PAC Identification of Many Good Arms in Stochastic Multi-Armed Bandits

Arghya Roy Chaudhuri [1] **Shivaram Kalyanakrishnan** [1]

Abstract

We consider the problem of identifying any k out of the best m arms in an n-armed stochastic multi-armed bandit; framed in the PAC setting, this particular problem generalises both the problem of "best subset selection" (Kalyanakrishnan & Stone, 2010) and that of selecting "one out of the best m" arms (Roy Chaudhuri & Kalyanakrishnan, 2017). We present a lower bound on the worst-case sample complexity for general k, and a fully sequential PAC algorithm, LUCB-k-m, which is more sample-efficient on easy instances. Also, extending our analysis to infinite-armed bandits, we present a PAC algorithm that is independent of n, which identifies an arm from the best ρ fraction of arms using at most an additive poly-log number of samples than compared to the lower bound, thereby improving over Roy Chaudhuri & Kalyanakrishnan (2017) and Aziz et al. (2018). The problem of identifying $k > 1$ distinct arms from the best ρ fraction is not always well-defined; for a special class of this problem, we present lower and upper bounds. Finally, through a reduction, we establish a relation between upper bounds for the "one out of the best ρ" problem for infinite instances and the "one out of the best m" problem for finite instances. We conjecture that it is more efficient to solve "small" finite instances using the latter formulation, rather than going through the former.

1. Introduction

The stochastic multi-armed bandit (Robbins, 1952; Berry & Fristedt, 1985) is a well-studied abstraction of decision making under uncertainty. Each *arm* of a bandit represents a decision. A *pull* of an arm represents taking the associated decision, which produces a real-valued reward. The reward is drawn i.i.d. from a distribution corresponding to the arm, independent of the pulls of other arms. At each round, the experimenter may consult the preceding history of pulls and rewards to decide which arm to pull.

The traditional objective of the experimenter is to maximise the expected cumulative reward over a horizon of pulls, or equivalently, to minimise the *regret* with respect to always pulling an optimal arm. Achieving this objective requires a careful balance between *exploring* (to reduce uncertainty about the arms' expected rewards) and *exploiting* (accruing high rewards). Regret-minimisation algorithms have been used in a variety of applications, including clinical trials (Robbins, 1952), adaptive routing (Awerbuch & Kleinberg, 2008), and recommender systems (Li et al., 2010).

Of separate interest is the problem of *identifying* an arm with the highest mean reward (Bechhofer, 1958; Paulson, 1964; Even-Dar et al., 2002), under what is called the "pure exploration" regime. For applications such as product testing (Audibert et al., 2010) and strategy selection (Goschin et al., 2012), there is a dedicated phase in the experiment in which the rewards obtained are inconsequential. Rather, the objective is to identify the best arm either (1) in the minimum number of trials, for a given confidence threshold (Even-Dar et al., 2002; Kalyanakrishnan & Stone, 2010), or alternatively, (2) with minimum error, after a given number of trials (Audibert et al., 2010; Carpentier & Valko, 2015). Our investigation falls into the first category, which is termed the "fixed confidence" setting. Conceived by Bechhofer (1958), best-arm-identification in the fixed confidence setting has received a significant amount of attention over the years (Even-Dar et al., 2002; Gabillon et al., 2011; Karnin et al., 2013; Jamieson et al., 2014). The problem has also been generalised to identify the best subset of arms (Kalyanakrishnan et al., 2012).

More recently, Roy Chaudhuri & Kalyanakrishnan (2017) have introduced the problem of identifying a single arm from among the best m in an n-armed bandit. This formulation is particularly useful when the number of arms is large, and in fact is a viable alternative even when the number of arms is *infinite*. In many practical scenarios, however, it is required to identify more than a single good arm. For example, imagine that a company needs to complete a task

[1] Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India. Correspondence to: Arghya Roy Chaudhuri <arghya@cse.iitb.ac.in>, Shivaram Kalyanakrishnan <shivaram@cse.iitb.ac.in>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

that is too large to be accomplished by a single worker, but which can be broken into 5 subtasks, each capable of being completed by one worker. Suppose there are a total of 1000 workers, and an indepdendent pilot survey has revealed that at least 15% of them have the skills to complete the subtask. To address the company's need, surely it would *suffice* to identify the best 5 workers for the subtask. However, if workers are to be identified based on a skill test that has stochastic outcomes, it would be unnecessarily expensive to indeed identify the "best subset". Rather, it would be enough to merely identify any 5 workers from among the best 150. This is precisely the problem we consider in our paper: identifying any k out of the best m arms in an n-armed bandit. In addition to distributed crowdsourcing (Tran-Thanh et al., 2014), applications of this problem include the management of large sensor networks (Mousavi et al., 2016), wherein multiple reliable sensors must be identified using minimal testing, and in drug design (Will et al., 2016, Chapter 43).

The problem assumes equal significance from a theoretical standpoint, since it generalises both the "best subset selection" problem (Kalyanakrishnan & Stone, 2010) (taking $k = m$) and that of selecting a "single arm from the best subset" (Roy Chaudhuri & Kalyanakrishnan, 2017) (taking $k = 1$). Unlike best subset selection, the problem remains feasible to solve even when n is large or infinite, as long as m/n is some constant $\rho > 0$. Traditionally, infinite-armed bandits have been tackled by resorting to side information such as distances between arms (Agrawal, 1995; Kleinberg, 2005) or the structure of their distribution of rewards (Wang et al., 2008). This approach introduces additional parameters, which might not be easy to tune in practice. Alternatively, good arms can be reached merely by selecting arms *at random* and testing them by pulling. This latter approach has been applied successfully both in the regret-minimisation setting (Herschkorn et al., 1996) and in the fixed-confidence setting (Goschin et al., 2012; Roy Chaudhuri & Kalyanakrishnan, 2017). Our formulation paves the way for identifying "many" (k) "good" (in the top m among n) arms in this manner.

The interested reader may refer to Table 1 right away for a summary of our theoretical results, which are explained in detail after formally specifying the (k, m, n) and (k, ρ) problems in Section 2. In Section 3 we present our algorithms and analysis for the finite setting, and in Section 4 for the infinite setting. We present experimental results in Section 5, and conclude with a discussion in Section 6.

2. Problem Definition and Contributions

Let $\mathcal{A}$ be the set of arms in our given bandit instance. With each arm $a \in \mathcal{A}$, there is an associated reward distribution supported on a subset of $[0, 1]$, with mean μ_a. When pulled, arm $a \in \mathcal{A}$ produces a reward drawn i.i.d. from the corresponding distribution, and independent of the pulls of other arms. At each round, based on the preceding sequence of pulls and rewards, an algorithm either decides which arm to pull, or stops and returns a set of arms.

For a finite bandit instance with n arms, we take $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, and assume, without loss of generality, that for arms $a_i, a_j \in \mathcal{A}$, $\mu_{a_i} \geq \mu_{a_j}$ whenever $i \leq j$. Given a tolerance $\epsilon \in [0, 1]$ and $m \in \{1, 2, \ldots, n\}$, we call an arm $a \in \mathcal{A}$ (ϵ, m)-optimal if $\mu_a \geq \mu_{a_m} - \epsilon$. We denote the set of all the (ϵ, m)-optimal arms as $\mathcal{TOP}_m(\epsilon) \overset{\text{def}}{=} \{a : \mu_a \geq \mu_{a_m} - \epsilon\}$. For brevity we denote $\mathcal{TOP}_m(0)$ as $\mathcal{TOP}_m$. Now, we introduce (k, m, n) problem.

(k, m, n) **Problem.** An instance of the (k, m, n) problem is of the form $(\mathcal{A}, n, m, k, \epsilon, \delta)$, where $\mathcal{A}$ is a set of arms with $|\mathcal{A}| = n \geq 2$; $m \in \{1, 2, \ldots, n-1\}$; $k \in \{1, \ldots, m\}$; tolerance $\epsilon \in (0, 1]$; and mistake probability $\delta \in (0, 1]$. An algorithm $\mathcal{L}$ is said to solve (k, m, n) if for every instance of (k, m, n), it terminates with probability 1, and returns k *distinct* (ϵ, m)-optimal arms with probability at least $1 - \delta$.

The (k, m, n) problem is interesting from a theoretical standpoint because it covers an entire range of problems, with single-arm identification ($m = 1$) at one extreme and subset selection ($k = m$) at the other. We note that the Q-F (Roy Chaudhuri & Kalyanakrishnan, 2017) problem is identical to $(1, m, n)$, and SUBSET (Roy Chaudhuri & Kalyanakrishnan, 2017) is identical to (m, m, n). Thus, any bounds on the sample complexity of (k, m, n) also apply to Q-F (Roy Chaudhuri & Kalyanakrishnan, 2017) and to SUBSET (Kalyanakrishnan & Stone, 2010). In this paper, we show that any algorithm that solves (k, m, n) must incur $\Omega\left(\frac{n}{(m-k+1)\epsilon^2} \log\left(\frac{\binom{m}{k-1}}{\delta}\right)\right)$ pulls for some instance of the problem. We are unaware of bounds in the fixed-confidence setting that involve such a combinatorial term inside the logarithm.

Table 1 places our bounds in the context of previous results. The bounds shown in the table consider the worst-case across problem instances; in practice one can hope to do better on easier problem instances by adopting a fully sequential sampling strategy. Indeed we adapt the LUCB1 algorithm (Kalyanakrishnan et al., 2012) to solve (k, m, n), denoting the new algorithm LUCB-k-m. Our analysis shows that for $k = 1$, and $k = m$, the upper bound on the sample complexity of this algorithm matches with those of $\mathcal{F}_2$ (Roy Chaudhuri & Kalyanakrishnan, 2017) and LUCB1 (Kalyanakrishnan et al., 2012), respectively, up to a multiplicative constant. Empirically, LUCB-k-m with $k = 1$ appears to be more efficient than $\mathcal{F}_2$ for solving Q-F.

Along the same lines that Roy Chaudhuri & Kalyanakrishnan (2017) define the Q-P problem for infinite instances, we define a generalisation of Q-P for selecting many good arms,

Table 1. Lower and upper bounds on the expected sample complexity (worst case over problem instances). The bounds for (k, ρ), $k > 1$ are for the special class of "at most k-equiprobable" instances.

Problem	Lower Bound	Previous Upper Bound	Current Upper Bound
$(1, 1, n)$ Best-Arm	$\Omega\left(\frac{n}{\epsilon^2} \log \frac{1}{\delta}\right)$ (Mannor & Tsitsiklis, 2004)	$O\left(\frac{n}{\epsilon^2} \log \frac{1}{\delta}\right)$ (Even-Dar et al., 2002)	Same as previous
(m, m, n) SUBSET	$\Omega\left(\frac{n}{\epsilon^2} \log \frac{m}{\delta}\right)$ (Kalyanakrishnan et al., 2012)	$O\left(\frac{n}{\epsilon^2} \log \frac{m}{\delta}\right)$ (Kalyanakrishnan & Stone, 2010)	Same as previous
$(1, m, n)$ Q-F	$\Omega\left(\frac{n}{m\epsilon^2} \log \frac{1}{\delta}\right)$ (Roy Chaudhuri & Kalyanakrishnan, 2017)	$O\left(\frac{n}{m\epsilon^2} \log^2 \frac{1}{\delta}\right)$	$O\left(\frac{1}{\epsilon^2}\left(\frac{n}{m} \log \frac{1}{\delta} + \log^2 \frac{1}{\delta}\right)\right)$ **This paper**
(k, m, n) Q-F$_k$	$\Omega\left(\frac{n}{(m-k+1)\epsilon^2} \log \frac{\binom{m}{k-1}}{\delta}\right)$ **This paper**	-	$O\left(\frac{k}{\epsilon^2}\left(\frac{n \log k}{m} \log \frac{k}{\delta} + \log^2 \frac{k}{\delta}\right)\right)^*$ **This paper** (*for $k \geq 2$)
$(1, \rho)$ $(\lvert\mathcal{A}\rvert = \infty)$ Q-P	$\Omega\left(\frac{1}{\rho\epsilon^2} \log \frac{1}{\delta}\right)$ (Roy Chaudhuri & Kalyanakrishnan, 2017)	$O\left(\frac{1}{\rho\epsilon^2} \log^2 \frac{1}{\delta}\right)$	$O\left(\frac{1}{\epsilon^2}\left(\frac{1}{\rho} \log \frac{1}{\delta} + \log^2 \frac{1}{\delta}\right)\right)$ **This paper**
(k, ρ) $(\lvert\mathcal{A}\rvert = \infty)$ Q-P$_k$	$\Omega\left(\frac{k}{\rho\epsilon^2} \log \frac{k}{\delta}\right)$ **This paper**	-	$O\left(\frac{k}{\epsilon^2}\left(\frac{\log k}{\rho} \log \frac{k}{\delta} + \log^2 \frac{k}{\delta}\right)\right)^*$ **This paper** (*for a special class with $k \geq 2$)

which we denote (k, ρ). Given a set of arms $\mathcal{A}$, a sampling distribution $P_{\mathcal{A}}$, $\epsilon \in (0, 1]$, and $\rho \in [0, 1]$, an arm $a \in \mathcal{A}$ is called $[\epsilon, \rho]$-optimal if $P_{a' \sim P_{\mathcal{A}}}\{\mu_a \geq \mu_{a'} - \epsilon\} \geq 1 - \rho$. For $\rho, \epsilon \in [0, 1]$, we define the set of all $[\epsilon, \rho]$-optimal arms as $\mathcal{TOP}_\rho(\epsilon)$, and we denote $\mathcal{TOP}_\rho(0)$ as $\mathcal{TOP}_\rho$. We recall the definition of Q-P from Roy Chaudhuri & Kalyanakrishnan (2017), and then generalise it to (k, ρ).

Q-P Problem (Roy Chaudhuri & Kalyanakrishnan, 2017). 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 $1 - \delta$.

(k, ρ) **Problem.** An instance of (k, ρ) problem is of the form $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta)$, where $\mathcal{A}$ is a set of arms; $P_{\mathcal{A}}$ is a probability distribution over $\mathcal{A}$; quantile fraction $\rho \in (0, 1]$; tolerance $\epsilon \in (0, 1]$; and mistake probability $\delta \in (0, 1]$. Such an instance is *valid* if $\lvert\mathcal{TOP}_\rho\rvert \geq k$, and *invalid* otherwise. An algorithm $\mathcal{L}$ is said to solve (k, ρ), if for every *valid* instance of (k, ρ), $\mathcal{L}$ terminates with probability 1, and returns k *distinct* $[\epsilon, \rho]$-optimal arms with probability at least $1 - \delta$.

At most k-equiprobable instances. Observe that (k, ρ) is well-defined only if the given instance has at least k distinct arms in $\mathcal{TOP}_\rho$; we term such an instance *valid*. It is worth noting that even valid instances can require an arbitrary amount of computation to solve. For example, consider an instance with $k > 1$ arms in $\mathcal{TOP}_\rho$, one among which has a probability γ of being picked by $P_{\mathcal{A}}$, and the rest each a probability of $(\rho - \gamma)/(k - 1)$. Since the arms have to be identified by sampling from $P_{\mathcal{A}}$, the probability of identifying the latter $k - 1$ arms diminishes to 0 as $\gamma \to \rho$, calling for an infinite number of guesses. To avoid this scenario, we restrict our analysis to a special class of valid instances in which $P_{\mathcal{A}}$ allocates no more than ρ/k probability to any arm in $\mathcal{TOP}_\rho$. We refer to such instances as "at most k-equiprobable" instances. Formally, a (k, ρ) problem instance given by $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta)$ is called "at most k-equiprobable" if $\forall a \in \mathcal{TOP}_\rho$, $\Pr_{\mathbf{a}' \sim P_{\mathcal{A}}}\{\mathbf{a}' = a\} \leq \frac{\rho}{k}$.[1]

Note that any instance of the $(1, \rho)$ or Q-P (Roy Chaudhuri & Kalyanakrishnan, 2017) problem is necessarily valid and at most 1-equiprobable. Interestingly, we improve upon the existing upper bound for this problem, so it matches the lower bound up to an *additive* $O\left(\frac{1}{\epsilon^2} \log^2 \frac{1}{\delta}\right)$ term. Below we summarise our contributions.

We generalise two previous problems—Q-F and SUBSET (Roy Chaudhuri & Kalyanakrishnan, 2017)—via (k, m, n). In Section 3 we derive a lower bound on the worst case sample complexity to solve (k, m, n), which generalises existing lower bounds for Q-F and SUBSET. Further, we In Section 3.2 we present a fully-sequential algorithm—*LUCB for k out of m* and establish an upper-bound on the sample complexity. In Section 4 we present algorithm $\mathcal{P}_3$ to solve Q-P with a sample complexity that is an additive $O((1/\epsilon^2) \log^2(1/\delta))$ term away from the lower bound. We extend it to an algorithm KQP-1 for solving at most k-equiprobable (k, ρ) instances. Also, $\mathcal{P}_3$ and KQP-1 can solve Q-F and (k, m, n) respectively, and their sample complexities are the tightest instance-independent upper bounds as yet. In Section 4.3 we present a general relation between the upper bound on the sample complexities for solving Q-F and Q-P. This helps in effectively transferring

[1] In a recent paper, Ren et al. (2018) claim to solve the (k, ρ) problem. However, they do not notice that the problem can be ill-posed. Also, even with an at most k-equiprobable instance as input, their algorithm fails to escape the $(1/\rho) \log^2(1/\delta)$ dependence.

any improvement in the upper bound on the former to the latter. Also, we conjecture the existence of a class of Q-F instances that can be solved more efficiently than their "corresponding" Q-P instances.

3. Algorithms for Finite Instances

We begin our technical presentation by furnishing a lower bound on the sample complexity of algorithms for (k, m, n).

3.1. Lower Bound on the Sample-Complexity

Theorem 3.1. *[Lower Bound for (k, m, n)] Let $\mathcal{L}$ be an algorithm that solves (k, m, n). Then, there exists an instance $(\mathcal{A}, n, m, k, \epsilon, \delta)$, with $0 < \epsilon \le \frac{1}{\sqrt{32}}$, $0 < \delta \le \frac{e^{-1}}{4}$, and $n \ge 2m$, $1 \le k \le m$, on which the expected number of pulls performed by $\mathcal{L}$ is at least $\frac{1}{18375} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m-k+1} \ln \frac{\binom{m}{k-1}}{4\delta}$.*

The proof, given in Appendix A, generalises the lower bound proofs for both (m, m, n) (Kalyanakrishnan et al., 2012, Theorem 8) and $(1, m, n)$ (Roy Chaudhuri & Kalyanakrishnan, 2017, Theorem 3.3). The core idea in these proofs is to consider two sets of bandit instances, $\mathcal{I}$ and $\mathcal{I}'$, such that over "short" trajectories, an instance from $\mathcal{I}$ will yield the same reward sequences as a corresponding instance from $\mathcal{I}'$, with high probability. Thus, any algorithm will return the same set of arms for both instances, with high probability. However, by construction, no set of arms can be simultaneously correct for both instances—implying that a correct algorithm must encounter sufficiently "long" trajectories. Our main contribution is in the design of $\mathcal{I}$ and $\mathcal{I}'$ when $k \in \{1, 2, \ldots, m\}$ (rather than exactly 1 or m) arms have to be returned.

Our algorithms to achieve improved *upper* bounds for Q-F and (k, m, n) (across bandit instances) follow directly from methods we design for the infinite-armed setting in Section 4 (see Corollary 4.2 and Corollary 4.5). In the remainder of this section, we present a fully-sequential algorithm for (k, m, n) whose expected sample complexity varies with the "hardness" of the input instance.

3.2. An Adaptive Algorithm for Solving (k, m, n)

Algorithm 1 describes LUCB-k-m, a fully sequential algorithm that generalises LUCB1 (Kalyanakrishnan et al., 2012), which solves (m, m, n). For $k = 1$ LUCB-k-m has the same guarantee on sample-complexity as $\mathcal{F}_2$ (Roy Chaudhuri & Kalyanakrishnan, 2017), but empirically appears to be more economical.

Algorithm 1 LUCB-k-m: Algorithm to select k (ϵ, m)-optimal arms

```
Input: $\mathcal{A}$ (such that $|\mathcal{A}| = n$), $k, m, \epsilon, \delta$.
Output: $k$ distinct $(\epsilon, m)$-optimal arms from $\mathcal{A}$.
  Pull each arm $a \in \mathcal{A}$ once. Set $t = n$.
  while $ucb(l^t_*, t+1) - lcb(h^t_*, t+1) > \epsilon$. do
    $t = t + 1$.
    $A^t_1 \overset{\text{def}}{=}$ Set of $k$ arms with the highest empirical means.
    $A^t_3 \overset{\text{def}}{=}$ Set of $n - m$ arms with the lowest empirical means.
    $A^t_2 \overset{\text{def}}{=} \mathcal{A} \setminus (A^t_1 \cup A^t_3)$.
    $h^t_* = \arg\max_{\{a \in A^t_1\}} lcb(a, t)$.
    $m^t_* = \arg\min_{\{a \in A^t_2\}} u^t_a$.
    $l^t_* = \arg\max_{\{a \in A^t_3\}} ucb(a, t)$.
    pull $h^t_*, m^t_*, l^t_*$.
  end while
  return $A^t_1$.
```

Under LUCB-k-m, at each round t, we partition $\mathcal{A}$ into three subsets. We keep the k arms with the highest empirical averages in A^t_1, the $n - m$ arms with the lowest empirical averages in A^t_3, and the rest in A^t_2; ties are broken arbitrarily (uniformly at random in our experiments). Letting u^t_a be the number of pulls obtained by the arm a until the horizon $t - 1$, and $\beta(a, t) \overset{\text{def}}{=} \sqrt{\frac{1}{2u^t_a} \ln\left(\frac{knt^4}{\delta}\right)}$ (with $k = 5/4$), we define the upper confidence bound (UCB) and the lower confidence bound (LCB) on the true mean of the arms as $ucb(a, t) = \hat{p}^t_a + \beta(a, t)$, and $lcb(a, t) = \hat{p}^t_a - \beta(a, t)$, respectively. At each round we choose a *contentious* arm from each of these three sets: from A^t_1 we choose h^t_*, the arm with the lowest LCB; from A^t_2 the arm which is least pulled is chosen, and called m^t_*; from A^t_3 we choose l^t_*, the arm with the highest UCB. The algorithm stops as soon as the difference between the LCB of h^t_*, and the UCB of l^t_* is no larger than the tolerance ϵ.

Let B_1, B_2, B_3 be corresponding sets based on the true means: that is, subsets of $\mathcal{A}$ such that $B_1 \overset{\text{def}}{=} \{1, 2, \cdots, k\}$, $B_2 \overset{\text{def}}{=} \{k+1, k+2, \cdots, m\}$ and $B_3 \overset{\text{def}}{=} \{m+1, m+2, \cdots, n\}$. For any two arms $a, b \in \mathcal{A}$ we define $\Delta_{ab} \overset{\text{def}}{=} \mu_a - \mu_b$, and overloading the notation, define

$$\Delta_a \overset{\text{def}}{=} \begin{cases} \mu_a - \mu_{m+1} & \text{if } a \in B_1 \\ \mu_k - \mu_{m+1} & \text{if } a \in B_2 \\ \mu_m - \mu_a & \text{if } a \in B_3. \end{cases} \tag{1}$$

We note that $\Delta_k = \Delta_{k+1} = \cdots = \Delta_m = \Delta_{m+1}$. Now, we define the hardness term as $H_\epsilon \overset{\text{def}}{=} \sum_{a \in \mathcal{A}} \frac{1}{\max\{\Delta_a, \epsilon/2\}^2}$.

Theorem 3.2. *[Expected Sample Complexity of* LUCB*-k-m]* LUCB*-k-m solves (k, m, n) using an expected sample complexity upper-bounded by $O\left(H_\epsilon \log \frac{H_\epsilon}{\delta}\right)$.*

Appendix-B gives the proof in detail. The core argument resembles that of Algorithm $\mathcal{F}_2$ (Roy Chaudhuri & Kalyanakrishnan, 2017). However, it subtly differs due to the different strategy for choosing arms and since the output set need not be singleton. In practice, one can use tighter confidence bound calculations for even more efficiency; we use KL-divergence based confidence bounds in our experiments.

4. Algorithm for Infinite Instances

In this section, we present algorithms for infinite-armed bandit instances. To find a single $[\epsilon, \rho]$-optimal arm, the sample complexity of existing algorithms (Roy Chaudhuri & Kalyanakrishnan, 2017; Aziz et al., 2018) scales as $(1/\rho)\log^2(1/\delta)$, for the given mistake probability δ. Here we present an algorithm $\mathcal{P}_3$ whose sample complexity is only an *additive* poly-log factor away from the lower bound of $\Omega((1/\rho\epsilon^2)\log 1/\delta)$ (Roy Chaudhuri & Kalyanakrishnan, 2017, Corollary 3.4). Note that Jamieson et al. (2016) solve a related, but different, problem in which $\mathcal{A}$, even if infinite, has only two possible values for the means of its arms.

4.1. Solving Q-P Instances

Roy Chaudhuri & Kalyanakrishnan (2017) presented an algorithm $\mathcal{P}_2$ to solve Q-P instances. It consists of choosing $N(\rho, \delta)$ arms from $\mathcal{A}$, followed by identifying the best arm with PAC guarantee. $\mathcal{P}_3$ is a two-phase algorithm. In the first phase, it runs a sufficiently large number of independent copies of $\mathcal{P}_2$ and chooses a large subset of arms (say of size u), in which every arm is $[\epsilon, \rho]$-optimal with probability at least $1 - \delta'$, where δ' is some small *constant*. The value u is chosen in a manner such that at least one of the chosen arms is $[\epsilon/2, \rho]$-optimal with probability at least $\delta/2$. The second phase solves the best arm identification problem $(1, 1, u)$ by applying MEDIAN ELIMINATION.

Algorithm 2 describes $\mathcal{P}_3$. It uses $\mathcal{P}_2$ (Roy Chaudhuri & Kalyanakrishnan, 2017) with MEDIAN ELIMINATION as a subroutine, to select an $[\epsilon, \rho]$-optimal arm with confidence $1 - \delta'$. We have assumed $\delta' = 1/4$, in practice the one can choose any sufficiently small value for it, which will merely affect the multiplicative constant in the upper bound.

Algorithm 2 $\mathcal{P}_3$: Algorithm to solve Q-P

```
Input: $\mathcal{A}, \epsilon, \delta$.
Output: One $[\epsilon, \rho]$-optimal arm.
  Set $\delta' = 1/4$, $u = \lceil (1/\delta') \log(2/\delta) \rceil = \lceil 4 \log(2/\delta) \rceil$.
  Run $u$ copies of $\mathcal{P}_2(\mathcal{A}, \rho, \epsilon/2, \delta')$ and store outputs in set $S$.
  Identify an $(\epsilon/2, 1)$-optimal arm in $S$ using MEDIAN ELIMINATION with confidence at least $1 - \delta/2$.
```

Theorem 4.1. *[Correctness and Sample Complexity of $\mathcal{P}_3$] $\mathcal{P}_3$ solves* Q-P*, with sample complexity* $O(\epsilon^{-2}(\rho^{-1}\log(1/\delta) + \log^2(1/\delta)))$.

Proof. First we prove the correctness and then upper-bound the sample complexity.

Correctness. We notice that each copy of $\mathcal{P}_2$ outputs an $[\epsilon/2, \rho]$-optimal arm with probability at least $1 - \delta'$. Now, $S \cap \mathcal{TOP}_\rho = \emptyset$ can only happen if all the u copies of $\mathcal{P}_2$ output sub-optimal arms. Therefore, $\Pr\{S \cap \mathcal{TOP}_\rho = \emptyset\} = (1 - \delta')^u \leq \delta/2$. On the other hand, the mistake probability of MEDIAN ELIMINATION is upper bounded by $\delta/2$. Therefore, by taking union bound, we upper bound the mistake probability by δ. Also, the mean of the output arm is not less than $\frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$ from the $(1 - \rho)$-th quantile.

Sample complexity. First we note that, for some appropriate constant C, the sample complexity (SC) of each of the u copies of $\mathcal{P}_2$ is $\frac{C}{\rho(\epsilon/2)^2}\left(\ln \frac{2}{\delta'}\right)^2 \in O\left(\frac{1}{\rho\epsilon^2}\right)$. Hence, SC of all the u copies $\mathcal{P}_2$ together is upper bounded by $\frac{C_1 \cdot u}{\rho\epsilon^2}$, for some constant C_1. Also, for some constants C_2, C_3, the sample complexity of MEDIAN ELIMINATION is upper-bounded by $\frac{C_2 \cdot u}{(\epsilon/2)^2} \ln \frac{2}{\delta} \leq \frac{C_3}{\epsilon^2} \ln^2 \frac{2}{\delta}$. Adding the sample complexities and substituting for u yields the bound. □

Corollary 4.2. *$\mathcal{P}_3$ can solve any instance of* Q-F $(\mathcal{A}, n, m, \epsilon, \delta)$ *with sample complexity* $O\left(\frac{1}{\epsilon^2}\left(\frac{n}{m}\log\frac{1}{\delta} + \log^2\frac{1}{\delta}\right)\right)$.

Proof. Let, $(\mathcal{A}, n, m, \epsilon, \delta)$ be the given instance of Q-F. We partition the set $\mathcal{A}^\infty = [0, 1]$ in to n equal segments and associate each with a unique arm in $\mathcal{A}$, and such that no two different subsets get associated with the same arm. Now, defining $P_{\mathcal{A}^\infty} = Uniform[0, 1]$, and $\rho' = m/n$, we realise that solving the Q-P instance $(\mathcal{A}^\infty, P_{\mathcal{A}^\infty}, \rho', \epsilon, \delta)$ solves the original Q-F instance, thereby proving the corollary. □

At this point it is of natural interest to find an efficient algorithm to solve (k, ρ). Next, we discuss the extension of Q-P to (k, ρ), and present lower and upper bounds on the sample complexity needed to solve it.

4.2. Solving "At Most k-equiprobable" (k, ρ) Instances

Now, let us focus on identifying k $[\epsilon, \rho]$-optimal arms. In Theorem 4.3 we derive the lower bound on the sample complexity to solve an instance (k, ρ) by reducing it to solving a SUBSET problem as follows.

Theorem 4.3. *[Lower Bound on the Sample Complexity for Solving (k, ρ)] For every $\epsilon \in (0, \frac{1}{\sqrt{32}}]$, $\delta \in (0, \frac{1}{\sqrt{32}}]$, and $\rho \in (0, \frac{1}{2}]$, there exists an instance of (k, ρ) given by $(\mathcal{A}, P_\mathcal{A}, \rho, \epsilon, \delta)$, such that any algorithm that solves (k, ρ) incurs at least $C \cdot \frac{k}{\rho\epsilon^2} \ln \frac{k}{8\delta}$ samples, where $C = \frac{1}{18375}$.*

Proof. We shall prove the theorem by contradiction. Let us assume that the statement is incorrect. Therefore, there exists an algorithm ALG that can solve any instance of (k, ρ) using no more than $C \cdot \frac{k}{\rho\epsilon^2} \ln \frac{k}{8\delta}$ samples, for $C = 1/18375$. Now, let $(n, \mathcal{A}, m, \epsilon, \delta)$ be an instance of SUBSET (Roy Chaudhuri & Kalyanakrishnan, 2017) with $n \geq 2m$. Letting $P_\mathcal{A} = Uniform\{1, 2, \ldots, n\}$, $k = m$, and $\rho = m/n$, we create an instance of (k, ρ) as $(\mathcal{A}, P_\mathcal{A}, \rho, k, \epsilon, \delta)$. Therefore, solving this (k, ρ) instance will solve the original SUBSET instance. According our claim, ALG solves the original SUBSET instance using at most $C \cdot \frac{k}{(k/n)\epsilon^2} \ln \frac{k}{8\delta} = C \cdot \frac{m}{(m/n)\epsilon^2} \ln \frac{m}{8\delta} = C \cdot \frac{n}{\epsilon^2} \ln \frac{m}{8\delta}$ samples. This observation contradicts the lower bound on the sample complexity for solving SUBSET (Kalyanakrishnan et al., 2012, Theorem 8); thereby proving the theorem. □

Solving at most k-equiprobable (k, ρ) instances. Let, for any $\mathcal{S} \subseteq \mathcal{A}$, $\nu(\mathcal{S}) \overset{\text{def}}{=} \Pr_{a \sim P_{\mathcal{A}}}\{a \in \mathcal{S}\}$. Therefore, $\nu(\mathcal{A}) = 1$. Now, we present an algorithm KQP-1 that can solve any at most k-equiprobable instance of (k, ρ). Algorithm 3 describes KQP-1. At each phase y, it solves an instance of Q-P to output an arm, say $a^{(y)}$, from $\mathcal{TOP}_\rho(\epsilon)$. In the next phase, it updates the bandit instance $\mathcal{A}^{y+1} = \mathcal{A}^y \setminus \{a^{(y)}\}$, the sampling distribution $P_{\mathcal{A}^{y+1}} = \frac{1}{1-\nu(\mathcal{A} \setminus \mathcal{A}^{y+1})} P_{\mathcal{A}^y}$, and the target quantile $\rho - \sum_{j=1}^{y} \nu(a^{(j)})$. However, as we are not given the explicit form of $P_{\mathcal{A}}$, we realise $P_{\mathcal{A}^{y+1}}$ by rejection-sampling—if $a' \in \mathcal{A} \setminus \mathcal{A}^{y+1}$ is chosen by $P_{\mathcal{A}}$, we simply discard a', and continue to sample $P_{\mathcal{A}}$ one more time. Because $\nu(\{a^y\})$ is not known explicitly, we rely on the fact that $\nu(\{a^y\}) \leq \rho/k$: it is for this reason we require the instance to be at most k-equiprobable. Therefore, in each phase $y \geq 2$, we update $\rho^y = \rho^{y-1} - \rho/k \leq \rho - \sum_{j=1}^{y-1} \nu\{a^j\}$, with $\rho^1 = \rho$. Hence, at the phase $y \geq 1$, KQP-1 solves an instance of Q-P given by $(\mathcal{A}^y, P_{\mathcal{A}^y}, \rho - (y-1)\rho/k, \epsilon, \delta/k)$.

Algorithm 3 KQP-1: Algorithm to solve an at most k-equiprobable (k, ρ) instances

```
Input: $\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta$.
Output: Set of $k$ distinct arms from $\mathcal{TOP}_\rho(\epsilon)$.
  $\mathcal{A}^1 = \mathcal{A}$.
  for $y = 1, 2, 3, \cdots, k$ do
    $\rho^y = \rho - (y-1)\rho/k$.
    Run $\mathcal{P}_3$ to solve the Q-P instance given by
    $(\mathcal{A}^y, P_{\mathcal{A}^y}, \rho^y, \epsilon, \frac{\delta}{k})$, and let $a^{(y)}$ be the output.
    $\mathcal{A}^{y+1} = \mathcal{A}^y \setminus \{a^{(y)}\}$.
  end for
```

Theorem 4.4. *Given any at most k-equiprobable instance of (k, ρ) with $k > 1$, KQP-1 solves the instance with expected sample-complexity upper-bounded by $O\left(\frac{k}{\epsilon^2}\left(\frac{\log k}{\rho}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right)\right)$.*

Proof. We prove correctness and then establish the sample complexity upper bound.

Correctness: Letting E_y be the event that $a^{(y)} \notin \mathcal{TOP}_\rho(\epsilon)$, the probability of mistake by KQP-1 can be upper bounded as $\Pr\{\text{Error}\} = \Pr\{\exists y \in \{1, \cdots, k\}\ E_y\} \leq \sum_{y=1}^{k} \Pr\{E_y\} \leq \sum_{y=1}^{k} \delta/k = \delta$.

Sample complexity: In phase y, the sample complexity of $\mathcal{P}_3$ is upper-bounded as $\mathrm{SC}(y) \leq C\epsilon^{-2}((1/\rho^y)\log\frac{k}{\delta} + \log^2\frac{k}{\delta})$, for some constant C. Therefore, the sample complexity of KQP-1 is

$$\begin{aligned}
\sum_{y=1}^{k}\mathrm{SC}(y) &\leq \sum_{y=1}^{k}\frac{C}{\epsilon^2}\left(\frac{1}{\rho^y}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right), \\
&\leq \frac{C}{\epsilon^2}\left(\log\frac{k}{\delta}\sum_{y=1}^{k}\frac{1}{\rho - (y-1)\frac{\rho}{k}} + k\log^2\frac{k}{\delta}\right), \\
&= \frac{Ck}{\epsilon^2}\left(\frac{1}{\rho}\log\frac{k}{\delta}\sum_{y=1}^{k}\frac{1}{k-y+1} + \log^2\frac{k}{\delta}\right), \\
&\leq \frac{C'k}{\epsilon^2}\left(\frac{\log k}{\rho}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right),
\end{aligned}$$

for $k > 1$, and some constant C'. □

Corollary 4.5. KQP-*1 can solve any instance of (k, m, n) given by $(\mathcal{A}, n, m, k, \epsilon, \delta)$ with $k \geq 2$, using $O\left(\frac{k}{\epsilon^2}\left(\frac{n\log k}{m}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right)\right)$ samples.*

We note that the sample complexity of KQP-1 is independent of the size of $\mathcal{A}$, and every instance of (k, m, n) given by $(\mathcal{A}, n, m, m, \epsilon, \delta)$, can be solved by KQP-1 by posing it as an instance of (k, ρ) given by $(\mathcal{A}, Uniform\{\mathcal{A}\}, m/n, m, \epsilon, \delta)$. However, for $k = m$, the sample complexity of KQP-1 reduces to $O\left(\frac{1}{\epsilon^2}\left(n\log m \cdot \log\frac{m}{\delta} + \log^2\frac{m}{\delta}\right)\right)$, which is higher than the sample complexity of HALVING (Kalyanakrishnan & Stone, 2010), that needs only $O\left(\frac{n}{\epsilon^2}\log\frac{m}{\delta}\right)$ samples. Hence, for the best subset selection problem in finite instances HALVING is preferable to KQP-1. In very large instances, where the probability of picking any given arm from $\mathcal{TOP}_\rho$ is small, solving (k, ρ) using KQP-1 is more efficient. The following corollary considers a common special case, for which a slightly tighter bound applies.

Corollary 4.6. *Every instance of (k, ρ) given by $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta)$, such that $|\mathcal{A}| = \infty$, and for all finite subset $S \subset \mathcal{A}$, $\Pr_{a \sim P_{\mathcal{A}}}\{a \in S\} = 0$; can be solved with sample complexity $O\left(k\epsilon^{-2}\left(\rho^{-1}\log(k/\delta) + \log^2(k/\delta)\right)\right)$, by independently solving k different* Q-P *instances, each given by $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta/k)$.*

The correctness of Corollary 4.6 gets proved by noticing that all the k outputs are unique with probability 1, and then taking a union bound over mistake probabilities.

4.3. On the Hardness of Solving Q-P

Theorem 4.7 presents a general relation between the upper bound on sample complexities for solving Q-F and Q-P.

Theorem 4.7. *Let $\gamma : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \mapsto \mathbb{R}^+$. If every instance of* Q-F *given by $(\mathcal{A}, n, m, \epsilon, \delta)$, can be solved with sample complexity $O\left(\frac{n}{m\epsilon^2}\log\frac{1}{\delta} + \gamma(n, m, \epsilon, \delta)\right)$, then, every instance of* Q-P *given by $(\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta)$ can be solved with sample complexity $O\left((1/\rho\epsilon^2)\log(1/\delta) + \gamma\left(\lceil 8\log(2/\delta)\rceil, \lfloor 4\log(2/\delta)\rfloor, \epsilon/2, \delta/2\right)\right)$.*

We assume that there exists an algorithm OptQF that solves every instance of Q-F given by $(\mathcal{A}, n, m, \epsilon, \delta)$, using $O\left(\frac{n}{m\epsilon^2}\log\frac{1}{\delta} + \gamma(n, m, \epsilon, \delta)\right)$ samples. We establish the upper bound on sample complexity for solving Q-P by constructing an algorithm OptQP that follows an approach similar to $\mathcal{P}_3$. Specifically, OptQP reduces the input Q-P instance to an instance of Q-F using $O\left(\frac{1}{\rho\epsilon^2}\log\frac{1}{\delta}\right)$ samples. Then, it solves that Q-F using OptQF as the subroutine. The detailed proof is given in Appendix-C.

Corollary 4.8. *Corollary 4.2 shows that every* Q-F *is solvable in* $O\left(\frac{1}{\epsilon^2}\left(\frac{n}{m}\log\frac{1}{\delta} + \log^2\frac{1}{\delta}\right)\right)$ *samples. Hence,* $\gamma(n, m, \epsilon, \delta) \in O\left(\frac{1}{\epsilon^2}\log^2\frac{1}{\delta}\right)$*, and therefore, every* Q-P *is solvable in* $O\left(\frac{1}{\epsilon^2}\left(\frac{1}{\rho}\log\frac{1}{\delta} + \log^2\frac{1}{\delta}\right)\right)$ *samples.*

On the other hand, if the lower bound for solving Q-F *provided by Roy Chaudhuri & Kalyanakrishnan (2017) matches the upper bound up to a constant factor, then* $\gamma(n, m, \epsilon, \delta) \in \Theta\left(\frac{n}{m\epsilon^2}\log\frac{1}{\delta}\right)$*. In that case,* Q-P *is solvable using* $\Theta\left(\frac{1}{\rho\epsilon^2}\log\frac{1}{\delta}\right)$ *samples.*

It is interesting to find a $\gamma(\cdot)$ such that the upper bound presented in Theorem 4.7 matches the lower bound up to a constant factor. We notice, Theorem 4.7 guarantees that there exists a constant C, such that for any given ϵ, δ, and $m \leq n/2$, $\gamma(n, m, \epsilon, \delta) \leq C \cdot \gamma\left(\lceil 8\log(2/\delta)\rceil, \lfloor 4\log(2/\delta)\rfloor, \frac{\epsilon}{2}, \frac{\delta}{2}\right)$. However, for $n < \lceil 8\log(2/\delta)\rceil$ we believe Q-F can be solved more efficiently than by reducing it to Q-P. Considering a set of functions, $\mathcal{U} \overset{\text{def}}{=} \{f : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \mapsto \mathbb{R}^+\}$, we present a related conjecture.

Definition For $g \in \mathcal{U}$, Q-F is solvable in $\Theta(g(\cdot))$ if there exists an algorithm that solves Q-F, taking $O(g(n, m, \epsilon, \delta))$ samples on every instance, and there is an instance $(\bar{\mathcal{A}}, \bar{n}, \bar{m}, \bar{\epsilon}, \bar{\delta})$ on which every algorithm that solves Q-F takes $\Omega(g(\bar{n}, \bar{m}, \bar{\epsilon}, \bar{\delta}))$ samples.

Conjecture 4.1. *There exist a constant* $C > 0$*, and functions* $g, h \in \mathcal{U}$*, such that for every* $\delta \in (0, 1]$*, there exists an integer* $n_0 < C\log\frac{2}{\delta}$*, where for every* $n \leq n_0$*,* Q-F *is solvable in* $\Theta(g(n, m, \epsilon, \delta))$ *samples. Let for all such instances with* $n \leq n_0$*, the equivalent* Q-P *instance (obtained by posing the the instance of* Q-F *as an instance of* Q-P*, as done in proving Corollary 4.2) need at least* $\Omega(h(n, m, \epsilon, \delta))$ *samples. Then,* $\lim_{\delta\downarrow 0}\frac{g(n,m,\epsilon,\delta)}{h(n,m,\epsilon,\delta)} = 0$.

5. Experiments and Results

We begin our experimental evaluation by comparing $\mathcal{F}_2$ (Roy Chaudhuri & Kalyanakrishnan, 2017) and LUCB-k-m based on the number of samples drawn on different instances of Q-F or $(1, m, n)$. $\mathcal{F}_2$ is a fully-sequential algorithm that resembles LUCB-k-m, but subtle differences in the way the algorithms partition $\mathcal{A}$ and select arms to pull lead to different results. At each time step t, $\mathcal{F}_2$ partitions $\mathcal{A}$ into $\bar{A}_1(t)$, $\bar{A}_2(t)$, and $\bar{A}_3(t)$. It puts the highest-LCB arm in $\bar{A}_1(t)$; among the rest, it puts $m-1$ arms with the highest UCBs in $\bar{A}_2(t)$; and the remaining $n-m$ arms in $\bar{A}_3(t)$. At each time step t, it samples three arms—the arm in $\bar{A}_1(t)$, the least sampled arm in $\bar{A}_2(t)$, and the highest -UCB arm in $\bar{A}_3(t)$. Ties are broken uniformly at random.

We take five Bernoulli instance of sizes $n = 10, 20, 50, 100,$ and 200, with the means linearly spaced between 0.999 and 0.001 (both inclusive), and sorted in descending order. We name the bandit instance of size n as $\mathcal{I}_n$. Now, setting $\epsilon = 0.05, \delta = 0.001$, and $m = 0.1 \times n$, we run the experiments and compare the number of samples drawn by $\mathcal{F}_2$ and LUCB-k-m to solve these five instances for $k = 1$. In our implementation we have used KL-divergence based confidence bounds (Cappé et al., 2013; Kaufmann & Kalyanakrishnan, 2013) for both $\mathcal{F}_2$ and LUCB-k-m. As depicted by Figure 1(a), as the number of arms (n) increases, the sample complexity of both the algorithms increases due to increase in hardness H_ϵ. However, the sample complexity of $\mathcal{F}_2$ increases much faster than LUCB-k-m.

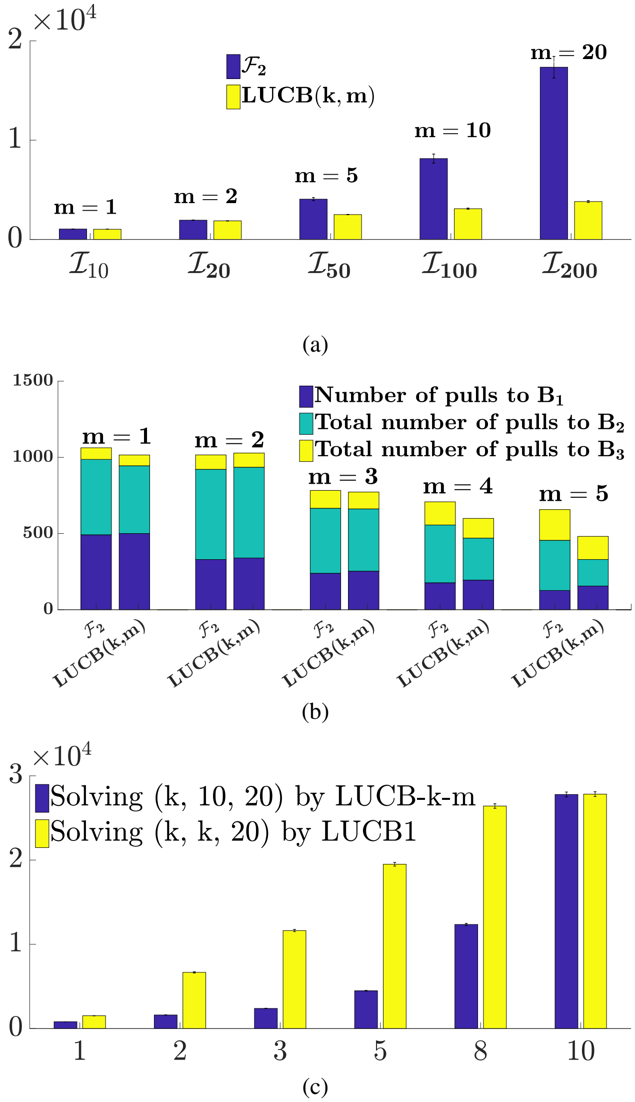
As shown by Jamieson & Nowak (2014) the efficiency of LUCB1 comes from the quick identification of the most optimal arm due to a large separation from the $(m+1)$-st arm. Intuitively, the possible reason for $\mathcal{F}_2$ to incur more samples is the delay in prioritising the optimal arm to pull more frequently. This should result in a smaller fraction of total samples taken from the best arm. Figure 1(b) affirms this intuition. It shows a comparison between $\mathcal{F}_2$ and LUCB-k-m on the number of samples obtained by three "ground-truth" groups—B_1, B_2, and B_3 on $\mathcal{I}_{10}$, keeping $k = 1$ and varying m from 1 to 5. We note that the lesser the difference between k and m, the higher the hardness (H_ϵ), and both $\mathcal{F}_2$ and LUCB-k-m find it hard to identify a correct arm. Hence, for $k = m = 1$, both of them spend almost the same fraction of pulls to the best arm. However, as m becomes larger, keeping $k = 1$, the hardness of the problem reduces, but $\mathcal{F}_2$ still struggles to identify the best arm and results in spending a significantly a lesser fraction of the total pulls to it, compared to LUCB-k-m.

In this paper, we have developed algorithms specifically for the (k, m, n) problem; previously one might have solved (k, m, n) either by solving (k, k, n) or (m, m, n): that is choosing the *best* k- or m-sized subset. In Figure 1(c) we present a comparison of the sample complexities for solving (k, m, n) and the best subset-selection problems. Fixing $\mathcal{A} = \mathcal{I}_{20}$, $n = 20$, $m = 10$, (k, m, n) instances are given by and varying $k \in \{1, 3, 5, 8, 10\}$, whereas, for the best subset-selection problem we set $m = k$. As expected, the number of samples incurred is significantly lesser for solving the problem instances with $k < m$, thereby validating the use of LUCB-k-m.

6. Conclusion

Identifying one arm out of the best m, in an n-armed stochastic bandit is an interesting problem identified by Roy Chaudhuri & Kalyanakrishnan (2017). They have mentioned the scenarios where identifying the best subset is practically infeasible. However, there are numerous examples in practice that demand efficient identification of multiple good solutions instead of only one; for example, assigning a distributed crowd-sourcing task, identification of good molecular combinations in drug designing, etc. In this paper, we present (k, m, n)—a generalised problem of identifying k out of the best m arms. Setting $k = 1$, (k, m, n) reduces to selection of one out of the best m arms, while setting $k = m$, makes it identical to "subset-selection" (Kalyanakrishnan & Stone, 2010). We have presented a lower bound on the sample complexity to solve (k, m, n). We have also presented a fully sequential adaptive PAC algorithm, LUCB-k-m, that solves (k, m, n), with expected sample complexity matching up to a constant factor that of $\mathcal{F}_2$ (Roy Chaudhuri & Kalyanakrishnan, 2017) and LUCB1 (Kalyanakrishnan et al., 2012) for $k = 1$ and $k = m$, respectively. Through an empirical comparison on different problem instances, we have shown that LUCB-k-m outperforms $\mathcal{F}_2$ by a large margin in terms of the number of samples as n grows.

For the problem of identification of a single $[\epsilon, \rho]$-optimal arm (Roy Chaudhuri & Kalyanakrishnan, 2017) in infinite bandit instances, the existing upper bound on the sample complexity differs from the lower bound by a multiplicative $\log \frac{1}{\delta}$ factor. It was not clear whether the lower bound was loose, or the upper can be improved, and left as an interesting problem to solve by Aziz et al. (2018). In this paper we reduce the gap by furnishing an upper bound which is optimal up to an *additive* poly-log term. Further, we show that the problem of identifying k distinct $[\epsilon, \rho]$-optimal arms is not well-posed in general, but when it is, we derive a lower bound on the sample complexity. Also, we identify a class of well-posed instances for which we present an efficient algorithm. Finally, we show how improved upper bounds on the sample complexity of solving Q-F can translate to improved upper bounds on the sample-complexity of solving Q-P. However, we conjecture that solving an instance of Q-F by posing it as an instance of Q-P with uniform sampling over arms will need more samples. Proving this conjecture and improving the bounds on the sample complexities are some interesting directions we leave for future work.



(a)

(b)

(c)

Figure 1. 1(a) Comparison of sample complexities of $\mathcal{F}_2$ and LUCB-k-m to solve Q-F with $m = n/10$, on the five instances detailed in Section 5. In this and subsequent graphs, the y-axis shows the average sample complexity over 100 runs, with standard error bars. 1(b) Comparison between $\mathcal{F}_2$ and LUCB-k-m on the number of pulls received by the camps B_1, B_2 and B_3, for solving different instances of Q-F on $\mathcal{I}_{10}$, by varying m from 1 to 5. Recall that B_1 is the singleton set, with the best arm being the only member. 1(c) Comparison of number of samples incurred for solving different instances of (k, m, n) defined on $\mathcal{I}_{20}$, by setting $m = 10$, and varying $k \in \{1, 2, 3, 5, 8, 10\}$. The x-axis shows k.

Acknowledgements

SK was partially supported by grants from SERB (ECR/2017/002479) and Ubisoft India.

References

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

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

Awerbuch, B. and Kleinberg, R. Online linear optimization and adaptive routing. In *J. Comput. Syst. Sci.*, volume 74, pp. 97–114. Academic Press, Inc., 2008.

Aziz, M., Anderton, J., Kaufmann, E., and Aslam, J. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *Proc. ALT 2018*, volume 83 of *PMLR*, pp. 3–24. PMLR, 2018.

Bechhofer, R. E. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. In *Biometrics*, volume 14, pp. 408–429. Wiley International Biometric Society, 1958.

Berry, D. and Fristedt, B. *Bandit Problems: Sequential Allocation of Experiments*. Chapman & Hall, 1985.

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

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

Even-Dar, E., Mannor, S., and Mansour, Y. PAC bounds for multi-armed bandit and Markov Decision Processes. In *Proc. COLT 2002*, pp. 255–270. Springer, 2002.

Gabillon, V., Ghavamzadeh, M., Lazaric, A., and Bubeck, S. Multi-bandit best arm identification. In *Adv. NIPS 24*, pp. 2222–2230. Curran Associates, Inc., 2011.

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

Herschkorn, S. J., Peköz, E., and Ross, S. M. Policies without memory for the infinite-armed Bernoulli bandit under the average-reward criterion. *Prob. in the Engg. and Info. Sc.*, 10(1):21–28, 1996.

Jamieson, K., Malloy, M., Nowak, R., and Bubeck, S. lil' UCB : An optimal exploration algorithm for multi-armed bandits. In *Proc. COLT 2014*, volume 35 of *PMLR*, pp. 423–439. PMLR, 2014.

Jamieson, K. G. and Nowak, R. D. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Proc. 48th Annual Conf. on Information Sciences and Systems (CISS)*, pp. 1–6. IEEE, 2014.

Jamieson, K. G., Haas, D., and Recht, B. On the detection of mixture distributions with applications to the most biased coin problem. *CoRR*, abs/1603.08037, 2016. URL `http://arxiv.org/abs/1603.08037`.

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

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

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

Karnin, Z., Koren, T., and Somekh, O. Almost optimal exploration in multi-armed bandits. In *Proc. ICML 2013*, volume 28, pp. 1238–1246. PMLR, 2013.

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

Kleinberg, R. Nearly tight bounds for the continuum-armed bandit problem. In *Adv. NIPS 17*, pp. 697–704. MIT Press, 2005.

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

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

Mousavi, S. H., Haghighat, J., Hamouda, W., and Dastbasteh, R. Analysis of a subset selection scheme for wireless sensor networks in time-varying fading channels. *IEEE Trans. Signal Process.*, 64(9):2193–2208, 2016.

Paulson, E. A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Stat.*, 35(1):174–180, 1964.

Ren, W., Liu, J., and Shroff, N. B. Exploring k out of top ρ fraction of arms in stochastic bandits. *CoRR*, abs/1810.11857, 2018. URL `http://arxiv.org/abs/1810.11857`.

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

Roy Chaudhuri, A. and Kalyanakrishnan, S. PAC identification of a bandit arm relative to a reward quantile. In *Proc. AAAI 2017*, pp. 1977–1985. AAAI Press, 2017.

Tran-Thanh, L., Stein, S., Rogers, A., and Jennings, N. R. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artif. Intl.*, 214:89 – 111, 2014.

Wang, Y., Audibert, J.-Y., and Munos, R. Algorithms for infinitely many-armed bandits. In *Adv. NIPS 21*, pp. 1729–1736. Curran Associates Inc., 2008.

Will, Y., McDuffie, J. E., Olaharski, A. J., and Jeffy, B. D. *Drug Discovery Toxicology: From Target Assessment to Translational Biomarkers*. Wiley, 2016.