakrishnan et al. 2012), which we cite wherever appropriate in the paper. Our work is not to be confused with the "quantile-based approach" of Szörényi *et al.* (2015), in which the quantile in question is of *each* arm's reward distribution, with the rewards themselves coming from an arbitrary totally-ordered set. On the other hand, our definition of an arm's optimality (in terms of a reward quantile) is similar to that proposed by Chaudhuri *et al.* (2009), who consider regret minimisation in the adversarial, full-information setting. Their objective is to devise a parameter-free on-line learning algorithm. By contrast, our objective is to eliminate the sampling algorithm's dependence on the number of arms in a PAC setting.

3 Worst-Case Bounds Across Problem Instances

In this section, we bound the worst-case sample complexity of algorithms for Q-F and Q-P across bandit instances. First we obtain an upper bound on the sample complexity of algorithms that solve Q-P; the *reducibility* of Q-F to Q-P implies a corresponding upper bound for Q-F. Specifically, let $I_F = (n, \mathcal{A}, m, \epsilon, \delta)$ be an instance of Q-F. Now consider the instance $I_P = (\mathcal{A}, U_{\mathcal{A}}, (m-1/2)/n, \epsilon, \delta)$ of Q-P, where $U_{\mathcal{A}}$ is the uniform distribution over $\mathcal{A}$. By this construction, an $[\epsilon, \rho]$-optimal arm for I_P is necessarily an (ϵ, m)-optimal arm for I_F, since

$$P_{a' \sim_{U_{\mathcal{A}}} \mathcal{A}}\{\mu_a \geq \mu_{a'} - \epsilon\} \geq 1 - (m - 1/2)/n$$
$$\Rightarrow \mu_a \geq \mu_{a_m} - \epsilon.$$

Therefore, an algorithm that solves Q-P can easily be adapted into one that solves Q-F. To get an upper bound for Q-P, we consider an algorithm, $\mathcal{P}_1$, which implements the strategy mentioned in Section 1 (and is similar to one proposed by Goschin *et al.* (2012, see Algorithm 5)). Given an instance $(\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta)$, the first step in $\mathcal{P}_1$ is to draw a set of arms from $\mathcal{A}$ i.i.d. according to $P_{\mathcal{A}}$, treating possible repetitions of arms as different arms. Let the set of arms drawn be $\mathcal{A}'$, and let the number of arms drawn be

$$|\mathcal{A}'| = N(\rho, \delta) = \lceil (1/\rho) \ln(2/\delta) \rceil .$$

The second step in $\mathcal{P}_1$ is to apply the Median-Elimination algorithm (Even-Dar, Mannor, and Mansour 2002) to find an $(\epsilon, 1)$-optimal arm in $\mathcal{A}'$, specifically by solving instance $(N(\rho, \delta), \mathcal{A}', 1, \epsilon, \delta/2)$ of SUBSET. This arm (more precisely, its corresponding arm in $\mathcal{A}$) is returned by $\mathcal{P}_1$.

Theorem 3.1 *$\mathcal{P}_1$ solves Q-P. Given an instance $(\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta)$, its number of pulls is upper-bounded by $O((1/\rho\epsilon^2)(\log(1/\delta))^2)$.*

Proof The probability that $\mathcal{A}'$ does *not* contain a $(0, \rho)$-optimal arm is at most $(1-\rho)^{N(\rho,\delta)} \leq \delta/2$, and the probability that the Median-Elimination algorithm does not find an $(\epsilon, 1)$-optimal arm in $\mathcal{A}'$ is at most $\delta/2$. Thus, the probability that $\mathcal{P}_1$ does not identify $[\epsilon, \rho]$-optimal arm in $\mathcal{A}$ is at most δ. The sample complexity is $O((N(\rho, \delta)/\epsilon^2) \log(1/\delta))$ (Even-Dar, Mannor, and Mansour 2002), which gives the desired result.

Let $\mathcal{F}_1$ be an algorithm for Q-F that translates its input into an instance of Q-P (as described above) and then applies $\mathcal{P}_1$. The correctness and sample complexity of $\mathcal{F}_1$ are immediate from Theorem 3.1.

Corollary 3.2 *$\mathcal{F}_1$ solves Q-F. Given an instance $(n, \mathcal{A}, m, \epsilon, \delta)$, its number of pulls is upper-bounded by $O((n/m\epsilon^2)(\log(1/\delta))^2)$.*

We can also show lower bounds for Q-F and Q-P, which match the corresponding upper bounds up to a $\Theta\left(\log \frac{1}{\delta}\right)$ factor.

Theorem 3.3 *For $0 < \epsilon \leq \frac{1}{\sqrt{32}}$, $0 < \delta < \frac{e^{-1}}{4}$, and $n \geq 2m$, an algorithm $\mathcal{L}$ that solves Q-F must incur a sample complexity of at least $\frac{1}{306} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m} \ln\left(\frac{1}{4\delta}\right)$.*

The above theorem admits a proof which is similar to the proof of Theorem 8 of Kalyanakrishnan *et al.* (2012), but differs in the fact that their proof relies on the highest m arms having to be returned (under Q-F, only one has to be returned). *Our* detailed proof is given in the supplemental material. This lower bound can be easily extended to Q-P.

Corollary 3.4 *For* $0 < \epsilon \leq \frac{1}{\sqrt{32}}$, $0 < \delta < \frac{e^{-1}}{4}$, *and* $0 < \rho \leq 1/2$ *an algorithm* $\mathcal{L}$ *that solves* Q-P *must incur a sample complexity of at least* $\frac{1}{306} \cdot \frac{1}{\epsilon^2} \cdot \frac{1}{\rho} \ln\left(\frac{1}{4\delta}\right)$.

4 Fully Sequential Algorithms for Q-F and Q-P

We expect bandit instances encountered in practice to have arms whose means are relatively well-spread apart. If so, our algorithms can (implicitly) estimate the gaps between the arms by monitoring empirical rewards, and use this information to sample and stop more efficiently. The LUCB algorithm (Kalyanakrishnan et al. 2012) for SUBSET applies this idea. Under LUCB, the m arms with the highest *empirical* means are kept in one set, and the remaining arms in another. The most "contentious" arm from each set—defined in terms of lower and upper confidence bounds—is sampled. The algorithm stops and returns the first set if its lowest lower confidence bound does not overlap the highest upper confidence bound in the second set by more than ϵ. This *fully sequential* algorithm (in the sense that it samples $O(1)$ arms before making the next decision) enjoys an expected sample complexity bound that depends on the gaps between the arms. Our first exercise in this section is to describe an algorithm, $\mathcal{F}_2$, which achieves a similar bound for Q-F.

4.1 Algorithm $\mathcal{F}_2$

Define, for $a \in \mathcal{A}$:

$$\Delta_a = \begin{cases} \mu_{a_1} - \mu_{a_{m+1}}, \text{ if } a \in \{a_1, a_2, \ldots, a_m\}, \\ \mu_{a_1} - \mu_a, \text{ otherwise,} \end{cases}$$

and define $H_\epsilon = \sum_{a \in \mathcal{A}} 1/\max(\Delta_a, \epsilon/2)^2$. Our main contribution is an $O(H_\epsilon \log(H_\epsilon/\delta))$ upper bound for $\mathcal{F}_2$, which resembles the bound provided by Kalyanakrishnan *et al.* (2012) for LUCB. Recall that Q-F can be solved by solving an instance of SUBSET with any subset size up to m. However, it can be verified that the resulting bounds (by applying LUCB) can never be tighter than ours: provably, the corresponding "Δ" values for the arms in the LUCB bound are equal or smaller than those in our bound.

Algorithm 1 describes $\mathcal{F}_2$, our fully sequential algorithm for solving Q-F. For convenience, we use notation similar to that used to describe LUCB (Kalyanakrishnan et al. 2012). The algorithm proceeds sequentially through rounds $t = 1, 2, \ldots$. Let u_a^t be the number of times arm $a \in \mathcal{A}$ has been pulled until (and excluding) round t, and $\hat{p}_a^t$ the corresponding empirical mean. We draw upper and lower confidence bounds on the true mean of the arms, given by $ucb(a,t) = \hat{p}_a^t + \sqrt{\frac{1}{2u_a^t} \ln\left(\frac{knt^4}{\delta}\right)}$, and $lcb(a,t) = \hat{p}_a^t - \sqrt{\frac{1}{2u_a^t} \ln\left(\frac{knt^4}{\delta}\right)}$, where $k = 5/4$ (we can use tighter concentration bounds in practice, as we do in our experiments).

At each round t, we partition the set of arms into three sets: we put the arm with the highest lower confidence bound in the set A_1^t. We sort the remaining arms in non-increasing order of their upper confidence bounds, placing the highest $m-1$ arms in the set A_2^t, and the remaining $n-m$ arms in the set A_3^t. Ties are broken arbitrarily. At each round the sampling strategy ensures exploration from each these three sets by pulling a "contentious" arm from each. Since A_1^t is a singleton set, there is no choice: the arm pulled is the unique element h_*^t. From A_2^t, we choose m_*^t, the arm that has been pulled the fewest times so far. The arm l_*^t pulled from A_3^t is the one with the highest upper confidence bound in this set. The algorithm terminates and returns h_*^t if its lower confidence bound gets sufficiently separated from the upper confidence bound of l_*^t.

Algorithm 1: $\mathcal{F}_2$

Input: $n, \mathcal{A}, m, \epsilon, \delta$
Output: $a \in \mathcal{A}$
Pull each arm $a \in \mathcal{A}$ for once. $t = n$.
do
  $t = t + 1$.
  $A_1^t = \{a : a \in \mathcal{A}, a = \arg\max_{a' \in \mathcal{A}} lcb(a', t)\}$.
  $A_2^t = \{a : a \in \mathcal{A} \setminus A_1^t \text{ s.t. } ucb(a,t) \geq ucb(a',t) \forall a' \in S \subset \mathcal{A} \setminus A_1^t \text{ s.t. } |S| = n - m\}$.
  $A_3^t = \mathcal{A} \setminus \{A_1^t \cup A_2^t\}$.
  $h_*^t \in A_1^t$. // Note that $|A_1^t| = 1$.
  $m_*^t = \arg\min_{\{a \in A_2^t\}} u_a^t$.
  $l_*^t = \arg\max_{\{a \in A_3^t\}} ucb(a,t)$.
  Pull h_*^t, m_*^t, and l_*^t.
while $ucb(l_*^t, t+1) - lcb(h_*^t, t+1) > \epsilon$
return h_*^t.

The analysis of $\mathcal{F}_2$ proceeds along similar lines as that of LUCB (Kalyanakrishnan et al. 2012). Our key contribution is in defining h_*^t, m_*^t, and n_*^t such that the desired dependence on H_ϵ can be shown.

Theorem 4.1 $\mathcal{F}_2$ *solves* Q-F*, and incurs an expected sample complexity of* $O\left(H_\epsilon \log\left(H_\epsilon/\delta\right)\right)$.

Proof The correctness of $\mathcal{F}_2$ is established by bounding the probability that the confidence bounds are ever violated (across arms and rounds), which is seen not to exceed δ. For each arm $a \in A$ and $t \geq 1$, define $u^*(a,t) = (1/2\max(\Delta_a, (\epsilon/2))^2) \ln(knt^4/\delta))$. We can show that the following event has a low probability.

$$CROSS_a^t \stackrel{\text{def}}{=} \mu_a < lcb(a,t) \text{ or } \mu_a > ucb(a,t).$$

Recall that $\mathcal{F}_2$ samples arms from the sets A_1^t, A_2^t, and A_3^t; our argument connects these arms with three related "ground truth" sets $B_1 = \{a_1\}, B_2 = \{a_2, ..., a_m\}$, and $B_3 = \{a_{m+1}, ..., a_n\}$. If a_1 is sampled at least $16u^*(a,t)$ times, the following event has a low probability.

$$ErrL_a^t \stackrel{\text{def}}{=} a \in B_3 \text{ and } lcb(a,t) > lcb(a_1,t).$$

If any arm $a \in \mathcal{A}$ is sampled at least $16u^*(a,t)$ times, the following events also have a low probability.

$$ErrU_a^t \stackrel{\text{def}}{=} a \in B_3 \text{ and } ucb(a,t) > ucb(a_1,t).$$

$$Needy_a^t = \begin{cases} \Big[(lcb(a,t) < \mu_a - \Delta_{a_{m+1}}/2) \wedge \\ (\hat{p}_a^t - lcb(a,t) > \epsilon/2)\Big] \text{ if } a \in B_1. \\ \Big[(ucb(a,t) > \mu_a + \Delta_a/2) \wedge \\ (ucb(a,t) - \hat{p}_a^t > \epsilon/2)\Big] \text{ if } a \in B_3. \end{cases}$$

The main part of the proof involves showing that if the "$CROSS$" event does not happen, and the algorithm has not terminated, then one of the "$ErrU$", "$ErrL$", and "$Needy$" events must have occurred. These events, in turn, imply with high probability that corresponding arms cannot have been sampled sufficiently. By an argument based on the pigeonhole principle (Kalyanakrishnan et al. 2012, see Lemma 5), the bound on the expected sample complexity follows.

Below we present the core logic underlying our analysis, to show that progress is made by the sampling strategy. If the algorithm has not terminated at round t, we carefully go through cases (it is worth recalling that m_*^t is the least sampled arm in A_2^t—a key property used by the proof).

if $\exists b_1 \in B_1 \cap A_1^t$
- **if** $\exists b_3 \in B_3 \cap A_2^t$ *and* $\forall b_3' \in B_3$ $ucb(b_3) \geq ucb(b_3')$:
  - Then either $Needy_{b_1}^t$ or $Needy_{b_3}^t$ must have occurred.
- **elif** $\exists b_3 \in B_3 \cap A_3^t$ *and* $\forall b_3' \in B_3$ $ucb(b_3) \geq ucb(b_3')$:
  - Then either $Needy_{b_1}^t$ or $Needy_{b_3}^t$ must have occurred.

elif $\exists b_1 \in B_1 \cap A_2^t$
- **if** $\exists b_3 \in B_3 \cap A_1^t$
  - Then the event $ErrL_a^t$ must have occurred.
- **elif** $\exists b_3 \in B_3 \cap A_2^t$ *and* $\forall b_3' \in B_3$ $ucb(b_3) \geq ucb(b_3')$:
  - Then either $Needy_{b_1}^t$ or $Needy_{b_3}^t$ must have occurred.
- **elif** $\exists b_3 \in B_3 \cap A_3^t$ *and* $\forall b_3' \in B_3$ $ucb(b_3) \geq ucb(b_3')$:
  - Then either $Needy_{b_1}^t$ or $Needy_{b_3}^t$ must have occurred.

elif $\exists b_1 \in B_1 \cap A_3^t$
- **if** $\exists b_3 \in B_3 \cap A_1^t$
  - **if** $b_1 == l_*^t$
    - Then the event $ErrL_a^t$ must have occurred.
  - **else**
    - Then either $l_*^t \in B_3$ or $B_3 \cap A_2^t \neq \emptyset$; therefore the event $ErrU_a^t$ must have occurred.
- **else**
  - Then $ErrU_a^t$ must have occurred.

The full proof of Theorem 4.1 is provided in the supplemental material.

Next we construct an algorithm for Q-P, using the LUCB algorithm (Kalyanakrishnan et al. 2012) as a subroutine. We denote this algorithm $\mathcal{P}_2$.

4.2 Algorithm $\mathcal{P}_2$

The basic structure underlying $\mathcal{P}_2$ is the same as that of $\mathcal{P}_1$, which was introduced in Section 1. Recall that $\mathcal{P}_1$ chooses $N(\rho, \delta)$ arms from $\mathcal{A}$, and proceeds to separate out an $(\epsilon, 1)$-optimal arm from among them by applying the Median-Elimination algorithm (Even-Dar, Mannor, and Mansour 2002). Surely we can improve upon this scheme in practice by replacing the call to the Median-Elimination algorithm with a call to LUCB instead (with a subset size of 1). The advantage is that LUCB will exploit the gaps between the means of the chosen arms while identifying an ϵ-optimal arm. We encounter a subtle difference between the two approaches (Median-Elimination and LUCB) when we consider the *memory* they require. In principle, the Median-Elimination approach can be implemented with constant-size memory, since only the arm with the *maximal* empirical mean, after each arm is sampled $\tilde{\Theta}(1/\epsilon^2)$ times, needs to be retained. On the other hand, LUCB gains efficiency by avoiding the $\tilde{\Theta}(1/\epsilon^2)$ pulls per arm, which it achieves by comparing confidence bounds. Thus, we can expect LUCB to perform best if, indeed, we can keep all of the $N(\rho, \delta) = \tilde{\Theta}(1/\rho)$ arms in memory and sample them judiciously. However, in practice it might not be feasible to store a large number of arms in memory.

We take a practical approach, giving the available memory size M as an *input* to the algorithm we develop for Q-P. This algorithm, $\mathcal{P}_2$ (see Algorithm 2), applies LUCB within batches of size M to separate out an ϵ-optimal *winner* efficiently. Batches are repeatedly refreshed until $N(\rho, \delta)$ arms have been evaluated. Note that a batch size of 2 corresponds to only storing the "currently maximal" arm and its challenger.

Algorithm 2: $\mathcal{P}_2$

Input: $\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta$, memory size $M \geq 2$.
Output: $a \in A$.

$\delta_0 = (\delta/2)\lfloor (M-1)/(N(\rho,\delta)-1) \rfloor$.
Choose an arm from $\mathcal{A}$ according to $P_{\mathcal{A}}$; place it in the set S.
$narms = 1$.
while $narms < N(\rho, \delta)$ **do**
  Choose $\min\{M-1, N(\rho,\delta) - narms\}$ arms from $\mathcal{A}$ according to $P_{\mathcal{A}}$; place them in the set R.
  $S = S \cup R$; $narms = narms + |R|$.
  $a = LUCB(|S|, S, \text{subset size} = 1, \epsilon, \delta_0)$.
  $S := \{a\}$.

The correctness of $\mathcal{P}_2$ is achieved by setting the mistake probability passed to LUCB accordingly. There are two separate aspects contributing to the sample complexity of $\mathcal{P}_2$: the hardness of a randomly chosen M-sized bandit instance, and the *sequence* in which such batches are encountered (since they are always compared with the currently maximal arm). It is easy to see that for the available memory size M, $\mathcal{P}_2$ needs $O((N(\rho,\delta)/\epsilon^2)\log(N(\rho,\delta)/(M\delta)))$ samples if it uses Median Elimination. Evidently the use of LUCB leads to lesser sample complexity. We verify this fact experimentally. Our experiments also validate the intuition that larger batch sizes lead to more efficiency.

5 Experiment and Results

We evaluate $\mathcal{F}_2$ by comparing it with (1) LUCB (Kalyanakrishnan et al. 2012), which can be applied with any setting of the subset size from 1 to m, and with (2) uniform sampling. All the algorithms terminate based on $\mathcal{F}_2$'s stopping rule. We simulate rewards based on Bernoulli distributions. For all algorithms we use KL-divergence based confidence

Table 1: $\mathcal{I}_1$: all arms except the best arm has the same mean. $\mathcal{I}_2$: difference of mean between two consecutive arms is the same. $\mathcal{I}_3$: the best 4 arms have the same mean and also the worst 6 arms have the same mean which is different from that of the best 4.

Arms:	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
$\mathcal{I}_1$	0.99	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
$\mathcal{I}_2$	0.99	0.88	0.77	0.66	0.55	0.44	0.33	0.22	0.11	0.01
$\mathcal{I}_3$	0.99	0.99	0.99	0.99	0.7	0.7	0.7	0.7	0.7	0.7

bounds (Cappé et al. 2013; Kaufmann and Kalyanakrishnan 2013), which, in our experiments, give at least a 10-fold improvement over the Hoeffding-based bounds used in Section 4 for the purposes of illustration.

We consider three 10-armed bandit instances, which are described in Table 1. We set $m = 4, \epsilon = 0$ and $\delta = 0.001$. The corresponding sample complexity is plotted in Figure 1(a). Observe that $\mathcal{F}_2$ outperforms both uniform sampling and LUCB on all three instances. Since $\mu_{a_4} = \mu_{a_5}$ and $\epsilon = 0$, LUCB with a subset size of 4 does not terminate on instance $\mathcal{I}_1$. On the other hand, $\mathcal{F}_2$ quickly identifies a_1. Figures 1(b) and 1(c) show the distribution of pulls across the arms on $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively: $\mathcal{F}_2$ seldom samples any single arm more than any of the other algorithms.

Although $\mathcal{F}_2$ is more efficient than LUCB on Q-F, both algorithms take progressively longer to terminate as the number of arms is increased. Figure 1(d) shows the performance of $\mathcal{F}_2$ on four bandit instances, with $n = 20, 100, 1000,$ and 3000, respectively. Each instance contains four equal-sized groups of arms, whose expected rewards are 0.99, 0.66, 0.33, and 0.01. After setting $\rho = 0.1, \epsilon = 0.05$, and $\delta = 0.001$, we pass a subset size of $m = \rho n$ to $\mathcal{F}_2$. Observe that even as the complexity of $\mathcal{F}_2$ increases with n, those of the other three algorithms in the figure—$\mathcal{P}_2$ with batch sizes of 2, 30, and 77 ($= N(\rho, \delta)$)—remains *constant*. On the 3000-armed instance, $\mathcal{P}_2$ is significantly better with any of these memory sizes; on each instance, a larger memory size leads to faster termination. Thus, in applications with a large number of arms, $\mathcal{P}_2$ is the method of choice.

6 Conclusion

This paper presents a PAC problem formulation for identifying an arm in a multi-armed bandit whose mean lies within a specified tolerance of a specified quantile. This problem is easier to solve than related problems studied previously in the literature (Even-Dar, Mannor, and Mansour 2002; Kalyanakrishnan and Stone 2010), and unlike them, remains viable to solve even when the number of arms is large (possibly infinite). Our formulation is similar to another previously-proposed one in which an arm that exceeds a specified reward threshold is to be returned (Goschin et al. 2012). It might not be easy for an experimenter to guess a suitable (or even valid) reward threshold; by contrast, our problem (in terms of quantiles) is well-defined for every bandit instance.

We discuss two variants of our problem: when the arms can be enumerated, and when they can only be chosen at

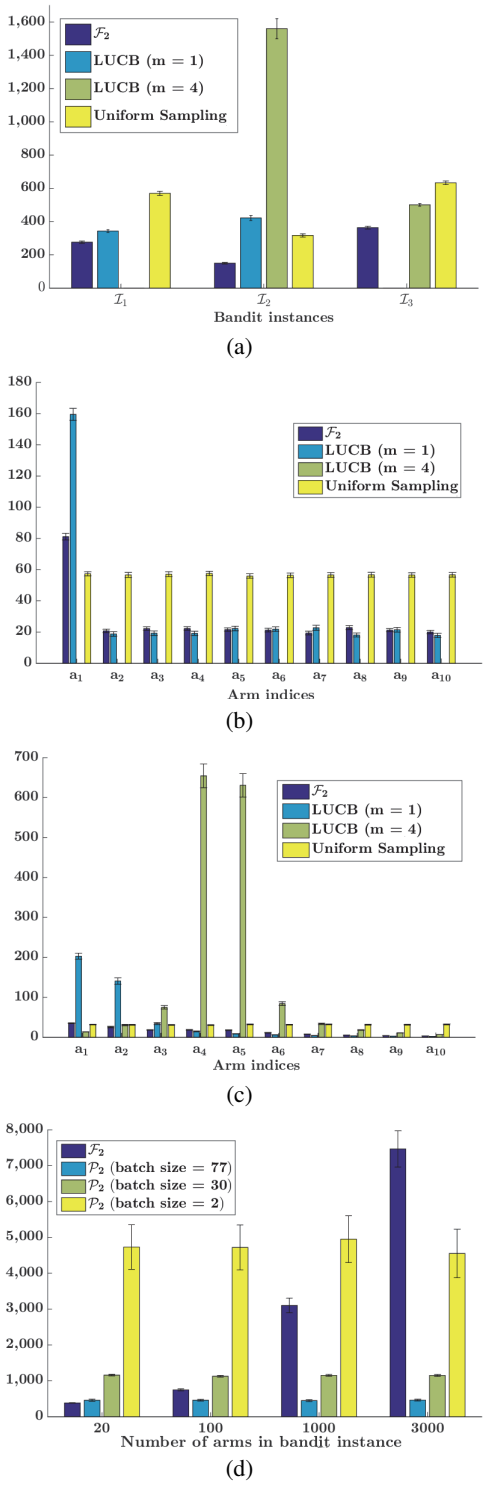

Figure 1: Sample complexity of $\mathcal{F}_2$, $\mathcal{P}_2$, LUCB (with different subset sizes) and uniform sampling. The y axis of each graph shows the sample complexity (averaged over 100 independent runs, with error bars corresponding to one standard error of the mean). Full details are provided in Section 5.

random. We present upper and lower bounds for both cases. We then adapt the fully sequential algorithm designed for a related problem (Kalyanakrishnan et al. 2012) to address ours. We furnish an instance-specific upper bound on the expected sample complexity that improves upon previous results. For infinite-armed bandits, we also demonstrate that whereas a PAC algorithm only requires finite memory, a larger memory-size can be used effectively to cut down the number of samples needed.

We also plan to compare our algorithms with ones that do make structural assumptions on the arms and rewards (Kleinberg, Slivkins, and Upfal 2008)—and especially identify the domains in which combining these contrasting approaches would be beneficial.

7 Acknowledgments

We are grateful to Peter Auer for guiding the derivation of a lower bound for Q-F. We are also thankful to the anonymous reviewers for their comments. Shivaram Kalyanakrishnan was partially supported by DST INSPIRE Faculty grant DST/INSPIRE/04/2013/001156.

References

Agrawal, R. 1995. The continuum-armed bandit problem. *SIAM J. Control Optim.* 33(6):1926–1951.

Amaran, S.; Sahinidis, N. V.; Sharda, B.; and Bury, S. J. 2015. Simulation optimization: a review of algorithms and applications. *Annals of Op. Res.* 1–30.

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best arm identification in multi-armed bandits. In *Proc. COLT 2010*, 41–53. Omnipress.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2003. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.* 32(1):48–77.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3):235–256.

Auer, P.; Ortner, R.; and Szepesvári, C. 2007. Improved rates for the stochastic continuum-armed bandit problem. In *Proc. COLT 2007*, 454–468. Springer.

Berry, D., and Fristedt, B. 1985. *Bandit Problems: Sequential Allocation of Experiments*. Monographs on Statistics and Applied Probability Series. Chapman & Hall.

Berry, D. A.; Chen, R. W.; Zame, A.; Heath, D. C.; and Shepp, L. A. 1997. Bandit Problems With Infinitely Many Arms. *The Annals of Statistics* 25(5):2103–2116.

Cappé, O.; Garivier, A.; Maillard, O.-A.; Munos, R.; and Stoltz, G. 2013. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics* 41(3):1516–1541.

Carpentier, A., and Valko, M. 2015. Simple regret for infinitely many armed bandits. In *Proc. ICML 2015*, 1133–1141. JMLR.

Chaudhuri, K.; Freund, Y.; and Hsu, D. J. 2009. A parameter-free hedging algorithm. In Bengio, Y.; Schuurmans, D.; Lafferty, J. D.; Williams, C. K. I.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc. 297–305.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2002. PAC bounds for multi-armed bandit and markov decision processes. In *In Fifteenth Annual Conference on Computational Learning Theory (COLT)*, 255–270.

Goschin, S.; Weinstein, A.; Littman, M. L.; and Chastain, E. 2012. Planning in reward-rich domains via PAC bandits. In *Proc. EWRL 2012*, volume 24 of *JMLR W & CP*, 25–42. JMLR.

Kalyanakrishnan, S., and Stone, P. 2010. Efficient selection of multiple bandit arms: Theory and practice. In *Proc. ICML 2010*, 511–518. Omnipress.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC subset selection in stochastic multi-armed bandits. In *Proc. ICML 2012*, 655–662. Omnipress.

Kaufmann, E., and Kalyanakrishnan, S. 2013. Information complexity in bandit subset selection. In *Proc. COLT 2013*, volume 30 of *JMLR W&CP*, 228–251. JMLR.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2008. Multi-armed bandits in metric spaces. In *Proc. STOC 2008*, 681–690. ACM.

Kleinberg, R. 2005. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17*, 697–704. MIT Press.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proc. WWW 2010*, 661–670. ACM.

Mannor, S.; Tsitsiklis, J. N.; Bennett, K.; and Cesa-bianchi, N. 2004. The sample complexity of exploration in the multi-armed bandit problem. *JMLR* 5:623–648.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* 58(5):527–535.

Szörényi, B.; Busa-Fekete, R.; Weng, P.; and Hüllermeier, E. 2015. Qualitative multi-armed bandits: A quantile-based approach. In *32nd International Conference on Machine Learning*, 1660–1668.

Wang, Y.; Audibert, J.-Y.; and Munos, R. 2008. Algorithms for infinitely many-armed bandits. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08. USA: Curran Associates Inc. 1729–1736.