REPRESENTATIVE ARM IDENTIFICATION: A FIXED CONFIDENCE APPROACH TO IDENTIFY CLUSTER REPRESENTATIVES

Sarvesh Gharat *
Centre for Machine Intelligence and Data Science
IIT Bombay

Aniket Yadav
Centre for Machine Intelligence and Data Science
IIT Bombay

Nikhil Karamchandani
Department of Electrical Engineering
IIT Bombay

Jayakrishnan Nair
Department of Electrical Engineering
IIT Bombay

August 27, 2024

ABSTRACT

We study the representative arm identification (RAI) problem in the multi-armed bandits (MAB) framework, wherein we have a collection of arms, each associated with an unknown reward distribution. An underlying instance is defined by a partitioning of the arms into clusters of predefined sizes, such that for any $j > i$, all arms in cluster i have a larger mean reward than those in cluster j. The goal in RAI is to reliably identify a certain prespecified number of arms from each cluster, while using as few arm pulls as possible. The RAI problem covers as special cases several well-studied MAB problems such as identifying the best arm or any M out of the top K, as well as both full and coarse ranking. We start by providing an instance-dependent lower bound on the sample complexity of any feasible algorithm for this setting. We then propose two algorithms, based on the idea of confidence intervals, and provide high probability upper bounds on their sample complexity, which orderwise match the lower bound. Finally, we do an empirical comparison of both algorithms along with an LUCB-type alternative on both synthetic and real-world datasets, and demonstrate the superior performance of our proposed schemes in most cases.

1 Introduction

The stochastic multi-armed bandit (MAB) problem [1] is a widely studied online decision-making framework, which consists of K arms each associated with an a priori unknown reward distribution. Each pull of an arm results in a random reward, generated i.i.d. from the associated distribution. At each round, the learner can decide which arm to pull based on the entire history of pulls and rewards. The MAB framework has found application in a wide variety of domains ranging from A/B testing [2] and online advertising [3] to network routing [4], clinical testing [5], and hyperparameter optimization [6] in machine learning.

There are several objectives that a learner might be interested in while interacting with the MAB. For example, one widely studied goal is to maximize the expected cumulative reward accrued by the learner over a certain time horizon, or equivalently to minimize the *regret* with respect to an oracle which knows the arm reward distributions beforehand. Several regret minimization algorithms have been proposed in the literature [7, 8], and they are typically based on the idea of balancing *exploration* (trying different arms to reduce uncertainty about their mean rewards) and *exploitation* (pulling the arms known to have high rewards).

Another popular learning objective is identifying the best arm in terms of the mean reward [9, 10]. The problem has been studied in both the *fixed confidence* setting [10], where the goal is to find the best arm using the minimum number of pulls while guaranteeing a certain pre-specified error probability δ; and the *fixed budget* setting [11] where the total

---
*sarveshgharat19@gmail.com

number of pulls is fixed beforehand and the aim is to minimize the probability of error. In our work, we will focus on the fixed confidence setting. Several such *pure exploration* problems beyond best arm identification have been studied in the literature, including identifying the top K arms [12] or the arms with mean rewards above a given threshold [13]. These problems can often require a very large number of samples to solve, which makes them impractical for many applications of interest. One way of rectifying this shortcoming is to relax the objective, for example to identifying an ϵ-best arm [14] whose mean reward is within some small ϵ gap to the best arm, or any M out of the top K arms [15, 16], or any 'good' arm whose mean reward is above a threshold [17]. All the above relaxations can be treated as MAB problems with *multiple correct answers*, which has been studied recently in [17] wherein a general lower bound on the sample complexity and an asymptotically optimal (as error probability $\delta \to 0$) algorithm are provided.

Along similar lines, in this work, we propose the Representative Arm Identification (RAI) problem for which an underlying MAB instance is defined by a partitioning of the arms into clusters of predefined sizes, such that for any $j > i$, all arms in cluster i have a larger mean reward than those in cluster j. The goal in RAI is to reliably identify a certain prespecified number of arms from each cluster, while using as few arm pulls as possible. The RAI problem naturally arises in several real-world applications. For example, consider a crowdsourcing platform which engages a collection of workers with apriori unknown skill levels. Say, the platform is hired to solve a large task, which can be broken into multiple sub-tasks of differing complexity. Then the platform might want to hire some workers amongst the best few to tackle the hard subtasks, some with around average skill level to deal with subtasks of medium hardness, and then a few with the lowest skill level to handle the easiest subtasks. Another application of the RAI problem is in online recommendation systems, for example a content creator might be interested in knowing a few movies amongst the best rated, and also some amongst the worst rated on the platform.

Beyond several natural applications, the RAI problem is also interesting from a theoretical viewpoint since it covers as special cases several well-studied MAB problems such as best arm identification [10] or identifying any M out of the top K arms [16], as well as both full and coarse ranking [18], and is able to provide a unifying viewpoint for all these problems; see Table 1 for a list of such problems. We are able to provide a lower bound on the sample complexity of the general RAI problem, in terms of a quantity we call the 'bottleneck gap' which depends on the underlying instance and the arms requirements, and encodes the hardness of the problem. We also present two algorithms, based on the idea of confidence intervals, for reliably solving any RAI problem and provide high probability upper bounds on the sample complexity of these schemes. Finally, we conduct an empirical comparison of both our algorithms along with a LUCB-type baseline on both synthetic and real-world datasets, and demonstrate the superior performance of our proposed schemes in most cases.

2 Problem Formulation

We consider a stochastic multi-armed bandit with a collection $\mathcal{N}$ of $N = |\mathcal{N}|$ arms, each associated with a $\frac{1}{2}$-subGaussian reward distribution[2], which is a priori unknown to the learner.

To define the representative arm identification (RAI) problem, we sort the arms in decreasing order of their mean rewards and then partition them into m clusters of predefined sizes given by $c = (c_1, c_2, \cdots, c_m)$, such that for any $j > i$, all arms in cluster i have a larger mean reward than those in cluster j. Clearly, $\sum_{i=1}^{m} c_i = N$.

We label the arms as follows: for each $i \in [m] := \{1, 2, \ldots, m\}$ and $j \in [c_i] := \{1, 2, \ldots, c_i\}$, we have an associated reward distribution Π_j^i with the j-th arm in cluster i, so that each pull of the arm results in an i.i.d. reward sample from Π_j^i. The corresponding mean reward is denoted by μ_j^i, and we have that $\mu_{j_1}^i \geq \mu_{j_2}^i$ for any $j_1 \geq j_2$. See Figure 1 for an illustration. We assume this partition of arms into clusters is uniquely defined, i.e., if $i_1 \neq i_2$, $\nexists\, j_1, j_2$ such that $\mu_{j_1}^{i_1} = \mu_{j_2}^{i_2}$.

---

[2] A random variable X is σ-subGaussian if, for any $t > 0$, $\mathbb{P}\left(|X - \mathbb{E}[X]| > t\right) \leq 2 \exp\left(-t^2/2\sigma^2\right)$.
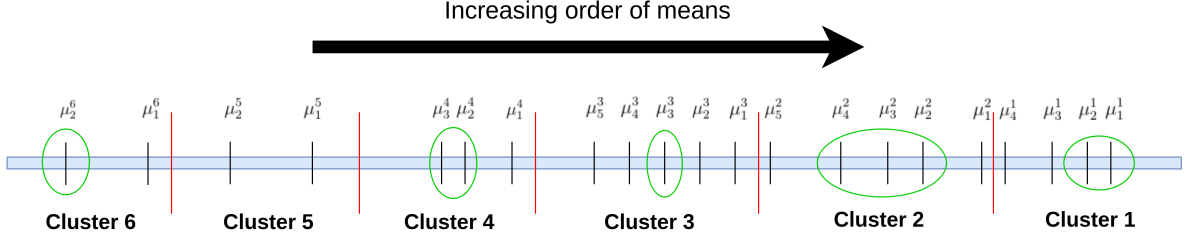
Figure 1: An RAI problem instance with $m = 6$ clusters, $c = (4, 5, 5, 3, 2, 2)$, and $r = (2, 3, 1, 2, 0, 1)$. The circled arms illustrate one of the correct outputs for this problem.

Finally, we have a prespecified vector $r = (r_1, r_2, \ldots, r_m)$ such that $0 \leq r_i \leq c_i$ for all i, and where r_i denotes the number of 'representative' arms belonging to cluster i that the learner needs to identify. As mentioned before, the RAI problem covers a wide range of very well-studied MAB problems, ranging from the best arm identification problem, where $c = (1, N-1), r = (1, 0)$, to full ranking, where $c = (1, 1, \cdots, 1), r = (1, 1, \cdots, 1)$. Table 1 lists several classical learning tasks from the literature that are special cases of the RAI problem. An important aspect of the RAI problem is that there can be *multiple correct answers*; specifically, this is the case if $0 < r_i < c_i$ for some $i \in [m]$.

To summarize, an instance of the RAI problem is defined by $\mathcal{I} = (c, r, \Pi)$, where $\Pi = (\Pi_j^i, i \in [m], j \in [c_i])$. To perform the RAI task, the learner can use an online algorithm, say $\mathcal{A}$, which at each time can either choose an arm to sample based on past observations; or decide to stop and output an estimated collection of representative arms from each cluster, given by $(O_i, i \in [m])$; here, O_i denotes the set of representatives from cluster i, with $|O_i| = r_i$. Given a prespecified error threshold $\delta \in (0, 1)$, we say that the algorithm $\mathcal{A}$ is δ-probably correct (δ-PC) if, for any underlying problem instance $\mathcal{I}$, the probability that algorithm output is incorrect is at most δ. More formally, denoting the (random) stopping time of the algorithm $\mathcal{A}$ by $T_\delta^{\mathcal{I}}(\mathcal{A})$, $\mathcal{A}$ is δ-PC if, for any instance $\mathcal{I}$,

$$\mathbb{P}\left(T_\delta^{\mathcal{I}}(\mathcal{A}) < \infty, \; \exists \, i \in [m] \text{ and } j \in O_i \text{ s.t. arm } j \text{ is not in cluster } i\right) \leq \delta.$$

The performance of a δ-PC algorithm $\mathcal{A}$ is captured via its sample complexity $T_\delta^{\mathcal{I}}(\mathcal{A})$, i.e., the number of arm pulls it makes before stopping. Our goal in this paper is to design δ-PC schemes for the RAI problem, whose sample complexity $T_\delta^{\mathcal{I}}(\mathcal{A})$ is as small as possible. Note that $T_\delta^{\mathcal{I}}(\mathcal{A})$ is itself a random quantity, and our results will be in terms of expectation or high probability bounds.

Table 1: MAB problems from the literature that are special cases of the RAI problem

Task	Number of Clusters	c	r
Best Arm Identification [9]	2	$(1, N-1)$	$(1, 0)$
One out of Top K [15]	2	$(K, N-K)$	$(1, 0)$
Top K arm identification [12]	2	$(K, N-K)$	$(K, 0)$
M out of Top K [16]	2	$(K, N-K)$	$(M, 0)$
Coarse Ranking [18]	m	$(c_1, c_2, \cdots, c_m)$	$(c_1, c_2, \cdots, c_m)$
Full Ranking [18]	N	$(1, 1, \cdots, 1)$	$(1, 1, \cdots, 1)$

3 Lower Bound

In this section, we derive a lower bound on the sample complexity of any δ-PC algorithm for the RAI problem. To state the result, we first need a few definitions.

Definition 1. (Arm Gap): Consider an instance $\mathcal{I} = (c, r, \Pi)$; recall that μ_j^i denotes the mean reward of the jth arm in cluster i. The *arm gap* Δ_j^i for that arm is defined as

$$\Delta_j^i := \min\{\mu_{c_{i-1}}^{i-1} - \mu_j^i, \mu_j^i - \mu_1^{i+1}\}, \quad \forall j \in c_i, i \in [m].$$

Here, for notational simplicity, we have assumed two dummy clusters, 0 and $m+1$, having one arm each, such that $\mu_1^0 = \infty, \mu_1^{m+1} = -\infty$.

In words, Δ_j^i represents the the gap (in mean reward) between the jth arm in cluster i and the 'nearest' arm in a neighboring cluster.

Next, we define the bottleneck gap associated with an instance.

Definition 2. (Bottleneck Gap): Consider an instance $\mathcal{I} = (c, r, \Pi)$. For each cluster $i \in [m]$, let $\Delta_\mathcal{I}^i$ denote the r_i-th largest arm gap amongst its arms. By convention, $\Delta_\mathcal{I}^i = \infty$ if $r_i = 0$. The *bottleneck gap* $\Delta_\mathcal{I}$ associated with the instance $\mathcal{I}$ is defined as

$$\Delta_\mathcal{I} := \min\{\Delta_\mathcal{I}^1, \Delta_\mathcal{I}^2, \cdots, \Delta_\mathcal{I}^m\}.$$

Intuitively, $\Delta_\mathcal{I}^i$ captures the complexity associated with the sub-task of identifying r_i arms from cluster i, while $\Delta_\mathcal{I}$ captures the complexity associated with the complete RAI task; the smaller these gaps, the harder the corresponding task. This is formalized in the following theorem.

Theorem 1. *For a given error threshold $\delta \in (0, 1)$, in the space of $\frac{1}{2}$-Gaussian instances (i.e., each arm has a Gaussian reward distribution with standard deviation $\sigma = \frac{1}{2}$), any δ-PC algorithm $\mathcal{A}$ for the RAI problem satisfies*

$$\liminf_{\delta \to 0} \frac{E[T_\delta^\mathcal{I}(\mathcal{A})]}{\log(1/\delta)} \geq \frac{1}{2(\Delta_\mathcal{I})^2}$$

The proof of Theorem 1 is provided in Appendix A. As mentioned before, the RAI problem falls in the class of MAB problems with multiple correct answers and as such, the general lower bound provided in [17] applies to the RAI problem as well. While the lower bound in [17] is in the form of a $\min \min \max$ optimization problem, the result above specializes it to the RAI problem, and also relaxes it to provide a simpler, more interpretable bound in terms of the bottleneck gap of the underlying instance. While the lower bound in Theorem 1 is in general looser than the (less explicit) bound that follows from [17], it captures the core complexity of the RAI task, and also enables a direct comparison to upper bounds on the sample complexity of our proposed algorithms (see Section 4).

We conclude this section by specializing the lower bound in Theorem 1 to the task of identifying M out of the top K arms, where $1 \leq M \leq K < N$ (this special case of the RAI problem was analysed in [16]).[3] Without loss of generality, suppose that the arms are also labelled $1, 2, \ldots, N$ such that the corresponding mean rewards satisfy $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_N$. Then it is easy to show that the bottleneck gap for this task is given by $\Delta_\mathcal{I} = \mu_M - \mu_{K+1}$. To the best of our knowledge, such an explicit, interpretable, instance-dependent complexity characterization is not available in the literature for this task.

4 Algorithms

This section describes the two algorithms we propose to solve the RAI problem and presents upper bounds on their sample complexity. In spirit, these algorithms are similar to the *successive elimination* style schemes which are widely used for MAB problems [10]. Under both algorithms, active arms are pulled in a round robin fashion, and suitable confidence intervals are maintained for the mean reward of each active arm. From time to time, arms whose membership in a certain cluster can be inferred based on the computed confidence intervals are 'selected' to be part of the algorithm output; these selected arms are then removed from the active set. The two algorithms differ with respect to the scheduling of the membership check—the *Vanilla Round Robin Algorithm* performs this check after each round robin cycle, whereas the *Butterscotch Round Robin Algorithm* performs the membership check only on each halving of the confidence interval widths.

Vanilla Round Robin Algorithm for RAI

The Vanilla Round Robin Algorithm is stated formally as Algorithm 1. This algorithm proceeds in rounds; each round involves the following steps:

- The algorithm samples every arm in its active set A once (line 4); this is the set of arms whose cluster membership is not yet confirmed.
- The arms in A are then partitioned into tentative clusters of sizes $\tilde{c}_1, \cdots, \tilde{c}_m$ based on their empirical mean rewards (line 6); here, $\tilde{c}_i$ is the (estimated) number of active arms from cluster i (more formally, $\tilde{c}_i = c_i - |O_i|$, where O_i denotes the set of representative arms that have been identified by the algorithm as being from cluster i by that point).

---

[3]Note that this task further subsumes other classical tasks like best arm identification ($M = K = 1$), and the identification of one of the top K arms (analysed in [15]) as special cases.

Algorithm 1 Vanilla Round Robin Algorithm for RAI

Input: cluster sizes $c = (c_1, c_2, \cdots, c_m)$, required arms $r = (r_1, r_2, \cdots, r_m)$, arm set $\mathcal{N}$, error threshold δ
Output: $O_1, O_2, \cdots, O_m$

1: Initialize $R \leftarrow 0$, $A \leftarrow \mathcal{N}$, and for $i \in \{1, 2, \cdots, m\}$ set $\tilde{c}_i = c_i$
2: **while** $|O_1| \neq r_1$ or $|O_2| \neq r_2$ or $\cdots |O_m| \neq r_m$ **do**
3:     Increment R by 1
4:     Sample every arm in A once
5:     Update empirical mean rewards
6:     Partition A into clusters $A_1, A_2, \cdots, A_m$ of sizes $\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_m$ respectively, based on the empirical means
7:     **for** i in $[m]$ **do**
8:         **for** arm a in A_i **do**
9:             $\text{Better}(a) = \left\{ j \in \cup_{j<i} A_j : \hat{\mu}_j - \hat{\mu}_a > 2\sqrt{\frac{\ln(\pi^2 R^2 N/3\delta)}{2R}} \right\}$
10:           $\text{Worse}(a) = \left\{ j \in \cup_{j>i} A_j : \hat{\mu}_a - \hat{\mu}_j > 2\sqrt{\frac{\ln(\pi^2 R^2 N/3\delta)}{2R}} \right\}$
11:             **if** $|\text{Better}(a)| \geq \sum_{j<i} \tilde{c}_j$ and $|\text{Worse}(a)| \geq \sum_{j>i} \tilde{c}_j$ **then**
12:                 **if** $|O_i| < r_i$ **then**
13:                     Add arm a to O_i
14:                 **end if**
15:                 Remove arm a from A
16:                 $\tilde{c}_i \leftarrow \tilde{c}_i - 1$
17:             **end if**
18:         **end for**
19:         **if** $i > 1$ and $|O_{i-1}| = r_{i-1}$ and $|O_i| = r_i$ **then**
20:             $\tilde{c}_i \leftarrow \tilde{c}_{i-1} + \tilde{c}_i$
21:             $\tilde{c}_{i-1} \leftarrow 0$
22:         **end if**
23:     **end for**
24: **end while**

- Next, for each arm a in the active set, a check is made to see whether its cluster assignment, say i, can be finalised. This entails checking whether the confidence intervals indicate arm a as being 'better' than $\sum_{j>i} \tilde{c}_i$ active arms, and 'worse' than $\sum_{j<i} \tilde{c}_i$ active arms (line 11). If this holds, arm a is assigned to cluster i (if additional representatives are required from cluster i), and it is removed from the active set.

- Finally, we merge adjacent clusters whose requirement for representative arms has been met (lines 19–22). Such a merger relaxes the membership check for the merged cluster (note that the algorithm does not need to output any arms from the merged cluster), hastening the removal of its members from the active set. Empirically, we find that this final step results in significantly fewer arm pulls by the algorithm, as we demonstrate in Section 5.

Of course, the algorithm terminates when the requisite number of representatives have been identified from each cluster. The following result establishes that the Vanilla Round Robin Algorithm is δ-PC, and also provides an instance-dependent high probability upper bound on the sample complexity of the algorithm.

Theorem 2. *The Vanilla Round Robin Algorithm (see Algorithm 1) is δ-PC for the RAI problem. With probability at least $1 - \delta$, its sample complexity $\mathcal{T}_\delta^{\mathcal{I}}$ satisfies*

$$\mathcal{T}_\delta^{\mathcal{I}} \leq \sum_{i=1}^{m} \sum_{j=1}^{c_i} \left(\mathbb{1}\{\Delta_j^i \geq \Delta_{\mathcal{I}}\} \frac{26}{(\Delta_j^i)^2} \ln\left(\frac{16\pi\sqrt{\frac{N}{3\delta}}}{(\Delta_j^i)^2} \right) + \mathbb{1}\{\Delta_j^i < \Delta_{\mathcal{I}}\} \frac{26}{(\Delta_{\mathcal{I}})^2} \ln\left(\frac{16\pi\sqrt{\frac{N}{3\delta}}}{(\Delta_{\mathcal{I}})^2} \right) + 1 \right).$$

We make the following remarks on the high probability upper bound on the stopping time under Algorithm 1. First, the dependence on the vector of required arms r is implicit through the definition of the bottleneck distance $\Delta_{\mathcal{I}}$. Secondly, the expression includes a summation over all the arms, each term capturing an upper bound on the number of rounds that arm is pulled. Intuitively, arms for which the gap Δ_j^i (as defined in Definition 1) exceeds the bottleneck gap $\Delta_{\mathcal{I}}$ are most likely to be assigned to their respective clusters by the algorithm; the number of rounds these arms remain active is (with high probability) inversely proportional to $(\Delta_j^i)^2$. On the other hand, arms whose gap Δ_j^i is less than the

---

Algorithm 2 Butterscotch Round Robin Algorithm for RAI

Input: cluster sizes $c = (c_1, c_2, \cdots, c_m)$, required arms $r = (r_1, r_2, \cdots, r_m)$, arm set $\mathcal{N}$, error threshold δ
Output: $O_1, O_2, \cdots, O_m$

1: Initialize $R \leftarrow 0, A \leftarrow \mathcal{N}$, and for $i \in \{1, 2, \cdots, m\}$ set $\tilde{c}_i = c_i$

2: Set $t_0 = 0$ and $t_R = 2^{(2R+5)} \ln\left(\frac{\pi^2 R^2 N}{3\delta}\right)$

3: **while** $|O_1| \neq r_1$ or $|O_2| \neq r_2$ or $\cdots |O_m| \neq r_m$ **do**

4:     Increment R by 1

5:     Sample every arm in A $t_R - t_{R-1}$ times

6:     Update empirical mean rewards

7:     Partition A into clusters $A_1, A_2, \cdots, A_m$ of sizes $\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_m$ respectively, based on the empirical means

8:     **for** i in $[m]$ **do**

9:         **for** arm a in A_i **do**

10:             $\mathrm{Better}(a) = \left\{ j \in \cup_{j<i} A_j : \hat{\mu}_j - \hat{\mu}_a > 2^{-(R+2)} \right\}$

11:             $\mathrm{Worse}(a) = \left\{ j \in \cup_{j>i} A_j : \hat{\mu}_a - \hat{\mu}_j > 2^{-(R+2)} \right\}$

12:             **if** $|\mathrm{Better}(a)| \geq \sum_{j<i} \tilde{c}_j$ and $|\mathrm{Worse}(a)| \geq \sum_{j>i} \tilde{c}_j$ **then**

13:                 **if** $|O_i| < r_i$ **then**

14:                     Add arm a to O_i

15:                 **end if**

16:                 Remove arm a from A

17:                 $\tilde{c}_i \leftarrow \tilde{c}_i - 1$

18:             **end if**

19:         **end for**

20:         **if** $i > 1$ and $|O_{i-1}| = r_{i-1}$ and $|O_i| = r_i$ **then**

21:             $\tilde{c}_i \leftarrow \tilde{c}_{i-1} + \tilde{c}_i$

22:             $\tilde{c}_{i-1} \leftarrow 0$

23:         **end if**

24:     **end for**

25: **end while**

---

bottleneck gap $\Delta_{\mathcal{I}}$ are most likely to remain active (i.e., not assigned to their respective clusters) until the algorithm stops; the number of rounds the algorithm needs to terminate being (with high probability) inversely proportional to $(\Delta_{\mathcal{I}})^2$. Finally, note that this bound is 'order-wise' consistent with the information theoretic lower bound in Theorem 1; both bounds exhibit an inverse proportionality to the square of the bottleneck gap. This suggests that the Vanilla Round Robin Algorithm is near optimal.[4]

Butterscotch Round Robin Algorithm for RAI

Next, we discuss the *Butterscotch Round Robin Algorithm*, stated formally as Algorithm 2. The key difference from the Vanilla Algorithm discussed before is that the arm pulls are conducted in batches (line 5). Specifically, in round R, each arm in the active set A is pulled $t_R - t_{R-1}$ times, where the value of t_R is chosen (line 2) so that the confidence interval for the mean reward of each arm in A is of size $\propto 2^{-R}$. This is reflected in the confidence-interval based cluster membership check conducted in line 10. The rest of Algorithm 2 proceeds in essentially the same way as Algorithm 1.

The structure of Algorithm 2 is inspired by the scheme proposed in [14] for the best arm identification problem, where it is shown that the batch property enables tighter confidence intervals and consequently a better upper bound on the sample complexity as compared to standard successive elimination based schemes. We find that something similar is true for the RAI problem as well, as illustrated by the following result and the discussion thereafter.

Theorem 3. *The Butterscotch Round Robin Algorithm (see Algorithm 2) is δ-PC for the RAI problem. With probability at least $1 - \delta$, its sample complexity $\mathcal{T}_\delta^{\mathcal{I}}$ satisfies*

$$\mathcal{T}_\delta^{\mathcal{I}} \leq \sum_{i=1}^m \sum_{j=1}^{c_i} \left(\mathbb{1}\{\Delta_j^i \geq \Delta_{\mathcal{I}}\} \max\left(\frac{32}{(\Delta_j^i)^2} \ln\left(\frac{N\pi^2}{3\delta} \left\lceil \log_2\left(\frac{1}{2\Delta_j^i} \right) \right\rceil \right)^2, 128 \ln\left(\frac{N\pi^2}{3\delta} \right) \right) \right)$$

---

[4]The lower bound in Theorem 1 is on the *expected* stopping time, whereas the upper bound in Theorem 2 holds with *high probability*; these quantities are not comparable strictly speaking. This 'abuse' is however standard in the analysis of confidence interval based MAB algorithms (see [10]).

$$+\mathbb{1}\{\Delta_j^i < \Delta_{\mathcal{I}}\} \max\left(\frac{32}{(\Delta_{\mathcal{I}})^2} \ln\left(\frac{N\pi^2}{3\delta}\left\lceil\log_2\left(\frac{1}{2\Delta_{\mathcal{I}}}\right)\right\rceil^2\right)\right), 128\ln\left(\frac{N\pi^2}{3\delta}\right)\right).$$

The bound above has a similar form to the one in Theorem 2. The main difference is that while the former has a $\log(1/(\text{arm gap}))$ (or $\log(1/\Delta_{\mathcal{I}})$) dependence, the latter has a term proportional to only $\log\log(1/(\text{arm gap}))$ (or $\log\log(1/\Delta_{\mathcal{I}})$). This potential improvement in sample complexity is due to the incorporation of batch pulling in Algorithm 2, which allows us to relax the union bound requirements in the proof. The empirical performance of the two algorithms is compared in Section 5. The proofs for Theorems 2 and 3 can be found in Appendix A. For each algorithm, the proof has two components: showing that it is δ-PC and then deriving an upper bound on the sample complexity.

5 Numerical Case Studies

In this section, we conduct an empirical evaluation of Algorithms 1 and 2 using both synthetic and real-world datasets. In addition, we also consider a suitably tailored version of LUCB-style sampling [12], which has been widely used in the multi-armed bandit literature and offers a sequential sampling strategy as opposed to the parallel nature of the successive elimination style strategies. In each round, the LUCB algorithm considers every empirical cluster for which the arms requirement hasn't yet been fulfilled and then based on the current confidence intervals of the mean estimates, selects from each cluster an arm (together with the 'boundary' arms of the neighboring clusters) whose membership is most likely to be confirmed with the additional pull; further details along with a proof that the proposed variant is δ-PC are provided in Appendix B. For our simulations, we assume the reward distribution to be Bernoulli$[0, 1]$, a member of the $1/2$ SubGaussian distribution family. Finally, we set the error probability $\delta = .01$ and present sample complexity results which are averaged over 100 independent runs of the corresponding algorithms.

We first examine an instance with 10 arms divided into clusters of sizes $3, 5$, and 2, respectively. The true means for this instance are given by: $[0.9, 0.85, 0.7, 0.66, 0.65, 0.6, 0.4, 0.35, 0.2, 0.15]$. Table 2 presents the average sample complexity of the three algorithms for various special cases of the RAI problem.

Table 2: Comparison of sample complexity between Algorithm 1, Algorithm 2, and an LUCB-style scheme for various instantiations of the RAI problem

Sr.no	Task	Required Arms	Vanilla	Butterscotch	LUCB
1	One arm from top cluster	$(1, 0, 0)$	4588	10370	3634
2	Identify top cluster	$(3, 0, 0)$	48632	31757	63802
3	2 arms each from top 2 clusters	$(2, 2, 0)$	8958	10370	19444
4	Identify center cluster	$(0, 5, 0)$	56730	50142	61302
5	One arm from worst cluster	$(0, 0, 1)$	8913	10370	9257

We observe that in almost all the cases, the Vanilla and Butterscotch schemes perform better than the LUCB-based algorithm. Intuitively, this is because the LUCB algorithm inherently targets sequential membership identification of arms which can lead to many unnecessary pulls for the boundary arms especially when the number of required arms is higher. Secondly, between the Vanilla and Butterscotch algorithms, the former performs better in Problems $1, 3, 5$ while the latter is superior for Problems $2, 4$. Problems $1, 3, 5$ turn out to be simpler problems that require only one round of the Butterscotch scheme (which is why the sample complexity is also the same for all three problems), while the Vanilla algorithm requires even fewer since it can stop at any time and does not have a batch constraint.

Both Algorithms 1 and 2 employ adaptive merging of clusters when the representative arm requirement for neighboring clusters is found to be already satisfied. To demonstrate the benefit of this feature, we conduct an empirical comparison of both algorithms with their corresponding non-merging versions, as shown in Table 3. The underlying bandit instance and the problems considered are the same as those in Table 2.
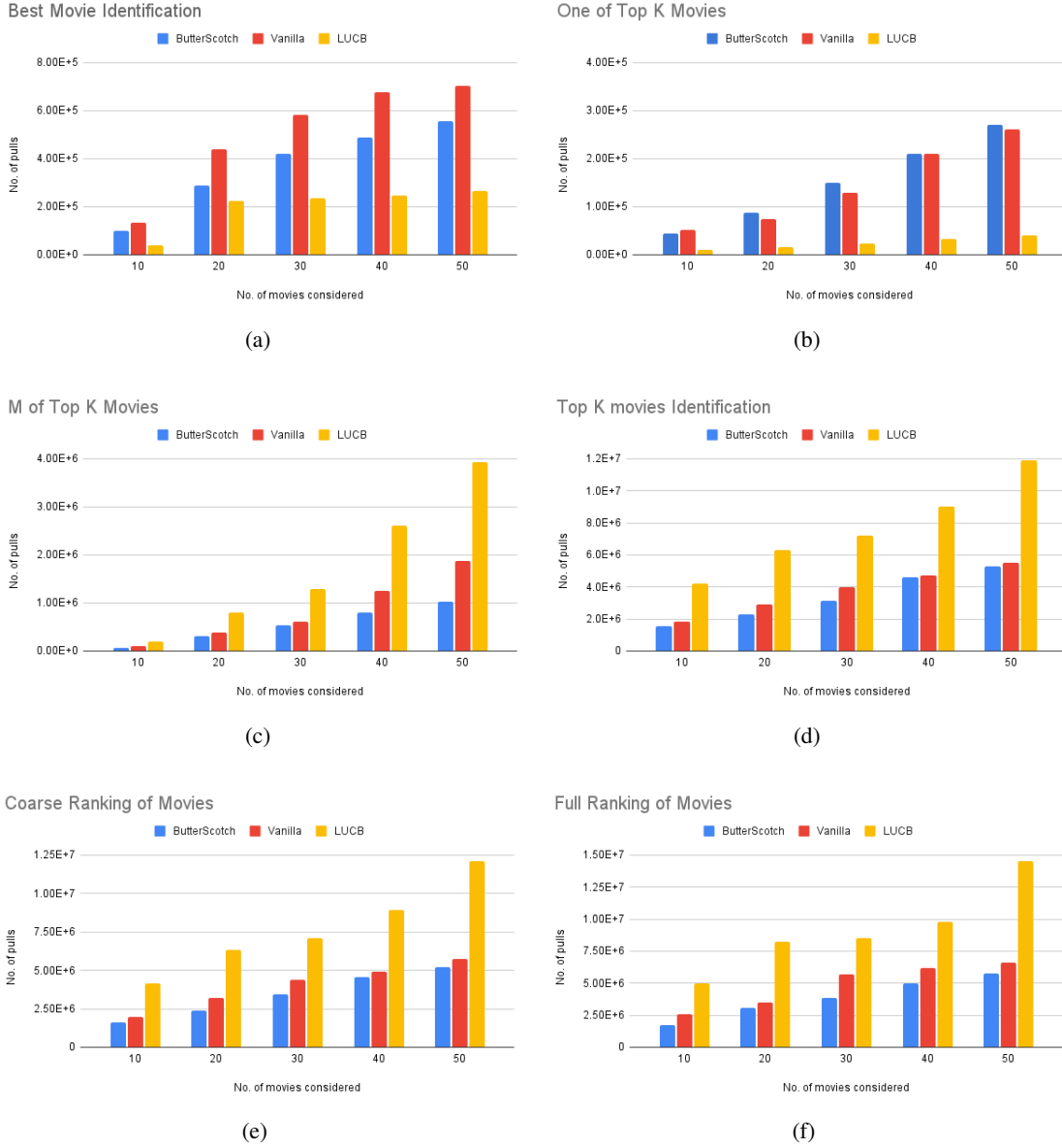
Figure 2: Comparision of sample complexity between Algorithm 1, Algorithm 2, and an LUCB-style scheme for special cases of the RAI problem, over an instance created from the MovieLens dataset

Table 3: Comparison of sample complexity between Algorithms 1, 2 and their versions which do not use adaptive merging of clusters

Sr.no	Vanilla	Vanilla (Non Merging)	Butterscotch	Butterscotch (Non Merging)
1	4588	6491	10370	10370
2	48632	50350	31757	44358
3	8958	10745	10370	10370
4	55932	56125	50162	51002
5	8913	10393	10370	10370

In addition to the synthetic dataset, we run our algorithms on the MovieLens dataset [19] which is a large database of movies and corresponding user ratings. Out of the approximately $27,000$ movies in the dataset, we shortlist a set of 50

movies that have received the highest number of ratings. We consider the movies as arms and their average user rating (normalized to [0,1]) as the corresponding mean reward. Each time we pull the arm associated with a movie, a random reward is generated which corresponds to the rating from a randomly chosen user. We experiment with different values of arms (N) ranging from 10 to 50, and consider all the special cases of the RAI problem stated in Table 1. These include best arm identification; identifying 1, M, and all of the top K arms; and finally coarse and full ranking. We select M and K to be 20% and 50% of N, respectively, and for the coarse ranking problem, the clusters are divided in the ratio of $3:5:2$. The results are presented in Figure 2.

From Figures 2(a),(b), the LUCB algorithm performs the best for the best arm and 1 out of the top K identification problems. This is in line with our earlier observation over the synthetic dataset in Table 2 where we found LUCB to be superior when the number of required arms is smaller. For the other four problems depicted in Figures 2(c),(d),(e),(f), the Vanilla and Butterscotch algorithms are significantly better than LUCB. The Butterscotch algorithm almost always performs the best, while providing a sizeable advantage over the Vanilla algorithm in many cases. The hardness of the various problems under consideration is also easily visible from the results; for example, for the hardest problem, i.e., getting a full ranking of movies, the number of pulls required is the maximum and is much larger than the problem of identifying 1 out of the top K movies.

6 Conclusion

We have proposed the representative arm identification (RAI) problem which generalizes several well studied problems in multi-armed bandits, including best arm identification, identifying M out of the top K arms, and coarse ranking. We provided a lower bound on the sample complexity of any reliable scheme and also proposed two algorithms based on the idea of confidence intervals. Upper bounds on the sample complexity of these algorithms were presented and their empirical performance was demonstrated over synthetic and real-world datasets.

Some questions remain open and several directions can be pursued in the future:

1. While Theorem 1 presents an instance-dependent and interpretable lower bound on the sample complexity of the RAI problem, it is in general loose. On the other hand, [17] provides a lower bound on problems with multiple correct answers (which includes the RAI problem) which can in general be tighter, but the bound is in the form of an optimization problem and is hard to compare against. Deriving a tighter lower bound which is still interpretable in terms of the problem parameters is a key open problem and can provide insights into multiple problems of interest given the many special cases that RAI covers.

2. *Federated RAI*: In this version of the problem, we have multiple clients, each having its own mean reward vector corresponding to the arms. Each client aims to solve its own local RAI problem, while a server which can communicate with the clients (at some cost) might be interested in solving a global RAI problem. A preliminary study of the problem indicates that a carefully crafted combination of the two algorithms proposed in this work can be used to solve the federated RAI problem, while ensuring low communication cost.

References

[1] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[2] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of a/b testing. In *Conference on Learning Theory*, pages 461–481. PMLR, 2014.

[3] Tong Geng, Xiliang Lin, Harikesh S Nair, Jun Hao, Bin Xiang, and Shurui Fan. Comparison lift: Bandit-based experimentation system for online advertising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15117–15126, 2021.

[4] Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.

[5] Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

[6] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

[7] Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5):1–24, 2017.

[8] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[9] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[10] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2014.

[11] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on learning theory-2010*, pages 13–p, 2010.

[12] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

[13] Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *International Conference on Machine Learning*, pages 1690–1698. PMLR, 2016.

[14] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *International conference on machine learning*, pages 1238–1246. PMLR, 2013.

[15] Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Pac identification of a bandit arm relative to a reward quantile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[16] Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Pac identification of many good arms in stochastic multi-armed bandits. In *International Conference on Machine Learning*, pages 991–1000. PMLR, 2019.

[17] Rémy Degenne and Wouter M Koolen. Pure exploration with multiple correct answers. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Nikolai Karpov and Qin Zhang. Batched coarse ranking in multi-armed bandits. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[19] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In *Proceedings of the fifth ACM conference on Recommender systems*, pages 387–388, 2011.

[20] Kota Srinivas Reddy, PN Karthik, and Vincent YF Tan. Almost cost-free communication in federated best arm identification. *arXiv preprint arXiv:2208.09215*, 2022.

[21] Faris Alzahrani and Ahmed Salem. Sharp bounds for the lambert w function. *Integral Transforms and Special Functions*, 29(12):971–978, 2018.

A Proof of Theorems

Before proceeding with the proofs, let us state the necessary concentration inequality used in proving Theorem 2 and Theorem 3

Lemma 1. *Let $X_1, X_2, \cdots, X_R \in \frac{1}{2}-SubGaussian$ be independent and identical random variables. Than for the empirical mean $\hat{\mu}(R) = \sum_{i=1}^{R} X_i / R$ we have*

$$P(|\hat{\mu}(R) - \mu| \geq \epsilon_R) \leq 2 \exp\left(-2R\epsilon_R^2\right)$$

Proof of Lemma 1. From Hoeffdings inequality for any σ^2-SubGaussian random variable, we have

$$P(|\hat{\mu}(R) - \mu| \geq \epsilon_R) \leq 2 \exp\left(-\frac{R\epsilon_R^2}{2\sigma^2}\right) \tag{1}$$

On substituting $\sigma = \frac{1}{2}$ in Equation 1, for any $\frac{1}{2}-$SubGaussian random variable, we get

$$P(|\hat{\mu}(R) - \mu| \geq \epsilon_R) \leq 2 \exp\left(-2R\epsilon_R^2\right) \tag{2}$$

A.1 Proof of Theorem 1

The proof of the lower bound on expected sample complexity starts from recognizing that the RAI problem fits in the 'multiple correct answers' framework of [17] and thus the general lower bound derived there applies to the RAI problem as well. However, that lower bound is in the form of a $\min \min \max$ optimization problem and the rest of the proof involves simplifying it to obtain an interpretable form.

To state the lower bound in [17], we have to introduce some additional notation. Consider an RAI problem instance $\mathcal{I} = (c, r, \Pi)$, where the arm reward distributions are Gaussian with standard deviation $1/2$ and the mean reward vector is given by $\mu := \{\mu_i^j\}$, where μ_i^j is the mean reward for arm j from cluster i under the current instance. Let $i^*[\mu]$ denote the set of all correct answers when the reward distributions are specified by μ. Note that each such correct answer corresponds to a set of arms such that r_1 of them belong to cluster 1, r_2 of them belong to cluster 2 and so on. Next, for any correct answer $a \in i^*[\mu]$, we need the notion of an *alternate mean reward vector* λ such that a is not a correct answer when the underlying arm reward distributions are Gaussian with standard deviation $1/2$ and the mean reward vector is $\lambda = \{\lambda_i^j\}$. We will denote the collection of all such alternate mean reward vectors by $\neg a$. Finally, let Δ_K denote the K-dimensional simplex and $d(a, b)$ denote the Kullback-Leibler (KL) divergence between two Gaussian distributions with means a and b, and variance $1/2$ each. Note that $d(a, b) = 2(a - b)^2$.

Next, from [17, Theorem 1], we have the following lower bound on the expected sample complexity of any δ-PC algorithm for an RAI problem $\mathcal{I} = (c, r, \Pi)$, where the arm reward distributions are Gaussian with variance $1/2$ and the mean reward vector is given by $\mu := \{\mu_i^j\}$:

$$\liminf_{\delta \to 0} \frac{E[T_\delta^{\mathcal{I}}(\mathcal{A})]}{\log(1/\delta)} \geq D(\mathcal{I})^{-1} \tag{3}$$

where $D(\mathcal{I}) = \max_{a \in i^*[\mu]} \max_{w \in \Delta_N} \inf_{\lambda \in \neg a} \sum_{i=1}^{m} \sum_{j=1}^{c_i} w_j^i d(\mu_j^i, \lambda_j^i)$.

Now, we will derive an upper bound on $D(\mathcal{I})$, which will yield the lower bound in the expression of Theorem 1. Consider any given correct answer $a \in i^*[\mu]$. Then, a particular alternate mean reward vector $\lambda \in \neg a$, can be constructed by shifting the mean reward for any one arm included in a, say arm k, so that its cluster membership is changed, and keeping all other mean rewards the same; see Figure 3 for an illustration. In particular, the minimum shift needed to do so is given by the arm gap for arm k, as defined in Definition 1. Amongst all the arms included in a, we will choose the one, say b_a, which requires the smallest change in mean reward to result in an alternate mean reward vector, i.e., for which the set of arms a is no longer a correct answer. Denote the corresponding change in the mean reward of arm b_a by l_a. Then we have

$$\begin{aligned}
D(\mathcal{I}) &= \max_{a \in i^*[\mu]} \max_{w \in \Delta_N} \inf_{\lambda \in \neg a} \sum_{i=1}^{m} \sum_{j=1}^{c_i} w_j^i d(\mu_j^i, \lambda_j^i) \\
&\leq \max_{a \in i^*[\mu]} \max_{w \in \Delta_N} w_{b_a} 2l_a^2 \\
&= \max_{a \in i^*[\mu]} 2l_a^2
\end{aligned}$$

11

$$= 2(\Delta_{\mathcal{I}})^2$$

where the last equality follows by recognizing that amongst all the correct answers $a \in i^*[\mu]$, the one that will need the biggest change in the mean reward of an arm to create an alternate reward distribution will be the one which contains the following: r_1 arms from cluster 1 with the highest arm gaps, r_2 arms from cluster 2 with the highest arm gaps, and so on; and the corresponding change will be equal to $\Delta_{\mathcal{I}}$ by Definition 2. Substituting the above bound in (3) completes the proof of Theorem 1.
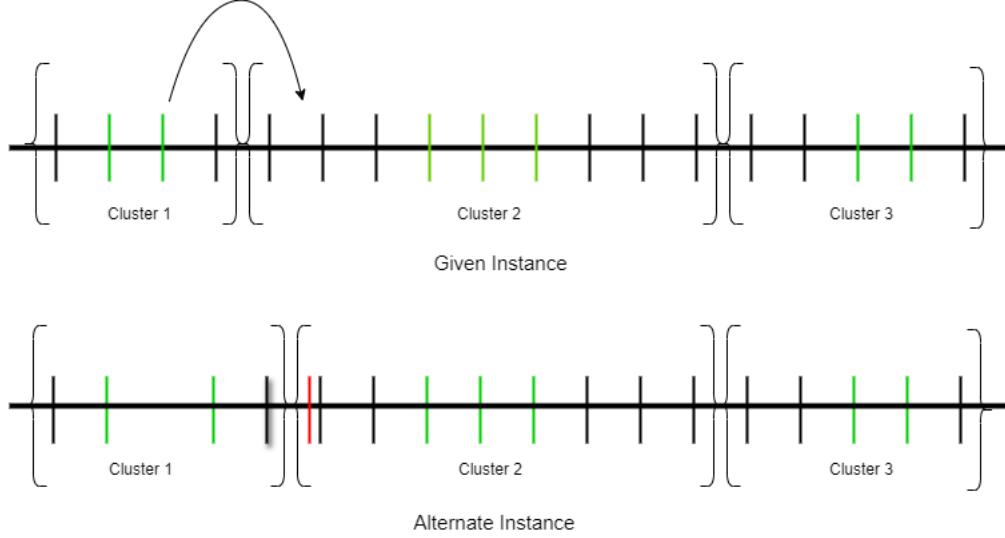

Figure 3: An illustration of alternate instance. In the given instance, the arms marked in green represent a set of correct answers. Now, to form an alternate instance, we shift an arm from cluster 1 to cluster 2, placing it just behind the boundary of cluster 2 in the given instance.

A.2 Proof of Theorem 2

The proof of this theorem is organized as follows. First, we define a good event ψ. Based on this condition, we then demonstrate the correctness of Claims 1 and 2.

$$\psi \triangleq \left\{ \forall R \in Z, \forall i \in [m], \forall j \in [c_i], |\hat{\mu}_j^i(R) - \mu_j^i| \leq \sqrt{\frac{\ln(\pi^2 R^2 N/3\delta)}{2R}} \right\} \tag{4}$$

Claim 1. Under $\mathcal{A} = Alg1$, let $\hat{\mu}_j^i(R)$ be the empirical mean of arm j from cluster i at round R. Let ψ be the event defined in 4. Than the $P_{\mathcal{I}}^{\mathcal{A}}(\psi) \geq 1 - \delta$

Proof of Claim 1. By the union bound, we have

$$P_{\mathcal{I}}^{\mathcal{A}}(\psi^c) \leq \sum_{R=1}^{\infty} \sum_{i=1}^{m} \sum_{j=1}^{c_i} P_{\mathcal{I}}^{\mathcal{A}} \left[|\hat{\mu}_j^i(R) - \mu_j^i| > \sqrt{\frac{\ln(\pi^2 R^2 N/3\delta)}{2R}} \right] \tag{5}$$

From Equation 2 and 5, we have

$$P_{\mathcal{I}}^{\mathcal{A}}(\psi^c) \leq \sum_{R=1}^{\infty} \sum_{i=1}^{m} \sum_{j=1}^{c_i} \frac{6\delta}{\pi^2 R^2 N} \tag{6}$$

$$\leq \sum_{R=1}^{\infty} \frac{6\delta}{\pi^2 R^2} \tag{7}$$

$$\leq \delta \tag{8}$$

Lemma 2. *Let $\mathcal{T}_{i,j}$ be the time step of $Alg1$ in which the identity of the arm j from cluster i is identified, conditioned on the good event (ψ), we have $\mathcal{T}_{i,j} \leq \mathcal{T}_{i,j}^{\mathcal{I}}$ where $\mathcal{T}_{i,j}^{\mathcal{I}}$ is defined as*

$$\mathcal{T}_{i,j}^{\mathcal{I}} = \frac{26}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi\sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}}\right) + 1$$

Proof of Lemma 2. The proof of Lemma 2 follows a similar approach to that of Lemma 3 in [20], differing only in the confidence intervals. However, for the reader's convenience, we provide the complete proof of the lemma here.

Consider the $Alg1$ identifies an arm j from cluster i in time step $\mathcal{T}_{i,j}$, then it must be the case that on the good event ψ, the following holds true

$$\min\left\{\hat{\mu}_{\text{arm}}^i(\mathcal{T}_{i,j}) - \hat{\mu}_1^{i+1}(\mathcal{T}_{i,j}), \hat{\mu}_{\bar{c}_{i-1}}^{i-1}(\mathcal{T}_{i,j}) - \hat{\mu}_{\text{arm}}^i(\mathcal{T}_{i,j})\right\} \geq 2\sqrt{\frac{\ln(\pi^2 \mathcal{T}_{i,j}^2 N/3\delta)}{2\mathcal{T}_{i,j}}}$$

Note that $\sqrt{\frac{\ln(\pi^2 R^2 N/3\delta)}{2R}} \to 0$ as $R \to \infty$.

Now, let $R_j^i := \inf\left\{R : \sqrt{\frac{\ln(\pi^2 R'^2 N/3\delta)}{2R'}} \leq \Delta_j^i/4 \; \forall R' \geq R\right\}$, then it must be that

$$\min\left\{\hat{\mu}_{\text{arm}}^i(R) - \hat{\mu}_1^{i+1}(R), \hat{\mu}_{\bar{c}_{i-1}}^{i-1}(R) - \hat{\mu}_{\text{arm}}^i(R)\right\} \geq 2\sqrt{\frac{\ln(\pi^2 R^2 N/3\delta)}{2R}}, \; \forall R > R_j^i \tag{9}$$

From equation 9, it follows that on the good event ψ, $\mathcal{T}_{i,j} < R_j^i$. We now derive an upper bound on R_j^i by letting

$$R_j^{i'} := \left\lceil \max\left\{n \in (1, \infty) : \sqrt{\frac{\ln(\pi^2 n^2 N/3\delta)}{2n}} = \Delta_j^i/4\right\}\right\rceil$$

The maximum in the above equation picks the largest solution for $n \in (1, \infty)$ satisfying $\sqrt{\frac{\ln(\pi^2 n^2 N/3\delta)}{2n}} = \Delta_j^i/4$, while ceil returning the smallest integer corresponding to n. We clearly see that $R_j^i \leq R_j^{i'}$.

Now, the exact expression for $R_j^{i'}$ is given by $R_j^{i'} = \left\lceil -\frac{1}{a} W_{-1}(-ae^{-b})\right\rceil$ where for $y < 0$, $W_{-1}(y)$ is the smallest value of x such that $xe^x = y$. Note that here $W_{-1}(y)$ refers to the Lambert W function [20]

Note that while all these steps are followed from [20], due to the difference in the confidence intervals in our case

$$a = \frac{\Delta_j^{i\,2}}{16} \text{ and } b = \frac{\ln\left(\frac{\pi^2 N}{3\delta}\right)}{2} \tag{10}$$

From [21], we also have

$$W_{-1}(y) > \frac{e}{e-1}\ln(-y) \tag{11}$$

Therefore, from equation 10 and 11, we have

$$\mathcal{T}_{i,j} \leq R_j^{i'}$$

$$\leq \left\lceil \frac{e}{e-1} \frac{b - \ln(a)}{a}\right\rceil$$

$$\leq \frac{e}{e-1} \frac{b - \ln(a)}{a} + 1$$

$$= \frac{16e}{e-1} \frac{1}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi\sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}}\right) + 1$$

$$= \frac{26}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi\sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}}\right) + 1$$

13

Claim 2. Conditioned on the good event ψ, the $Alg1$ takes no more than $\mathcal{T}_\delta^\mathcal{I}(Alg1)$ pulls to the solve the representative identification problem where

$$\mathcal{T}_\delta^\mathcal{I}(Alg1) \leq \sum_{i=1}^m \sum_{j=1}^{c_i} \left(\mathbb{1}\{\Delta_j^i \geq \Delta_\mathcal{I}\} \frac{26}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}} \right) + \mathbb{1}\{\Delta_j^i < \Delta_\mathcal{I}\} \frac{26}{\Delta_\mathcal{I}^2} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_\mathcal{I}^2} \right) + 1 \right)$$

Proof of Claim 2. Given the number of arms (N), the error probability (δ), and the arm gap (Δ_j^i), from Lemma 2, we know that for any arm j from cluster i, conditioned on the good event ψ, the maximum number of pulls for the arm by $Alg1$ is no more than $\mathcal{T}_{i,j}^\mathcal{I}$, where

$$\mathcal{T}_{i,j}^\mathcal{I} = \frac{26}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}} \right) + 1 \tag{12}$$

However, as at every round R, the algorithm pulls all the unidentified arms, the total number of pulls by $Alg1$ will be equivalent to the sum of the individual pulls. Now, from the definition of the Bottleneck gap (Refer Definition 2), we know that by the time the arm corresponding to the bottleneck gap gets identified, the cluster requirement would be satisfied resulting in the stopping of the algorithm. This leaves us with the 2 cases.

Case 1: When $\Delta_j^i \geq \Delta_\mathcal{I}$
In this case, the number of pulls $T_{i,j}^\mathcal{I}$ for arm j from cluster i is

$$\mathcal{T}_{i,j}^\mathcal{I} = \frac{26}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}} \right) + 1 \tag{13}$$

Case 2: When $\Delta_j^i < \Delta_\mathcal{I}$ For this case, as the algorithm terminates before identifying the identity of arm j from cluster i, we have

$$\mathcal{T}_{i,j}^\mathcal{I} = \frac{26}{\Delta_\mathcal{I}^2} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_\mathcal{I}^2} \right) + 1 \tag{14}$$

Finally, on summing up the number of pulls for every arm, we have

$$\mathcal{T}_\delta^\mathcal{I}(Alg1) \leq \sum_{i=1}^m \sum_{j=1}^{c_i} \left(\mathbb{1}\{\Delta_j^i \geq \Delta_\mathcal{I}\} \frac{26}{\Delta_j^{i\,2}} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_j^{i\,2}} \right) + \mathbb{1}\{\Delta_j^i < \Delta_\mathcal{I}\} \frac{26}{\Delta_\mathcal{I}^2} \ln\left(\frac{16\pi \sqrt{\frac{N}{3\delta}}}{\Delta_\mathcal{I}^2} \right) + 1 \right)$$

A.3   Proof of Theorem 3:

The proof of Theorem 3, follows a similar approach to that of Appendix A.2. We start with defining the good event, conditioned on which the rest of the proof follows.

$$\psi \triangleq \left\{ \forall R \in Z, \forall i \in [m], \forall j \in [c_i], |\hat{\mu}_j^i(R) - \mu_j^i| \leq 2^{-(R+3)} \right\} \tag{15}$$

Claim 3. Under $\mathcal{A} = Alg2$, let μ_j^i be the empirical mean of arm j from cluster i after t_R pulls, where t_R is defined in the line 2 of the $Alg2$. Given ψ be the good event defined in 15, we have $P_\mathcal{I}^\mathcal{A}(\psi) \geq 1 - \delta$

Proof of Claim 3. From union bounding, we have

$$P_\mathcal{I}^\mathcal{A}(\psi^c) \leq \sum_{R=1}^\infty \sum_{i=1}^m \sum_{j=1}^{c_i} P_\mathcal{I}^\mathcal{A}\left[|\hat{\mu}_j^i(R) - \mu_j^i| > \frac{2^{-R}}{8} \right] \tag{16}$$

Now, on using the equations 2 and 16, and from line 2 of the Algorithm 2, we have

$$P_\mathcal{I}^\mathcal{A}(\psi^c) \leq \sum_{R=1}^\infty \sum_{i=1}^m \sum_{j=1}^{c_i} \frac{6\delta}{\pi^2 R^2 N} \tag{17}$$

14

$$\leq \sum_{R=1}^{\infty} \frac{6\delta}{\pi^2 R^2} \tag{18}$$

$$\leq \delta \tag{19}$$

Lemma 3. *Given the good event, to identify the identity of arm j from cluster i, the Alg2, takes no more than $\mathcal{T}_{i,j}^{\mathcal{I}}$ pulls, where $\mathcal{T}_{i,j}^{\mathcal{I}}$ is defined as*

$$\mathcal{T}_{i,j}^{\mathcal{I}} = \max\left(\frac{32}{\Delta_j^{i\,2}} \ln\left(\frac{N\pi^2}{3\delta}\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right), 128\ln\left(\frac{N\pi^2}{3\delta}\right)\right)$$

Proof of Lemma 3. Following the argument in Lemma 2, we know that an arm j from cluster i will be identified with certainty at round R if $2^{-(R+3)} = \Delta_j^i/4$. Simplifying this further, we obtain:

$$R = \left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil \tag{20}$$

Next, from the definition of t_R and the value of R from equation20, we have

$$\mathcal{T}_{i,j}^{\mathcal{I}} = \ln\left(\frac{\pi^2 N}{3\delta}\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right) 2^{\left(2\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil + 5\right)}$$

$$\leq \ln\left(\frac{\pi^2 N}{3\delta}\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right) 2^{\left(2\left(\log_2\left(\frac{1}{2\Delta_j^i}\right) + 1\right) + 5\right)}$$

$$= \frac{128}{4\Delta_j^{i\,2}} \ln\left(\frac{\pi^2 N}{3\delta}\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right)$$

$$= \frac{32}{\Delta_j^{i\,2}} \ln\left(\frac{\pi^2 N}{3\delta}\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right)$$

However, since there are no pulls in round 0, it's essential to consider the total number of pulls at round 1 as part of the worst-case bound. Therefore, for any arm j, from cluster i, the total number of pulls will be bounded by

$$\mathcal{T}_{i,j}^{\mathcal{I}} = \max\left(\frac{32}{\Delta_j^{i\,2}} \ln\left(\frac{N\pi^2}{3\delta}\left\lceil \log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right), 128\ln\left(\frac{N\pi^2}{3\delta}\right)\right)$$

Claim 4. Conditioned on the good event ψ, the $Alg2$ takes no more than $\mathcal{T}_{\delta}^{\mathcal{I}}(Alg2)$ pulls to the solve the representative identification problem where

$$\mathcal{T}_{\delta}^{\mathcal{I}}(Alg2) \leq \sum_{i=1}^{m}\sum_{j=1}^{c_i}\left(\mathbb{1}\{\Delta_j^i \geq \Delta_{\mathcal{I}}\}\max\left(\frac{32}{\Delta_j^{i\,2}}\ln\left(\frac{N\pi^2}{3\delta}\left\lceil\log_2\left(\frac{1}{2\Delta_j^i}\right)\right\rceil^2\right), 128\ln\left(\frac{N\pi^2}{3\delta}\right)\right)\right.$$

$$\left.+\mathbb{1}\{\Delta_j^i < \Delta_{\mathcal{I}}\}\max\left(\frac{32}{\Delta_{\mathcal{I}}^2}\ln\left(\frac{N\pi^2}{3\delta}\left\lceil\log_2\left(\frac{1}{2\Delta_{\mathcal{I}}}\right)\right\rceil^2\right)\right), 128\ln\left(\frac{N\pi^2}{3\delta}\right)\right)$$

Proof of Claim 4. The proof of Claim 4 follows a similar argument as that of Claim 2

B LUCB

The LUCB version for the representative identification problem is inspired from [12] and is stated in Algorithm 3. In this algorithm, we have an intelligent sampling rule following a pull strategy based on the LCB and UCB of the arms. We start this section, by defining the upper and lower confidence bounds.

Definition 3. (LCB and UCB of the arms): Given $\hat{\mu}_j^i(R)$ be the empirical mean of arm j from cluster i, we define LCB_j^i at round R as

$$\text{LCB}_j^i(R) = \hat{\mu}_j^i(R) - \sqrt{\frac{\ln(\pi^2 R^3 N/3\delta)}{2R_j^i}}$$

and UCB_j^i at round R as

$$\text{UCB}_j^i(R) = \hat{\mu}_j^i(R) + \sqrt{\frac{\ln(\pi^2 R^3 N/3\delta)}{2R_j^i}}$$

where R_j^i is the number of times the arm j from cluster i till round R

Next, we identify the empirical gaps corresponding to every arm. These empirical gaps are defined in terms of the confidence bounds and play an important role in identifying the potential arms to pull.

Definition 4. (Empirical Gap): For all clusters $i \in [1, 2, \cdots, m]$, let $\text{LCB}^i(R)$ and $\text{UCB}^i(R)$ be

$$\text{LCB}^i(R) = \min(\text{LCB}_1^i(R), \cdots \text{LCB}_{c_i}^i(R))$$

and

$$\text{UCB}^i(R) = \max(\text{UCB}_1^i(R), \cdots \text{UCB}_{c_i}^i(R))$$

than the empirical gap $\hat{\Delta}_j^i$ for arm j from cluster i is defined as

$$\hat{\Delta}_j^i = \max(\text{UCB}_j^i(R) - \text{LCB}^{i-1}(R), \text{UCB}^{i+1}(R) - \text{LCB}_j^i(R)$$

Algorithm 3 LUCB styled Algorithm for RAI

Input: cluster sizes $c = (c_1, c_2, \cdots, c_m)$, required arms $r = (r_1, r_2, \cdots, r_m)$, arm set $\mathcal{N}$, error threshold δ
Output: $O_1, O_2, \cdots, O_m$

1: Initialize $R \leftarrow 0, A \leftarrow \mathcal{N}$, and for $i \in \{1, 2, \cdots, m\}$ set $\tilde{c}_i = c_i$
2: Sample every arm in A once
3: **while** $|O_1| \neq r_1$ or $|O_2| \neq r_2$ or $\cdots |O_m| \neq r_m$ **do**
4:     Increase R by 1
5:     Partition A into clusters $A_1, A_2, \cdots, A_m$ of sizes $\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_m$ respectively, based on the empirical means
6:     **for** i in $[m]$ **do**
7:       **for** arm in A_i **do**
8:         Calculate $\text{LCB}_{\text{arm}}^i(R)$ and $\text{UCB}_{\text{arm}}^i(R)$
9:         **if** $\text{LCB}(R)_{\text{arm}}^i > \text{UCB}(R)_1^{i+1}$ and $\text{UCB}_{\tilde{c}_{i-1}}^{i-1}(R) > \text{LCB}_{\text{arm}}^i(R)$ **then**
10:           **if** $|O_i| < r_i$ **then**
11:             Add arm to O_i
12:           **end if**
13:           Remove arm from A
14:           $\tilde{c}_i \leftarrow \tilde{c}_i - 1$
15:         **end if**
16:       **end for**
17:       **if** $O_i \neq r_i$ **then**
18:         $\text{LCB}^{i-1}(R) \leftarrow \min(\text{LCB}_1^{i-1}(R), \cdots \text{LCB}_{c_i}^{i-1}(R))$
19:         $\text{UCB}^{i-1}(R) \leftarrow \max(\text{UCB}_1^{i-1}(R), \cdots \text{UCB}_{c_i}^{i-1}(R))$
20:         $\hat{\Delta}^i(R) \leftarrow \min(\hat{\Delta}_1^i, \hat{\Delta}_2^i, \cdots \hat{\Delta}_{c_i}^i)$
21:         Pull the arms corresponding to $\text{LCB}^{i-1}(R), \text{UCB}^{i-1}(R)$ and $\hat{\Delta}^i(R)$
22:       **end if**
23:     **end for**
24: **end while**

The algorithm starts by sampling every arm once (Line 2) before using a smarter pull-based strategy. It then enters a loop which continues until enough arms are identified from every cluster. While in the loop the algorithm first divides all the arms into clusters based on their empirical (line 5). It then calculates the LCB and the UCB corresponding to every arm, based on which the membership criteria is checked in Line 9. Finally, depending on whether the required number of arms are identified or not, it than decides if there are any additional pulls required for that cluster.

Theorem 4. *The LUCB version for the representative identification problem stated in Algorithm 3 solves the problem with at least a probability of $1 - \delta$*

Proof of Theorem 4:

The proof of Theorem 4 follows a similar approach as that of Claim 1 with modified confidence bounds and an additional summation over R_j^i going from 1 to R. Basically on taking the union bound and using the inequality stated in 1, we have

$$P[\text{Bad Event}] \leq \sum_{R=1}^{\infty} \sum_{R_j^i=1}^{R} \sum_{i=1}^{m} \sum_{j=1}^{c_i} 2\exp\left(-2R_j^i\left(\sqrt{\frac{\ln(\pi^2 R^3 N/3\delta)}{2R_j^i}}\right)^2\right) \leq \delta$$