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 $\rho$ 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 $\rho$ 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 $\rho$” 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...
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 independent 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), whereby 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 $(k, \rho)$ 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 $A$ be the set of arms in our given bandit instance. With each arm $a \in A$, there is an associated reward distribution supported on a subset of $[0, 1]$, with mean $\mu_a$. When pulled, arm $a \in A$ 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 $A = \{a_1, a_2, \ldots, a_n\}$, and assume, without loss of generality, that for arms $a_i, a_j \in A$, $\mu_{a_i} \geq \mu_{a_j}$ whenever $i \leq j$. Given a tolerance $\epsilon \in [0, 1]$ and $m \in \{1, 2, \ldots, n\}$, we call an arm $a \in A(\epsilon, m)$-optimal if $\mu_a \geq \mu_{a_m} - \epsilon$. We denote the set of all the $(\epsilon, m)$-optimal arms as $TOP_m(\epsilon) \triangleq \{a : \mu_a \geq \mu_{a_m} - \epsilon\}$. For brevity we denote $TOP_m(0)$ as $TOP_m$.

Now, we introduce $(k, m, n)$ problem.

$(k, m, n)$ Problem. An instance of the $(k, m, n)$ problem is of the form $(A, n, m, k, \epsilon, \delta)$, where $A$ is a set of arms with $|A| = n \geq 2$; $m \in \{1, 2, \ldots, n - 1\}$; $k \in \{1, \ldots, m\}$; 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 $(\epsilon, m)$-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 $\text{SUBSET}$ (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 $\text{SUBSET}$ (Kalyanakrishnan & Stone, 2010). In this paper, we show that any algorithm that solves $(k, m, n)$ must incur $\Omega\left(\frac{n}{(m-k+1)^2} \log \left(\frac{(c_{m,k})}{\epsilon}\right)\right)$ 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 $\text{LUCB1}$ (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,
which we denote \((k, \rho)\). Given a set of arms \(A\), a sampling distribution \(P_A, \epsilon \in (0, 1]\), and \(\rho \in (0, 1]\), an arm \(a \in A\) is called \([\epsilon, \rho]\)-optimal if \(P_a \sim P_A \{a \geq a', \epsilon \leq 1 - \rho\} \geq 1 - \rho \). For \(\rho, \epsilon \in (0, 1]\), we define the set of all \([\epsilon, \rho]\)-optimal arms as \(TOP_\rho(\epsilon)\), and we denote \(TOP_\rho(0)\) as \(TOP_\rho\). We recall the definition of Q-P from Roy Chaudhuri & Kalyanakrishnan (2017), and then generalise it to \((k, \rho)\).

**Q-P Problem** (Roy Chaudhuri & Kalyanakrishnan, 2017). An instance of Q-P is fixed by a bandit instance with a set of arms \(A\); a probability distribution \(P_A\) over \(A, \rho \in (0, 1]\); \(\epsilon \in (0, 1]\); and \(\delta \in (0, 1]\). An algorithm \(L\) is said to solve Q-P if and only if for every instance \((A, P_A, \rho, \epsilon, \delta)\), \(L\) terminates with probability 1, and returns an \([\epsilon, \rho]\)-optimal arm with probability at least 1 − \(\delta\).

\((k, \rho)\) **Problem.** An instance of \((k, \rho)\) problem is of the form \((A, P_A, k, \rho, \epsilon, \delta)\), where \(A\) is a set of arms; \(P_A\) is a probability distribution over \(A\); quantile fraction \(\rho \in (0, 1]\); tolerance \(\epsilon \in (0, 1]\); and mistake probability \(\delta \in (0, 1]\). Such an instance is valid if \(|TOP_\rho| \geq k\), and invalid otherwise. An algorithm \(L\) is said to solve \((k, \rho)\), if for every valid instance of \((k, \rho)\), \(L\) 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 \((k, \rho)\) is well-defined only if the given instance has at least \(k\) distinct arms in \(TOP_\rho\); 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 \(TOP_\rho\), one among which has a probability \(\gamma\) of being picked by \(P_A\), and the rest each a probability of \((\rho - \gamma)/(k - 1)\). Since the arms have to be identified by sampling from \(P_A\), 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 \(P_A\) allocates no more than \(\rho/k\) probability to any arm in \(TOP_\rho\). We refer to such instances as “at most \(k\)-equiprobable” instances. Formally, a \((k, \rho)\) problem instance given by \((A, P_A, k, \rho, \epsilon, \delta)\) is called “at most \(k\)-equiprobable” if \(\forall a \in TOP_\rho, P_{a' \sim P_A} \{a' = a\} \leq \frac{\rho}{k}\).

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 \(O\left(\frac{1}{\epsilon^2} \log^2 \frac{2}{\delta}\right)\) 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 \(P_3\) 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 \((k, \rho)\) instances. Also, \(P_3\) 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 \((k, \rho)\) 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.

---

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Lower Bound</th>
<th>Previous Upper Bound</th>
<th>Current Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 1, n)) Best-Arm</td>
<td>(\Omega\left(\frac{n}{\rho} \log \frac{2}{\delta}\right)) *((0.5 \log \frac{2}{\delta}))</td>
<td>(O\left(\frac{n}{\rho} \log \frac{2}{\delta}\right)) *((0.5 \log \frac{2}{\delta}))</td>
<td>Same as previous</td>
</tr>
<tr>
<td>((m, m, n)) SUBSET</td>
<td>(\Omega\left(\frac{n}{\rho} \log \frac{2}{\delta}\right)) *((0.5 \log \frac{2}{\delta}))</td>
<td>(O\left(\frac{n}{\rho} \log \frac{2}{\delta}\right)) *((0.5 \log \frac{2}{\delta}))</td>
<td>Same as previous</td>
</tr>
<tr>
<td>((1, m, n)) Q-F</td>
<td>(\Omega\left(\frac{m}{\rho} \log \frac{2}{\delta}\right))</td>
<td>(O\left(\frac{m}{\rho} \log \frac{2}{\delta}\right))</td>
<td>(O\left(\frac{k}{\rho} \log \frac{k}{\delta} + \log^2 \frac{2}{\delta}\right)) This paper</td>
</tr>
<tr>
<td>((k, m, n)) Q-F (_k)</td>
<td>(\Omega\left(\frac{n}{m-k+1} \log \frac{m-k+1}{\delta}\right))</td>
<td>(O\left(\frac{n}{m-k+1} \log \frac{m-k+1}{\delta}\right))</td>
<td>(O\left(\frac{k}{\rho} \log \frac{k}{\delta} + \log^2 \frac{2}{\delta}\right)) This paper</td>
</tr>
<tr>
<td>((1, \rho)) (</td>
<td>A</td>
<td>= \infty) Q-P</td>
<td>(\Omega\left(\frac{1}{\rho} \log \frac{2}{\delta}\right))</td>
</tr>
<tr>
<td>((k, \rho)) (</td>
<td>A</td>
<td>= \infty) Q-P (_k)</td>
<td>(\Omega\left(\frac{1}{\rho} \log \frac{2}{\delta}\right))</td>
</tr>
</tbody>
</table>
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 \(L\) be an algorithm that solves \((k, m, n)\). Then, there exist an instance \(\mathcal{A}, n, m, k, \epsilon, \delta\), with \(0 < \epsilon \leq \frac{1}{\sqrt{32}}, 0 < \delta \leq \frac{\epsilon^3}{4}\), and \(n \geq 2m, 1 \leq k \leq m\), on which the expected number of pulls performed by \(L\) is at least \(
\frac{1}{18375} \cdot \frac{1}{k^2} \cdot m \cdot n \cdot \sum_{t=1}^{m-k+1} \ln \left(\frac{m-1}{\epsilon^2 \delta}\right)
\)

The proof, given in Appendix A, generalises the lower bound proofs for both \((m, n)\) (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 full-sequentially 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 \((m, n)\). 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.

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 \(A_1^k\); the \(n - m\) arms with the lowest empirical averages in \(A_3^k\); and the rest in \(A_2^k\); ties are broken arbitrarily (uniformly at random in our experiments). Letting \(u_a^t\) be the number of pulls obtained by the arm \(a\) until the horizon \(t-1\), and \(\beta(a, t, k) \equiv \frac{1}{\sqrt{2k^2}} \ln \left(\frac{k^2 n t^2}{\epsilon^2 \delta}\right)\) (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 \(\hat{\mu}_a^t + \beta(a, t, k)\) and \(\hat{\mu}_a^t - \beta(a, t, k)\) respectively. At each round we choose a contentious arm from each of these three sets: from \(A_1^k\) we choose \(h_a^t\), the arm with the lowest LCB; from \(A_2^k\) the arm which is least pulled is chosen, and called \(m_a^t\); from \(A_3^k\) we choose \(l_a^t\), the arm with the highest UCB. The algorithm stops as soon as the difference between the LCB of \(h_a^t\), and the UCB of \(l_a^t\) is no larger than the tolerance \(\epsilon\).

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

\[
\Delta_a \equiv \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}

\]

We note that \(\Delta_k = \Delta_{k+1} = \cdots = \Delta_m = \Delta_{m+1}\). Now, we define the hardness term as \(H_e \equiv \frac{1}{\max(\Delta_a, \epsilon/2)}\).

**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 \(O\left(H_e \log \frac{H_e}{\delta}\right)\).

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 $\delta$. 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. $\mathcal{P}_2$ 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 $\delta'$ 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:** $\mathcal{A}, \rho, \epsilon$

**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 $[\epsilon/2, \rho]$-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, $S \cap T \mathcal{O} P_\rho = \emptyset$ can only happen if all the $u$ copies of $\mathcal{P}_2$ output sub-optimal arms. Therefore, $\Pr\{S \cap T \mathcal{O} P_\rho = \emptyset\} = (1 - \delta')^u \leq \delta/2$. 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 $\delta$. 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 $\frac{C}{\rho(\epsilon/2)^2} (\ln \frac{2}{\delta})^2 \leq O\left(\frac{1}{\rho \epsilon^2}\right)$. Hence, SC of all the $u$ copies $\mathcal{P}_2$, together is upper bounded by $\frac{C'u}{\rho \epsilon^2}$, for some constant $C_1$. Also, for some constants $C_2, C_3$, the sample complexity of MEDIAN ELIMINATION is upper bounded by $\frac{C'_u}{\rho \epsilon}$, for some constant $C_2$, $C_3$. 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 $O\left(\frac{1}{\rho} \left(\frac{n}{m} \log^2 \frac{1}{\delta} + \log^2 \frac{1}{\delta}\right)\right)$.

**Proof.** Let, $(\mathcal{A}, n, m, \epsilon, \delta)$ be the given instance of Q-F. We partition the set $\mathcal{A} = [0, 1]$ into $n$ 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 $P_{\mathcal{A}} = \text{Uniform}[0, 1]$, and $\rho = m/n$, we realise that solving the Q-P instance $(\mathcal{A}^\infty, P_{\mathcal{A}})$, 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 $(k, \rho)$, Next, we discuss the extension of Q-P to $(k, \rho)$, and present lower and upper bounds on the sample complexity needed to solve it.

4.2. Solving “At Most $k$-equiprobable” $(k, \rho)$ 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 $(k, \rho)$ by reducing it to solving a SUBSET problem as follows.

**Theorem 4.3.** [Lower Bound on the Sample Complexity for Solving $(k, \rho)$] 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 $(k, \rho)$ given by $(\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta)$, such that any algorithm that solves $(k, \rho)$ incurs at least $C \cdot \frac{k}{\rho^2} \ln \frac{k}{8\delta}$ 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 $\text{ALG}$ that can solve any instance of $(k, \rho)$ using no more than $C \cdot \frac{k}{\rho^2} \ln \frac{k}{8\delta}$ 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 $P_{\mathcal{A}} = \text{Uniform}[1, 2, \ldots, n]$, $k = m$, and $\rho = m/n$, we create an instance of $(k, \rho)$ as $(\mathcal{A}, P_{\mathcal{A}}, k, \epsilon, \delta)$. Therefore, solving this $(k, \rho)$ instance will solve the original SUBSET instance. According our claim, $\text{ALG}$ solves the
original SUBSET instance using at most \( C \cdot \frac{k^m}{(m/k)^{2r}} \ln \frac{k}{85} = C \cdot \frac{n^m}{(m/n)^{2r}} \ln \frac{k}{85} \) 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 \((k,\rho)\) instances. Let, for any \( S \subseteq A \), \( \nu(S) \triangleq \Pr_{a \sim P_A}(a \in S) \). Therefore, \( \nu(A) = 1 \). Now, we present an algorithm KQP-1 that can solve any at most \( k \)-equiprobable instance of \((k,\rho)\). Algorithm 3 describes KQP-1. At each phase \( y \), it solves an instance of Q-P to output an arm, say \( a(y) \), from \( TOP_\rho(\epsilon) \). In the next phase, it updates the bandit instance \( A^{y+1} = A^y \setminus \{a(y)\} \), the sampling distribution \( P_{A^{y+1}} = \frac{1}{\nu(a(y))} P_{A^y} \), and the target quantile \( \rho - \sum_{j=1}^y \nu(a(j)) \). However, as we are not given the explicit form of \( P_A \), we realise \( P_{A^{y+1}} \) by rejection-sampling—
i.e. \( a' \in A \setminus A^{y+1} \) is chosen by \( P_A \), we simply discard \( a' \), and continue to sample \( P_A \) one more time. Because \( \nu(\{a(y)\}) \) is not known explicitly, it is chosen 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 \((A^y, P_{A^y}, \rho - (y-1)\rho/k, \epsilon, \delta/k)\).

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

Input: \( A, P_A, k, \rho, \epsilon, \delta \).

Output: Set of \( k \) distinct arms from \( TOP_\rho(\epsilon) \).

\( A^1 = A \).

for \( y = 1, 2, 3, \ldots, k \) do

\( \rho^y = \rho - (y-1)\rho/k \).

Run \( \mathcal{P}_3 \) to solve the Q-P instance given by \((A^y, P_{A^y}, \rho^y, \epsilon/k^2)\), and let \( a(y) \) be the output.

\( A^{y+1} = A^y \setminus \{a(y)\} \).

end for

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

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

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

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

\[
\sum_{y=1}^k \text{SC}(y) \leq \sum_{y=1}^k C \epsilon^{-2} \left( \frac{1}{\rho^y} \log \frac{k}{\delta} + \log^2 \frac{k}{\delta} \right).
\]

\[
\leq C \epsilon^{-2} \left( \log \frac{k}{\delta} \sum_{y=1}^k \frac{1}{\rho - (y-1)\rho/k} + \log^2 \frac{k}{\delta} \right).
\]

\[
= Ck \epsilon^{-2} \left( \frac{1}{\rho} \log \frac{k}{\delta} \sum_{y=1}^k \frac{1}{k - y + 1} + \log^2 \frac{k}{\delta} \right),
\]

\[
\leq C'k \epsilon^{-2} \left( \frac{\log k}{\rho} \log \frac{k}{\delta} + \log^2 \frac{k}{\delta} \right),
\]

for \( k > 1 \), and some constant \( C' \).

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

We note that the sample complexity of KQP-1 is independent of the size of \( A \), and every instance of \((m,n,k,\epsilon,\delta)\) given by \((A,UniForm(A),m/n,\epsilon,\delta)\) can be solved by KQP-1 by posing it as an instance of \((k,\rho)\) given by \((A,UniForm(A),m/n,\epsilon,\delta)\). However, for \( k = m \), the sample complexity of KQP-1 reduces to \( O \left( \frac{n}{k} \left( \log m \cdot \log \frac{k}{\delta} + \log^2 \frac{k}{\delta} \right) \right) \), which is higher than the sample complexity of HALVING (Kalyanakrishnan & Stone, 2010), that needs only \( O \left( \frac{\log m}{k} \log \frac{k}{\delta} \right) \) 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 \( TOP_\rho(\epsilon) \) is small, solving \((k,\rho)\) 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 \((k,\rho)\) given by \((A, P_A, k, \rho, \epsilon, \delta)\), such that \(|A| = \infty\), and for all finite subset \( S \subseteq A \), \( \Pr_{a \sim P_A}(a \in S) = 0 \); 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 \((A, P_A, k, \rho, \epsilon, \delta/k)\).

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 \( \gamma : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \rightarrow \mathbb{R}^+ \). If every instance of Q-F given by \((A, n, m, \epsilon, \delta)\), can be solved with sample complexity \( O \left( \frac{n}{m^2} \log \frac{k}{\delta} + \gamma(n,m,\epsilon,\delta) \right) \), then, every instance of Q-P given by \((A, P_A, \rho, \epsilon, \delta)\) can be solved with sample complexity \( O \left( \frac{1}{\epsilon^2} \log (1/\delta) + \gamma \left( \frac{8 \log (2/\delta)}{1} \right) \epsilon / (2 \delta/2) \right) \).
We assume that there exists an algorithm \( \text{OPTQF} \) that solves every instance of Q-F given by \( (A, n, m, \epsilon, \delta) \), using \( O \left( \frac{n^2}{\epsilon^2} \log \frac{1}{\delta} + \gamma(n, m, \epsilon, \delta) \right) \) samples. We establish the upper bound on sample complexity for solving Q-P by constructing an algorithm \( \text{OPTQP} \) that follows an approach similar to \( F_3 \). Specifically, \( \text{OPTQP} \) reduces the input Q-P instance to an instance of Q-F using \( O \left( \frac{1}{\epsilon^2} \log \frac{1}{\delta} \right) \) samples. Then, it solves that Q-F using \( \text{OPTQF} \) 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 \( O \left( \frac{1}{\epsilon^2} \left( \frac{n^2}{\epsilon^2} \log \frac{1}{\delta} + \gamma(n, m, \epsilon, \delta) \right) \right) \) samples. Hence, \( \gamma(n, m, \epsilon, \delta) \in O \left( \frac{1}{\epsilon^2} \log 2 \log \frac{1}{\delta} \right) \), 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 \( \gamma(n, m, \epsilon, \delta) \in \Theta \left( \frac{n^2}{\epsilon^2} \log \frac{1}{\delta} \right) \). In that case, Q-P is solvable using \( \Theta \left( \frac{1}{\epsilon^2} \log \frac{1}{\delta} \right) \) 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 \( \epsilon, \delta \), and \( m \leq n/2 \), \( \gamma(n, m, \epsilon, \delta) \leq C \cdot \gamma \left( \left[ \log \frac{2}{\delta} \right], \left[ \log \frac{2}{\delta} \right], \frac{1}{2}, \frac{1}{2} \right) \). However, for \( n < \left[ \log \frac{2}{\delta} \right] \) we believe Q-F can be solved more efficiently than by reducing it to Q-P. Considering a set of functions, \( \mathcal{U} \triangleq \{ f : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0, 1] \times [0, 1] \rightarrow \mathbb{R}^+ \} \), 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 \( (A, n, m, \epsilon, \delta) \) on which every algorithm that solves Q-F takes \( \Omega(g(n, m, \epsilon, \delta)) \) 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 \( n_0 < C \log \frac{3}{\delta} \), 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, \( \lim_{\delta \to 0} \frac{g(n, m, \epsilon, \delta)}{h(n, m, \epsilon, \delta)} = 0 \).

5. Experiments and Results

We begin our experimental evaluation by comparing \( F_2 \) (Roy Chaudhuri & Kalyanakrishnan, 2017) and LUCB-k-m based on the number of samples drawn on different instances of Q-F or \( (1, m, n) \). \( F_2 \) is a fully-sequential algorithm that resembles LUCB-k-m, but subtle differences in the way the algorithms partition \( A \) and select arms to pull lead to different results. At each time step \( t \), \( F_2 \) partitions \( A \) into \( A_1(t) \), \( A_2(t) \), and \( A_3(t) \). It puts the highest-LCB arm in \( A_1(t) \); among the rest, it puts \( m-1 \) arms with the highest UCBs in \( A_2(t) \); and the remaining \( n-m \) arms in \( A_3(t) \). At each time step \( t \), it samples three arms—the arm in \( A_1(t) \), the least sampled arm in \( A_2(t) \), and the highest UCB arm in \( A_3(t) \). 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 \( I_n \). 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 \( F_2 \) 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 \( F_2 \) 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 \( H_n \). However, the sample complexity of \( F_2 \) 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 \( F_2 \) 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 \( F_2 \) and LUCB-k-m on the number of samples obtained by three “ground-truth” groups—\( B_1 \), \( B_2 \), and \( B_3 \) on \( I_{100} \), 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 \( (H_n) \), and both \( F_2 \) 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 \( F_2 \) still struggles to identify the best arm and results in spending a significantly 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 \( A = I_{50}, n = 20, m = 10, (k, m, n) \) 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.

Acknowledgements

SK was partially supported by grants from SERB (ECR/2017/002479) and Ubisoft India.
PAC Identification of Many Good Arms in Stochastic Multi-Armed Bandits

References


A. Lower Bound on the Worst Case Sample Complexity to Solve \((k, m, n)\)

**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 \(0 < \epsilon < \frac{1}{\sqrt{32}}, 0 < \delta \leq \frac{\epsilon^{-1}}{4}\), and \(n \geq 2m, 1 \leq k \leq m\), on which the expected number of pulls performed by \(\mathcal{L}\) is at least
\[
\frac{1}{18375} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m-k+1} \ln \left( \frac{(m-k+1)}{4\delta} \right).
\]

The proof technique for Theorem 3.1 follows a path similar to that of (Kalyanakrishnan et al., 2012, Theorem 8), but differs in the fact that any \(k\) of the \(m\) \((\epsilon, m)\)-optimal arms needs to be returned as opposed to all the \(m\).

**A.1. Bandit Instances:**

Assume we are given a set of \(n\) arms \(\mathcal{A} = \{0, 1, 2, \ldots, n-1\}\). Let \(I_0 \stackrel{\text{def}}{=} \{0, 1, 2, \ldots, m-k\}\) and \(\mathcal{I}_l \stackrel{\text{def}}{=} \{I : I \subseteq \mathcal{A} \setminus I_0 \wedge |I| = l\}\). Also for \(I \subseteq \{m-k+1, m-k+2, \ldots, n-1\}\), define
\[
\bar{I} \stackrel{\text{def}}{=} \{m-k+1, m-k+2, \ldots, n-1\} \setminus I.
\]

With each \(I \in \mathcal{I}_{k-1} \cup \mathcal{I}_m\) we associate an \(n\)-armed bandit instance \(\mathcal{B}^I\), in which each arm \(a\) produces a reward from a Bernoulli distribution with mean \(\mu_a\) defined as:
\[
\mu_a = \begin{cases} 
\frac{1}{2} & \text{if } a \in I_0 \\
\frac{1}{2} + 2\epsilon & \text{if } a \in I \\
\frac{1}{2} - 2\epsilon & \text{if } a \in \bar{I}.
\end{cases}
\]

Notice that all the instances in \(\mathcal{I}_{k-1} \cup \mathcal{I}_m\) have exactly \(m\) \((\epsilon, m)\)-optimal arms. For \(I \in \mathcal{I}_{k-1}\), all the arms in \(I_0\) are \((\epsilon, m)\)-optimal, but for \(I \in \mathcal{I}_m\) they are not. With slight overloading of notation we write \(\mu(S)\) to denote the multi-set consisting of means of the arms in \(S \subseteq \mathcal{A}\).

The key idea of the proof is that without sufficient sampling of each arm, it is not possible to correctly identify \(k\) of the \((\epsilon, m)\)-optimal arms with high probability.

**A.2. Bounding the Error Probability:**

We shall prove the theorem by first making the following assumption, which we shall demonstrate leads to a contradiction.

**Assumption 1.** Assume, that there exists an algorithm \(\mathcal{L}\), that solves each problem instance in \((k, m, n)\) defined on bandit instance \(\mathcal{B}^I, I \in \mathcal{I}_{k-1}\), and incurs a sample complexity \(SC_I\). Then for all \(I \in \mathcal{I}_{k-1}\),
\[
\mathbb{E}[SC_I] < \frac{1}{18375} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m-k+1} \ln \left( \frac{(m-k+1)}{4\delta} \right), \text{ for } 0 < \epsilon < \frac{1}{\sqrt{32}}, 0 < \delta \leq \frac{\epsilon^{-1}}{4}, \text{ and } n \geq 2m, \text{ where } C = \frac{1}{18375}.
\]

For convenience, we denote by \(Pr_I\) the probability distribution induced by the bandit instance \(\mathcal{B}^I\) and the possible randomisation introduced by the algorithm \(\mathcal{L}\). Also, let \(S_\mathcal{L}\) be the set of arms returned (as output) by \(\mathcal{L}\), and \(T_\mathcal{L}\) be the total number of times the arms in \(S \subseteq \mathcal{A}\) get sampled until \(\mathcal{L}\) stops.

Then, as \(\mathcal{L}\) solves \((k, m, n)\), for all \(I \in \mathcal{I}_{k-1}\)
\[
Pr_I\{S_\mathcal{L} \subseteq I_0 \cup I\} \geq 1 - \delta.
\]

Therefore, for all \(I \in \mathcal{I}_{k-1}\)
\[
\mathbb{E}_I[T_\mathcal{L}] \leq C \cdot \frac{n}{(m-k+1)} \epsilon^2 \ln \left( \frac{(m-k+1)}{4\delta} \right).
\]

**A.2.1. Changing \(Pr_I\) to \(Pr_{I \cup Q}\) where \(Q \in \bar{I}\) s.t. \(|Q| = m-k+1\):**

Consider an arbitrary but fixed \(I \in \mathcal{I}_{k-1}\). Consider a fixed partitioning of \(\mathcal{A}\), into \(\left\lfloor \frac{n}{m-k+1} \right\rfloor\) subsets of size \((m-k+1)\) each. If Assumption (1) is correct, then for the instance \(\mathcal{B}^I\), there are at most \(\left\lfloor \frac{n}{3(m-k+1)} \right\rfloor - 1\) partitions \(B \subseteq \bar{I}\), such that
\[ \mathbb{E}_I \{ T_B \} \geq \frac{4C}{\epsilon^2} \ln \left( \frac{1}{\delta} \right). \]

Now, as \( \left\lfloor \frac{n-m}{m-k+1} \right\rfloor - \left( \frac{n}{4(m-k+1)} - 1 \right) \geq \left\lfloor \frac{n}{4(m-k+1)} \right\rfloor + 1 > 0 \); therefore, there exists at least one subset \( Q \in \mathcal{I} \) such that \( |Q| = m - k + 1 \), and \( \mathbb{E}_I \{ T_Q \} < \frac{4C}{\epsilon^2} \ln \left( \frac{m}{4(m-k+1)} \right) \). Define \( T^* = \frac{16C}{\epsilon^2} \ln \left( \frac{m}{4(m-k+1)} \right) \). Then using Markov’s inequality we get:

\[
\Pr_I \{ T_Q \geq T^* \} < \frac{1}{4}.
\]  

(5)

Let \( \Delta = 2\epsilon T^* + \sqrt{T^*} \) and also let \( K_Q \) be the total rewards obtained from \( Q \).

**Lemma A.1.** If \( I \in \mathcal{I}_{k-1} \) and \( Q \in \mathcal{I} \) s.t. \(|Q| = m - k + 1\), then

\[
\Pr_I \left\{ T_Q \leq T^* \land K_Q \leq \frac{T_Q}{2} - \Delta \right\} \leq \frac{1}{4}.
\]

**Proof.** Let \( K_Q(t) \) be the total sum obtained from \( Q \) at the end of the trial \( t \). As for \( B^{I_i} \), \( \forall j \in Q \) \( \mu_j = 1/2 - 2\epsilon \) hence selecting and pulling one arm at each trial from \( Q \) following any rule (deterministic or probabilistic) is equivalent to selection of a single arm from \( Q \) for once and subsequently perform pulls on it. Hence whatever be the strategy of pulling one arm at each trial from \( Q \), the expected reward for each pull will be \( 1/2 - 2\epsilon \). Let \( r_i \) be the i.i.d. reward obtained from the \( i \)th trial. Then \( K_Q(t) = \sum_{i=1}^{t} r_i \) and \( \operatorname{Var}[r_i] = \left( \frac{1}{2} - 2\epsilon \right) \left( \frac{1}{2} + 2\epsilon \right) = \left( \frac{1}{2} - 4\epsilon^2 \right) < \frac{1}{2} \). As \( \forall i : 1 \leq i \leq t \), \( r_i \) are i.i.d., we get \( \operatorname{Var}[K_Q(t)] = \sum_{i=1}^{t} \operatorname{Var}[r_i] < \frac{t}{4} \). Now we can write the following:

\[
\Pr_I \left\{ \min_{1 \leq i \leq T^*} \left( K_Q(t) - t \left( \frac{1}{2} - 2\epsilon \right) \right) \leq -\sqrt{T^*} \right\} 
\leq \Pr_I \left\{ \max_{1 \leq i \leq T^*} \left| K_Q(t) - t \left( \frac{1}{2} - 2\epsilon \right) \right| \geq \sqrt{T^*} \right\} 
\leq \frac{\operatorname{Var}[K_Q(T^*)]}{T^*} < \frac{1}{4},
\]

(6)

wherein we have used Kolmogorov’s inequality.

**Lemma A.2.** Let \( I \in \mathcal{I}_{k-1} \) and \( Q \in \mathcal{I}_{m-k+1} \) such that \( Q \subseteq \mathcal{I} \), and let \( W \) be some fixed sequence of rewards obtained by a single run of algorithm \( \mathcal{L} \) on \( B^{I_i} \) such that \( T_Q \leq T^* \) and \( K_Q \geq \frac{T_Q}{2} - \Delta \), then:

\[
\Pr_{I \cup Q} \{ W \} > \Pr_I \{ W \} \cdot \exp(-32\epsilon \Delta).
\]

(7)

**Proof.** Recall the fact that all the arms in \( Q \) have the same mean. Hence, if chosen one at each trial (following any strategy), the expected reward at each trial remains the same. Hence the probability of getting a given reward sequence generated from \( Q \) is independent of the sampling strategy. Now as the arms in \( Q \) have higher mean in \( B^Q \), the probability of getting the sequence (of rewards) decreases monotonically as the 1-rewards for \( B^{I_i} \) become fewer. So we get

\[
\Pr_{I \cup Q} \{ W \} = \Pr_I \{ W \} \left( \frac{1}{2} + 2\epsilon \right)^{K_Q} \left( \frac{1}{2} - 2\epsilon \right)^{T_Q - K_Q} 
\geq \Pr_I \{ W \} \left( \frac{1}{2} + 2\epsilon \right)^{T_Q - \Delta} \left( \frac{1}{2} - 2\epsilon \right)^{T_Q - \Delta} 
\geq \Pr_I \{ W \} \cdot \left( \frac{1}{2} - 2\epsilon \right)^{2\Delta} 
> \Pr_I \{ W \} \cdot \exp(-32\epsilon \Delta) \quad \text{for } 0 < \epsilon \leq \frac{1}{\sqrt{32}}.
\]
Lemma A.3. If (5) holds for an $I \in \mathcal{I}_{k-1}$ and $Q \in \mathcal{I}_{m-k+1}$ such that $Q \subseteq \bar{I}$, and if $W$ is the set of all possible reward sequences $W$, obtained by algorithm $\mathcal{L}$ on $B^I$, then $\Pr_{I \cup Q}\{W\} > \left(\Pr_I\{W\} - \frac{1}{2}\right) \cdot 4\delta$. In particular,

$$\Pr_{I \cup Q}\{S_L \subseteq I_0 \cup I\} > \frac{\delta}{m^{m-k+1}}. \quad (8)$$

Proof. Let for some fixed sequence (of rewards) $W$, $T_W^Q$ and $K_W^Q$ respectively denote the total number of samples received by the arms in $Q$ and the total number of 1-rewards obtained before the algorithm $\mathcal{L}$ stopped. Then:

$$\Pr_{I \cup Q}\{W\} = \Pr_{I \cup Q}\{W : W \in W\} \geq \Pr_{I \cup Q}\{W : W \in W \land T_W^Q \leq T^* \land K_W^Q \geq T_W^Q/2 - \Delta\} > \Pr_I\{W : W \in W \land T_W^Q \leq T^* \land K_W^Q \geq T_W^Q/2 - \Delta\} \cdot \exp(-32\epsilon\Delta) \geq \left(\Pr_I\{W\} - \frac{1}{4}\right) \cdot \exp(-32\epsilon\Delta) \geq \left(\Pr_I\{W\} - \frac{1}{2}\right) \cdot \frac{4\delta}{m^{m-k+1}} \quad \text{for } C = \frac{1}{18375}, \delta < \frac{\epsilon^{-1}}{4}.$$

In the above, the 3rd, 4th and the last step are obtained using Lemma A.2, Lemma A.1 and Equation (5) respectively. The inequality (8) is obtained by using inequality (3), as $\Pr_I\{S_L \in I_0\} > 1 - \delta \geq 1 - \frac{\epsilon^{-1}}{4} > \frac{3}{4}$. \qed

A.2.2. Summing Over $\mathcal{I}_{k-1}$ and $\mathcal{I}_m$

Now, we sum up the probability of errors across all the instances in $\mathcal{I}_{k-1}$ and $\mathcal{I}_m$. If the Assumption 1 is true, using the pigeon-hole principle we show that there exists some instance for which the mistake probability is greater than $\delta$. 
Hence, we get a contradiction to Assumption 1, thereby proving the theorem.

Recall that \( \forall \alpha \in I_0 : S_\alpha = \{ J \setminus J' \} \cup \{ a \} \). Therefore,

\[
\sum_{J \in I_m} \Pr \{ S_\alpha \notin J \} \\
\geq \sum_{J \in I_m} \sum_{J' \subset J : |J'| = m-k+1} \Pr \{ S_\alpha \subseteq \{ J \setminus J' \} \cup I_0 \} \\
\geq \sum_{J \in I_m} \sum_{J' \subset J : |J'| = m-k+1} \Pr \{ \exists \alpha \in I_0 : S_\alpha = \{ J \setminus J' \} \cup \{ a \} \} \\
= \sum_{J \in I_m} \sum_{J' \subset J : |J'| = m-k+1} \sum_{I \in I_{k-1}} \mathbb{1}[I \cup J' = J] \cdot \Pr \{ S_\alpha \subseteq I \cup I_0 \} \\
= \sum_{J \in I_m} \sum_{J' \subset A \setminus I_0 : |J'| = m-k+1} \sum_{I \in I_{k-1}} \mathbb{1}[I \cup J' = J] \cdot \Pr \{ S_\alpha \subseteq I \cup I_0 \} \\
= \sum_{I \in I_{k-1}} \sum_{J \in I_m} \sum_{J' \subset I : |J'| = m-k+1} \mathbb{1}[I \cup J' = J] \cdot \Pr \{ S_\alpha \subseteq I \cup I_0 \} \\
= \sum_{I \in I_{k-1}} \Pr \{ \bigcup_{J \cup J'} \{ S_\alpha \subseteq I \cup I_0 \} \}
\]

Hence, we get a contradiction to Assumption 1, thereby proving the theorem.
B. Analysis of LUCB-k-m

Let at time $t$, $\hat{p}_a^t$ be the empirical mean of the arm $a \in A$, and $u_a^t$ be the number of times the arm $a$ has been pulled until (and excluding) time $t$. For a given $\delta \in (0, 1]$, we define $\beta(u_a^t, t, \delta) = \sqrt{\frac{k_1}{2at}} \ln \frac{k_1 \ln t}{\delta}$, where $k_1 = 5/4$. We define upper and lower confidence bound on the estimate of the true mean of arm $a \in A$ as $ucb(a, t) = \hat{p}_a + \beta(u_a^t, t, \delta)$, and $lcb(a, t) = \hat{p}_a - \beta(u_a^t, t, \delta)$ respectively.

To analyse the sample complexity, first we define some events, at least one of which must occur if the algorithm does not stop at the round $t$.

**Probable Events.** Let $a, b \in A$, such that $\mu_a > \mu_b$. During the run of the algorithm, any of the following five events may occur:

i) The empirical mean of an arm may falls outside the upper or the lower confidence bound. We define it as:

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

ii) The empirical mean of arm $a$ may be lesser than that of arm $b$; we define it as:

$$ErrA(a, b, t) \overset{\text{def}}{=} \{ \hat{p}_a < \hat{p}_b \}.$$  

iii) The lower and upper confidence bounds of arm $a$ may fall below those of arm $b$; we define them as:

$$ErrL(a, b, t) \overset{\text{def}}{=} \{ lcb(a, t) < lcb(b, t) \},$$

$$ErrU(a, b, t) \overset{\text{def}}{=} \{ ucb(a, t) < ucb(b, t) \}.$$  

iv) If an arm’s confidence bounds are above a certain radius (say $d$), we define that event as

$$NEEDY_a^t(d) \overset{\text{def}}{=} \{ lcb(a, t) < \mu_a - d \lor \{ ucb(a, t) > \mu_a + d \} \}.$$  

Let $u^*(a, t) \overset{\text{def}}{=} \max \left\{ \frac{32}{\max(\Delta_a, \frac{\Delta_b}{2})^2} \ln \frac{k_1 \ln t}{\delta} \right\}$ for all $a \in A$, where $k_1 = 5/4$. We show that any arm $a$, if sampled sufficiently, that is $u_a^t \geq u^*(a, t)$, then occurrence of any of the Probable Events imply occurrence of $CROSS_a^t$. First we show that if $CROSS_a^t$ does not occur for any $a \in A$, then occurrence of any one of the Probable Events implies the occurrence of $NEEDY_a^t(\cdot)$ or $NEEDY_b^t(\cdot)$.

**Lemma B.1.** (Expressing Probable Events in terms of NEEDY-a and CROSS-a) To prove that \{ $-CROSS_a^t \land -CROSS_b^t \} \land \{ ErrA(a, b, t) \lor ErrU(a, b, t) \lor ErrL(a, b, t) \} \implies \{ NEEDY_a^t(\frac{\Delta_a}{2}) \lor NEEDY_b^t(\frac{\Delta_b}{2}) \}.$

**Proof.** $ErrA(a, b, t)$: To prove that $-\{ CROSS_a^t \lor CROSS_b^t \} \land ErrA(a, b, t) \implies NEEDY_a^t(\frac{\Delta_a}{2}) \lor NEEDY_b^t(\frac{\Delta_b}{2}).$

$$ErrA(a, b, t) \implies \hat{p}_a^t < \hat{p}_b^t \implies (p_a - \beta(u_a^t, t, \delta)) < (p_b - \beta(u_b^t, t, \delta) + (\beta(u_a^t, t, \delta) + \beta(u_b^t, t, \delta)) - \Delta_{ab}/2) \implies NEEDY_a^t(\frac{\Delta_a}{2}) \lor NEEDY_b^t(\frac{\Delta_b}{2}).$$

$ErrU(a, b, t)$: To prove that $-\{ CROSS_a^t \lor CROSS_b^t \} \land ErrU(a, b, t) \implies NEEDY_a^t(\frac{\Delta_a}{2}).$
We show that given a threshold $d$.

Assuming $\neg\text{CROSS}_a^t \wedge \neg\text{CROSS}_b^t$ we get

$$\text{ErrU}(a, b, t) \implies \{ucb(b, t) > ucb(a, t)\}$$

$$\implies \{\hat{p}_b^t + \beta(u_b^t, t, \delta) > \hat{p}_a^t + \beta(u_a^t, t, \delta)\}$$

$$\implies \{\hat{p}_b^t > \mu_b + \beta(u_b^t, t, \delta)\} \lor \{\hat{p}_a^t < \mu_a - \beta(u_a^t, t, \delta)\} \lor \{2\beta(u_a^t, t, \delta) > \Delta_{ab}\}$$

$$\implies \text{NEEDY}_a^t \left(\frac{\Delta_{ab}}{2}\right).$$

$\text{ErrL}(a, b, t)$: To prove that $\neg\{\text{CROSS}_a^t \lor \text{CROSS}_b^t\} \wedge \text{ErrL}(a, b, t) \implies \text{NEEDY}_a^t \left(\frac{\Delta_{ab}}{2}\right)$.

Assuming $\neg\text{CROSS}_a^t \wedge \neg\text{CROSS}_b^t$ we get

$$\text{ErrL}(a, b, t) \implies \{lc(b, t) > lcb(a, t)\}$$

$$\implies \{\hat{p}_b^t - \beta(u_b^t, t, \delta) > \hat{p}_a^t - \beta(u_a^t, t, \delta)\}$$

$$\implies \{\hat{p}_b^t > \mu_b + \beta(u_b^t, t, \delta)\} \lor \{\hat{p}_a^t < \mu_a - \beta(u_a^t, t, \delta)\} \lor \{2\beta(u_a^t, t, \delta) > \Delta_{ab}\}$$

$$\implies \text{NEEDY}_a^t \left(\frac{\Delta_{ab}}{2}\right).$$

We show that given a threshold $d$, if an arm $a$ is sufficiently sampled, such that $\beta(u_a^t, t, \delta) \leq \frac{d}{2}$, then NEEDY$^t_a$ infers CROSS$^t_a$.

**Lemma B.2.** For any $a \in \mathcal{A}$, $\{\text{NEEDY}_a^t(d) | \beta(u_a^t, t, \delta) < d/2\} \implies \text{CROSS}_a^t$.

**Proof.** First, we show that $\{lc(b, t) < \mu_a - d|\beta(u_a^t, t, \delta) < d/2\} \implies \text{CROSS}_a^t$.

$$\{lc(b, t) < \mu_a - d|\beta(u_a^t, t, \delta) < d/2\}$$

$$\implies \{\hat{p}_b^t - \beta(u_b^t, t, \delta) < \mu_a - d|\beta(u_a^t, t, \delta) < d/2\}$$

$$\implies \{\hat{p}_b^t < \mu_a - d + \beta(u_a^t, t, \delta)|\beta(u_a^t, t, \delta) < d/2\}$$

$$\implies \{\hat{p}_b^t < \mu_a - d/2|\beta(u_a^t, t, \delta) < d/2\}$$

$$\implies \text{CROSS}_a^t.$$ (9)

The other part follows the similar way.

By the very definition of confidence bound, at any round $t$, the probability that the empirical mean of an arm will lie outside it, is very low. In other words, the probability of occurrence $\text{CROSS}_a^t$ is very low for all $t$ and $a \in \mathcal{A}$.

**Lemma B.3.** [Upper bounding the probability of $\text{CROSS}_a^t$] \forall a \in \mathcal{A} \land \forall t \geq 0$, $\Pr\{\text{CROSS}_a^t\} \leq \frac{\delta}{knt^3}$. Hence, $P \left[\exists t \geq 0 \land \exists a \in \mathcal{A} : \text{CROSS}_a^t | u_a^t \geq 0\right] \leq \frac{\delta}{knt^3}$.

**Proof.** $\Pr\{\text{CROSS}_a^t\}$ is upper bounded by using Hoeffding’s inequality, and the next statement gets proved by taking union bound over all arms and $t$.

Now, recalling the definition of $h_i^t$, and $l_i^t$ from Algorithm 1, we present the key logic underlying the analysis of LUCB-k-m. The idea is to show that if the algorithm has not stopped, then one of those PROBABLE EVENTS must have occurred. Then using Lemma B.1, and Lemma B.2, Lemma B.3 we show that beyond a certain number of rounds, the probability that LUCB-k-m will continue is sufficiently small. Lastly, using the argument based on pigeon-hole principle, similar to Lemma 5 of Kalyanakrishnan (2011), we establish the upper bound on the sample complexity. Below we present the core logic that shows, until the algorithm stops one of the PROBABLE EVENTS must occur.
Case 1 $h^*_t \in B_1 \land l^*_t \in B_1$

if $\exists b_3 \in A_3^t \cap B_3$ then
Then $ErrL(h^*_t, b_3, t)$ has occurred.
else
$\exists b_3 \in A_2^t \cap B_3$
Then $ErrA(h^*_t, b_3, t)$ has occurred.
end if

Case 2 $h^*_t \in B_1 \land l^*_t \in B_2$

if $\exists b_3 \in A_3^t \cap B_3$ then
Then $ErrL(h^*_t, b_3, t)$ has occurred.
else
$\exists b_3 \in A_2^t \cap B_3$.
if $\Delta_{h^*_t} \geq \frac{\Delta_{l^*_t}}{2}$ then
Then $NEEDY_{h^*_t}^t(\Delta_{h^*_t}/4) \lor NEEDY_{l^*_t}^t(\Delta_{l^*_t}/4)$ has occurred.
else
Then $ErrL(l^*_t, b_t, t)$ has occurred.
end if
end if

Case 3 $h^*_t \in B_1 \land l^*_t \in B_3$

Then $NEEDY_{h^*_t}^t(\Delta_{h^*_t}/4) \lor NEEDY_{l^*_t}^t(\Delta_{l^*_t}/4)$ has occurred.

Case 4 $h^*_t \in B_2 \land l^*_t \in B_3$

if $\Delta_{h^*_t} \geq \frac{\Delta_{l^*_t}}{2}$ then
Then $ErrA(l^*_t, h^*_t, t)$ has occurred.
else
if $\exists b_3 \in A_3^t \land B_3$ then
Then $ErrL(h^*_t, b_3^t, t)$ has occurred.
else
$\exists b_3 \in A_2^t \land B_3$
$: ErrA(l^*_t, b_3, t)$ has occurred.
end if
end if
Case 5 \( h^*_t \in B_2 \land l^*_t \in B_2 \) and \( \Delta_{h^*_t} > 0 \)

Here, \( \exists b_1 \in (A^*_2 \cup A^*_3) \cap B_1 \) and \( \exists b_3 \in (A^*_1 \cup A^*_2) \cap B_3 \)

if \( |\Delta_{h^*_t}| < \Delta_{h^*_t} / 2 \) then

if \( \Delta_{b_1 h^*_t} > \Delta_{b_1} / 4 \) then

if \( b_1 \in A^*_1 \cap B_1 \) then

\( ErrA(b_1, h^*_t, t) \)

else

\( b_1 \in A^*_3 \cap B_1 \)

\( ErrU(b_1, l^*_t, t) \) has occurred.

end if

else

\( \Delta_{b_3 l^*_t} \leq \Delta_{b_3} / 4 \) and hence \( \Delta_{b_3 l^*_t} \geq \Delta_{l^*_t} / 4 \)

if \( b_3 \in A^*_2 \cap B_3 \) then

\( ErrA(l^*_t, b_3, t) \) has occurred.

else

\( b_3 \in A^*_1 \cap B_3 \)

\( ErrL(h^*_t, b_3, t) \) has occurred.

end if

end if

else

\( |\Delta_{h^*_t l^*_t}| > \Delta_{h^*_t} / 2 \)

\( NEEDY_{h^*_t} (\Delta_{h^*_t} / 4) \lor NEEDY_{l^*_t} (\Delta_{l^*_t} / 4) \) has occurred.

end if

---

Case 5 (continued) \( h^*_t \in B_2 \land l^*_t \in B_2 \) and \( \Delta_{h^*_t} \leq 0 \)

Here, \( \exists b_1 \in (A^*_1 \cup A^*_2) \cap B_3 \) and \( \exists b_3 \in (A^*_1 \cup A^*_2) \cap B_3 \)

if \( |\Delta_{h^*_t l^*_t}| < \Delta_{h^*_t} / 2 \) then

if \( \Delta_{b_1 h^*_t} > \Delta_{b_1} / 4 \) then

if \( b_1 \in A^*_2 \cap B_1 \) then

\( ErrA(b_1, h^*_t, t) \)

else

\( b_1 \in A^*_3 \cap B_1 \)

\( ErrU(b_1, l^*_t, t) \) has occurred.

end if

else

\( \Delta_{b_3 l^*_t} \leq \Delta_{b_3} / 4 \) and hence \( \Delta_{b_3 l^*_t} \geq \Delta_{l^*_t} / 4 \)

if \( b_3 \in A^*_2 \cap B_3 \) then

\( ErrA(l^*_t, b_3, t) \) has occurred.

else

\( b_3 \in A^*_1 \cap B_3 \)

\( ErrL(h^*_t, b_3, t) \) has occurred.

end if

end if

else

\( |\Delta_{h^*_t l^*_t}| > \Delta_{h^*_t} / 2 \)

\( NEEDY_{h^*_t} (\Delta_{h^*_t} / 4) \lor NEEDY_{l^*_t} (\Delta_{l^*_t} / 4) \) has occurred.

end if
Case 6 If $h^*_t \in B_2 \land l^*_t \in B_3$

If $\Delta h^*_t, l^*_t \geq \frac{\Delta_t}{2}$ then $\text{NEEDY}_{h^*_t}^l (\Delta / 4) \lor \text{NEEDY}_{l^*_t}^l (\Delta / 4)$ has occurred.

Else

$\Delta h^*_t, l^*_t < \frac{\Delta_t}{2}$

:. $\forall b_1 \in \{ A_2^t \cup A_3^t \} \cap B_1$, $\Delta_{b_1, h^*_t} > \frac{\Delta_{b_1}}{2}$.

If $\exists b_1 \in A_2^t \cap B_1$

$\text{ErrA}(b_1, h^*_t, t)$ has occurred.

Else

$\exists b_1 \in A_3^t \cap B_1$.

:. $\text{ErrU}(b_1, l^*_t, t)$ has occurred.

end if

end if

Case 7 $h^*_t \in B_3 \land l^*_t \in B_1$

:. $\text{ErrA}(l^*_t, h^*_t, t)$ has occurred.

Case 8 $h^*_t \in B_3 \land l^*_t \in B_2$

If $\Delta h^*_t, l^*_t \geq \frac{\Delta_t}{4}$ $\text{ErrA}(l^*_t, h^*_t, t)$ has occurred.

Else

$\Delta h^*_t, l^*_t < \frac{\Delta_t}{4}$

:. $\forall b_1 \in \{ A_2^t \cup A_3^t \} \cap B_1$, $\Delta_{b_1, l^*_t} > \frac{\Delta_{b_1}}{2}$.

If $\exists b_1 \in A_2^t \cap B_1$

$\text{ErrA}(b_1, h^*_t, t)$ has occurred.

Else

$\exists b_1 \in A_3^t \cap B_1$.

:. $\text{ErrU}(b_1, l^*_t, t)$ has occurred.

end if

end if

Case 9 $h^*_t \in B_3 \land l^*_t \in B_3$

$\exists b_1 \in \{ A_2^t \cup A_3^t \} \cap B_1$

If $\exists b_1 \in A_2^t \cap B_1$

$\text{ErrA}(b_1, h^*_t, t)$ has occurred.

Else

$\exists b_1 \in A_3^t \cap B_1$.

:. $\text{ErrA}(b_1, l^*_t, t)$ has occurred.

end if

Lemma B.4 (H). If $T = CH_\epsilon \ln \left( \frac{H_\epsilon}{\delta} \right)$, then for $C \geq 2732$, the following holds:

$$T > 2 + 2 \sum_{a \in \mathcal{A}} u^*(a, T).$$

Proof. This proof is taken from Appendix B.3 of Kalyanakrishnan (2011).

$$2 + 2 \sum_{a \in \mathcal{A}} u^*(a, T) = 2 + 64 \sum_{a \in \mathcal{A}} \left[ \frac{1}{\max(\Delta_a, (\epsilon/2))^2} \ln \frac{knt^4}{\delta} \right]$$
\[ \leq 2 + 64n + 64H_e \ln \frac{knT^4}{\delta} \]
\[ = 2 + 64n + 64H_e \ln k + 64H_e \ln \frac{n}{\delta} + 256H_e \ln T \]
\[ < (66 + 64 \ln k)H_e + 64H_e \ln \frac{n}{\delta} + 256H_e \left[ \ln C + \ln H_e + \ln \frac{H_e}{\delta} \right] \]
\[ < (66 + 64 \ln k)H_e + 64H_e \ln \frac{n}{\delta} + 256H_e \left[ \ln C + \ln H_e + \ln \frac{H_e}{\delta} \right] \]
\[ < 130H_e + 64H_e \ln \frac{n}{\delta} + 256H_e \left[ \ln C + 2 \ln \frac{H_e}{\delta} \right] \]
\[ < (706 + 256 \ln C)H_e \ln \frac{H_e}{\delta} < CH_e \ln \frac{H_e}{\delta} \quad \text{[For } C \geq 2732] \]

**Lemma B.5.** Let \( T^* = \left\lceil \frac{2732H_e \ln \left( \frac{H_e}{\delta} \right)}{\delta} \right\rceil \). For every \( T > T^*_1 \), the probability that the Algorithm 1 has not terminated after \( T \) rounds of sampling is at most \( \frac{8\delta}{T^2} \).

**Proof.** Letting \( \bar{T} = \frac{T}{2} \) we define two events for \( \bar{T} \leq t \leq T-1 \): \( E^{(1)} \equiv \exists a \in A : CROSS'_a \) and \( E^{(2)} \equiv \exists NEEDY'_a \left( \frac{\Delta_a}{4} \right) \).

If the algorithm stops for \( t < \bar{T} \) then there is nothing to prove. On the contrary, let the algorithm has not stopped after \( t > \bar{T} \) and neither \( E^{(1)} \) nor \( E^{(2)} \) has occurred. Letting \( N_{\text{rounds}} \) be the the required number of rounds beyond \( T \), we can upper bound it as:

\[
N_{\text{rounds}} = \sum_{t=\bar{T}}^{T-1} \left( \Pi \left[ NEEDY'_{h^*_t} \left( \frac{\Delta_{h^*_t}}{4} \right) \vee NEEDY'_{m^*_t} \left( \frac{\Delta_{m^*_t}}{4} \right) \vee NEEDY'_{l^*_t} \left( \frac{\Delta_{l^*_t}}{4} \right) \right] \right)
\]
\[
\leq \sum_{t=\bar{T}}^{T-1} \sum_{a \in A} \left[ a \in \{ h^*_t, m^*_t, l^*_t \} \wedge NEEDY'_a \left( \frac{\Delta_a}{4} \right) \right]
\]
\[
= \sum_{t=\bar{T}}^{T-1} \sum_{a \in A} \left[ a \in \{ h^*_t, m^*_t, l^*_t \} \wedge (u^*_a < u^*(a, t)) \right]
\]
\[
\leq \sum_{t=\bar{T}}^{T-1} \sum_{a \in A} \left[ a \in \{ h^*_t, m^*_t, l^*_t \} \wedge (u^*_a < u^*(a, t)) \right]
\]
\[
\leq \sum_{a \in A} u^*(a, t).
\]

Using Lemma B.4, \( T \geq T^* \Rightarrow T > 2 + 2 \sum_{a \in A} u^*(a, t) \). Hence, if neither \( E^{(1)} \) nor \( E^{(2)} \) occurs then the algorithm runs for at most \( \bar{T} + N_{\text{rounds}} \leq \left\lceil \frac{T}{2} \right\rceil + \sum_{a \in A} 16u^*(a, t) < T \) number of rounds.

The probability that the algorithm does not stop within \( T \) rounds, is upper-bounded by \( P[E^{(1)} \vee E^{(2)}] \). Applying Lemma B.2 and Lemma B.3,

\[
P[E^{(1)} \vee E^{(2)}] \leq \sum_{t=\bar{T}}^{T-1} \left( \frac{\delta}{k_1t^3} + \frac{\delta}{kt^3} \right) \leq \sum_{t=\bar{T}}^{T-1} \frac{\delta}{k_1t^3} \left( 1 + \frac{2}{t} \right) \leq \left( \frac{T}{2} \right) \frac{8\delta}{k_1T^3} \left( 1 + \frac{4}{T} \right) < \frac{8\delta}{T^2}.
\]
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 \(O(\frac{H}{\epsilon} \log \frac{H}{\delta})\).

Using Lemma B.4, and Lemma B.5 the expected sample complexity of the Algorithm 1 can be upper bounded as

\[
E[SC] \leq 2 \left( T_1^* + \sum_{i=T_1^*}^{\infty} \frac{8\delta}{T^2} \right) \leq 5464 \cdot \left( H \ln \left( \frac{H \epsilon}{\delta} \right) \right) + 32.
\]  

(10)
C. Proof of Theorem 4.7

Algorithm 4 describes \textsc{OPTQP}. It uses \textsc{PQ} (Roy Chaudhuri & Kalyanakrishnan, 2017) with \textsc{Median Elimination} as the subroutine (inside \textsc{PQ}), 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 4 \textsc{OPTQP}

\textbf{Input:} \(A, \epsilon, \delta\), and \textsc{OPTQF}.

\textbf{Output:} A single \([\epsilon, \rho]\)-optimal arm

Set \(\delta' = 1/4\),

\[ u = \left\lceil \frac{\epsilon}{2 \cdot (0.5 - \delta')} \cdot \log \frac{2}{\delta} \right\rceil = \left\lceil 8 \log \frac{2}{\delta} \right\rceil. \]

Run \(u\) copies of \(\textsc{PQ} \left( A, \rho, \epsilon/2, \delta' \right) \) and form set \(S\) with the output arms.

Return the output from \textsc{OPTQF} \(\left( S, u, \left\lfloor \frac{u}{2} \right\rfloor, 1, \frac{\epsilon}{2}, \delta \right)\).

\textbf{Theorem C.1.} \textbf{[Correctness and Sample Complexity of \textsc{OPTQP}]} If \textsc{OPTQF} exists, then \textsc{OPTQP} solves \textsc{Q-P}, within the sample complexity \(\Theta \left( \frac{1}{\rho \epsilon^2} \log \frac{1}{\delta} + \gamma(\cdot) \right)\).

\textbf{Proof.} First we prove the correctness and then upper bound the sample complexity.

\textbf{Correctness.} First we notice that each copy of \(\textsc{PQ}\) outputs an \([\epsilon/2, \rho]\)-optimal arm with probability at least \(1 - \delta'\). Also, \textsc{OPTQF} outputs an \([\epsilon/2, \rho]\)-optimal arm with probability \(1 - \delta\). Let, \(\hat{X}\) be the fraction of sub-optimal arms in \(S\). Then

\[ \Pr \{ \hat{X} \geq \frac{1}{2} \} = \Pr \{ \hat{X} - \delta' \geq \frac{1}{4} \} \leq \exp(\frac{-2 \cdot (\frac{1}{4})^2 \cdot u}{8 \log \frac{2}{\delta}}) = \exp\left(\frac{-1}{16} \cdot 8 \log \frac{2}{\delta}\right) < \frac{\delta}{2}. \]

On the other hand, the mistake probability of \textsc{OPTQF} is upper bounded by \(\delta/2\). Therefore, by taking union bound, we get the mistake probability is upper bounded by \(\delta\). Also, the mean of the output arm is not less than \(\frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon\) from the \((1 - \rho)\)-th quantile.

\textbf{Sample complexity.} First we note that, for some appropriate constant \(C\), the sample complexity (SC) of each of the \(u\) copies of \(\textsc{PQ}\) is \(\frac{C}{u \epsilon^2} \left( \log \frac{2}{\delta} \right)^2 \in O \left( \frac{1}{\rho \epsilon^2} \right)\). Hence, SC of all the \(u\) copies \(\textsc{PQ}\) together is upper bounded by \(\frac{C_1 u}{\rho \epsilon^2} \), for some constant \(C_1\). Also, for some constant \(C_2\), the sample complexity of \textsc{OPTQF} is upper bounded by \(C_2 \left( \frac{u}{(u/2)(\epsilon/2)^2} \log \frac{2}{\delta} + \gamma(\cdot) \right) = C_2 \left( \frac{8}{\epsilon^2} \log \frac{2}{\delta} + \gamma(\cdot) \right)\). Now, adding the sample complexities, and substituting for \(u\) we prove the bound. \qed