Exploratory Multi-Player Multi-Armed Bandits

Abstract

We study the exploratory Multi-Armed Bandit (MAB) problem in a setting where k players collaborate in order to identify an ε-optimal arm. Our motivation comes from recent employment of MAB models in large-scale, distributed web applications. Our results demonstrate a trade-off between the number of arm pulls required for the task, and the required amount of communication between the players. In particular, our main result shows that by allowing the k players to communicate only once, they are able to learn using only $\sqrt{k}$ times more arm pulls than what required in the single player setting. This implies that distributing the learning to k players gives rise to a factor $\sqrt{k}$ speedup in learning time as compared to a single player. We complement this result with a lower bound showing this is in general the best possible. On the other extreme, we present an algorithm that uses the same amount of arm pulls as needed for a single player with communication logarithmic in $1/\varepsilon$, as well as an algorithm that explicitly tradeoffs arm pulls with communication.

1. Introduction

Over the past few years, reinforcement learning in general and multi-armed bandit (MAB) algorithms in particular have been employed in an increasing amount of web applications. MAB algorithms choose between stories to showcase on media sites (Agarwal et al., 2009; 2008), among ranking algorithms of search results (Moon et al., 2010; Radlinski et al., 2008; Yue and Joachims, 2009), trigger ads (Chakrabarti et al., 2008), and more. In many of these applications, websites cannot be served from a single front-end machine. The sheer volume of user requests they face, along with the geographical diversity those request come from, require sites to use multiple front-end machines that are often dispersed geographically among multiple data centers.

Assume that a single machine cannot accommodate the throughput of requests; rather, k different servers are needed. We will refer to these as *players*. Each player is responsible for deciding which arms to pull for its share of requests, and for updating its model once rewards can be inferred from user interactions. This paper studies the tradeoffs between the communication among the k players, and their overall learning performance. At one extreme, if all players broadcast to each other each and every ⟨pull, reward⟩ pair as they happen, they can each simulate the decisions of a centralized algorithm. However, the load on each player would then be similar to the load of a centralized model, which we assumed to be prohibitive. At the other extreme, if the players never communicate, each will suffer the learning curve (i.e. number of pulls) required in order to attain good performance, thereby multiplying the cost of learning by k. We aim to quantify this tradeoff between inter-player communication and the cost of learning.

Following Audibert et al. (2010); Even-Dar et al. (2006), we focus on the problem of identifying a "good" arm with few arm pulls as possible (that is, minimizing the *simple regret*). Our objective is thus purely explorative, and specifically we do not care about the incurred costs or the involved regret. As discussed by Bubeck et al. (2009); Audibert et al. (2010), this is indeed the natural goal in many situations. In this work, we build upon the results known in the single-player exploratory setup, and in particular the Successive Elimination algorithm (Even-Dar et al., 2006), showing how they can be used in a multi-player setting in a communication-efficient way.

Our setting is accordingly as follows. The players aim to identify an ε-best (i.e., ε-optimal) bandit arm with high confidence. They are evaluated by the number of total arm pulls, aggregated across all k players, required for them to do so. The arm pulls are evenly divided among the k players, and are assumed to take place synchronously. In order to reach their common goal, players may communicate with each other. Fol-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

lowing Balcan et al. (2012), we measure communication by counting the number of *communication rounds* required by them in a protocol, where in a single round of communication each player broadcasts a single message to all the other players.

Our main results deal with the case where players are allowed to communicate only *once*. For this scenario, we present k-player algorithms that need only $\sqrt{k}$ times more arm pulls for identifying the best and ε-best bandit arm, as compared to the conventional, single-player setting. A single round of communication thus leads to an improvement of $\sqrt{k}$ factor in the required number of iterations over a single player. We complement our algorithmic results for the single round scenario with a lower bound on the required number of pulls, implying that our results are essentially optimal.

In addition, we investigate how many communication rounds are required for competing with the learning performance of a single-player algorithm. We present a k-player algorithm achieving the *same* performance as a centralized algorithm with communication depending only logarithmically on $1/\varepsilon$. Interestingly, Balcan et al. (2012) established a similar result in a distributed PAC setting using boosting algorithms. Our algorithm achieves the same logarithmic dependence of communication on $1/\varepsilon$ for the multi-player MAB problem, while enjoying the optimal, distribution-dependent sample complexity upper bound.

1.1. Related Work

The classic (single-player) Multi-Armed Bandits problem was first considered under the PAC model by Even-Dar et al. (2002), who proposed the *Successive Elimination* (SE) strategy for ε-optimal arm identification. Mannor and Tsitsiklis (2004) provided tight, distribution-dependent sample complexity lower bounds for several variants of the PAC model, showing that the SE algorithm is essentially optimal in terms of its sample complexity. The setting was revisited by Bubeck et al. (2009), that investigated the links between the simple and cumulative regret measures and indicated that regret-minimizing algorithms (like UCB of Auer et al. (2002)) are not well-suited for pure exploration tasks. More recently, Audibert et al. (2010) presented algorithms for best-arm identification, proved their optimality and demonstrated their effectiveness in simulations.

In a recent work, Gabillon et al. (2011) consider a multi-player (termed "multi-bandit") MAB problem similar to ours, in which each player has to identify the best arm. However, in their setting each player gets a different set of arms (with its own rewards) and the challenge is to distribute the limited resources at hand (namely T arm pulls) so as to maximize the confidence of all players simultaneously. In our setting, each player gets access to the same set of arms, with the joint goal of identifying the (ε-)best arm while communicating with each other as few times as possible. Another variant of a multi-player MAB problem is studied by Liu and Zhao (2010). In their setup, where the goal is to minimize the cumulative regret, players do not communicate but rather compete with each other: once two players pull the same arm at the same time, a "collision" occurs and they share the reward in some way. In contrast, we study how sharing observations helps players achieve their goal jointly.

A related setting was considered by Balcan et al. (2012), who considered a distributed PAC model and investigated the involved communication trade-offs. In their model, data is distributed between the various players and the goal is to learn well with respect to the overall (i.e. mixture) distribution of the data. Some of their results also focus on learning using only one round of communication. Our MAB setting features an additional difficulty of designing an *adaptive* sampling strategy for each player (i.e., *choosing* which arms to sample), without exchanging samples with other players. While their results do not have direct implications in our setting, the main theme is similar.

2. Problem Setup and Background

2.1. Model and Notation

Our basic model is a generalization of the classic Multi-Armed Bandit (MAB) model. In the Multi-Player Multi-Armed Bandit setting, there are $k \geqslant 1$ individual players. The players are given n arms, enumerated by $[n] := \{1, 2, \ldots, n\}$. Each arm $i \in [n]$ is associated with a reward, which is a binary random variable with expectation $p_i \in [0, 1]$. For convenience, we assume that the arms are ordered by their expected rewards, that is $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$. At every time step $t = 1, 2, \ldots, T$, each player pulls one arm of his choice and observes an independent sample of its reward. Each player may choose any of the arms, regardless of the other players and their actions. At the end of the game, each player must commit to a single arm.

In the *best-arm identification* version of the problem, the goal of a multi-player algorithm given some target confidence level $\delta > 0$, is that with probability at least $1-\delta$ *all* players correctly identify the best arm (i.e. the arm having the maximal expected reward). For sim-

plicity, we assume in this setting that the best arm is unique. Similarly, in the (ε, δ)-*PAC* variant the goal is that each player finds an ε-optimal (or "ε-best") arm, that is an arm i with $p_i \geqslant p_1 - \varepsilon$, with high probability. In this paper we focus on the more general (ε, δ)-PAC setup, which also includes best-arm identification for $\varepsilon = 0$. We use the term *sample complexity* to refer to the total number of arm pulls needed for achieving the players' goal (at the desired confidence).

During a *communication round*, each player may broadcast a message to all other players. A round may take place at any predefined time step, and is assumed not to cause any delay to the players, nor consume any time steps. While we do not restrict the size (in bits) of each message, in a reasonable implementation a message should not be larger than $O(n)$ bits.

We introduce the notation $\Delta_i := p_1 - p_i$ to denote the suboptimality gap of arm i, and occasionally use $\Delta_\star := \Delta_2$ for denoting the minimal gap. In the best-arm version of the problem, where we assume that the best arm is unique, we have $\Delta_i > 0$ for all i. When dealing with the (ε, δ)-PAC setup, we also consider the truncated gaps $\Delta_i^\varepsilon := \max\{\Delta_i, \varepsilon\}$. In this work, we are interested in deriving distribution-dependent bounds over the sample complexity of algorithms. Namely, our bounds are stated as a function of ε, δ and also the distribution-specific values $\Delta := (\Delta_2, \ldots, \Delta_n)$[1]. The $\tilde{O}$ notation in our sample complexity bounds hides poly-logarithmic factors in $n, k, \varepsilon, \delta$ and $\Delta_2, \ldots, \Delta_n$.

Clearly, a multi-player problem with $k = 1$ players is just the standard exploratory MAB problem. When $k > 1$, the problem becomes more challenging as we constrain the communication between the players. In the rest of the section we review relevant results known in the single-player case, and give an overview of our results for the k-player MAB setting.

2.2. Pure Exploration in Multi-Armed Bandits

Most state-of-the-art algorithms for exploration in MAB, including the *Successive Elimination* (Even-Dar et al., 2006) and *Successive Rejects* (Audibert et al., 2010) algorithms, are based on sequential elimination of suboptimal arms. Once an arm is eliminated, it is never pulled again and resources are directed towards the remaining arms.

The Successive Elimination (SE) algorithm, given ε and a target confidence $1 - \delta$, works in phases, each of which consists of uniform sampling of the remaining arms, followed by some elimination of the "worst" arms; see Algorithm 3 of Even-Dar et al. (2006). The SE algorithm enjoys the following sample complexity upper bound.

Theorem 2.1 (Even-Dar et al. 2006, Theorem 8)**.** *With probability at least $1 - \delta$, the Successive Elimination algorithm $\texttt{SE}(\varepsilon, \delta)$ needs at most*

$$O\left(\sum_{i=2}^{n} \frac{1}{(\Delta_i^\varepsilon)^2} \log \frac{n}{\delta \Delta_i^\varepsilon}\right) \tag{1}$$

arm pulls for terminating and identifying an ε-best arm.

The above result is essentially tight (up to poly-logarithmic factors), as implied by Theorem 5 of Mannor and Tsitsiklis (2004). Intuitively, the hardness of the task is therefore captured by the quantity

$$H_\varepsilon := \sum_{i=2}^{n} \frac{1}{(\Delta_i^\varepsilon)^2}\,, \tag{2}$$

which is roughly the number of arm pulls needed to find an ε-best arm with a reasonable probability; see also (Audibert et al., 2010) for a discussion.

2.3. Summary of Our Results

Our results are summarized in Table 1 below, which compares the different algorithms on both the sample complexity and communication fronts. As a baseline for evaluating our results, we consider two trivial methods. In the first ("1P baseline"), the k-players simulate the centralized SE strategy with confidence δ, thus communicating upon each time step. In the second approach ("kP naïve") each player independently executes SE with target confidence δ/k, which ensures that with probability $1 - \delta$ all players find an ε-best arm but pays an $\tilde{O}(k)$ factor in the sample complexity.

Algorithm	Sample Comp.	Communication
1P baseline	$\tilde{O}(H_\varepsilon)$	every time step
kP naïve	$\tilde{O}(k\, H_\varepsilon)$	none
Alg. 1,2	$\tilde{O}(\sqrt{k}\, H_\varepsilon)$	1 round
Alg. 3	$\tilde{O}(H_\varepsilon)$	$O(\log \frac{1}{\varepsilon})$ rounds
Alg. 3'	$\tilde{O}(\varepsilon^{-2/r}\, H_\varepsilon)$	r rounds

Table 1. Summary of our results.

Our first algorithmic result (Algorithm 1) deals with the case where only one round of communication is allowed, and shows an $\tilde{O}(\sqrt{k}\, H_0)$ upper bound on the

[1] If one is interested in distribution-free bounds, then the problem at hand is trivial as the (one-round) uniform sampling strategy is optimal in this setting (up to poly-logarithmic factors); see also (Mannor and Tsitsiklis, 2004) for a discussion.

number of pulls required for best-arm identification, thus improving upon the naïve approach by a $\sqrt{k}$ factor. We then extend this result to the (ε, δ)-PAC setup (in Algorithm 2), establishing an $\tilde{O}(\sqrt{k}\, H_\varepsilon)$ upper bound for the one-round case. We complement these results with a lower bound, showing that in general it is impossible to lower the $\sqrt{k}$ gap between the sample complexity of single-player and one-round k-player algorithms.

We then relax the communication constraint and allow multiple communication rounds to take place. We show that k players can reproduce the optimal $\tilde{O}(H_\varepsilon)$ bound achievable in the single-player setup, using only $\log_2(1/\varepsilon) + O(1)$ rounds. A rather simple modification of this result yields an algorithm that uses at most r rounds and $\tilde{O}(\varepsilon^{-2/r}\, H_\varepsilon)$ arm pulls overall, for any positive integer r.

3. One Communication Round

This section considers the most basic variant of the multi-player MAB problem, where only one round of communication is allowed at the end of the game. Here, given a budget of arm pulls, each player decides which arms to pull according to a policy based solely on his results. After exhausting the given budget, the players share their results and compute the output. For the clarity of exposition, we first present in Section 3.1 an algorithm for best-arm identification under this setting. Section 3.2 deals with the (ε, δ)-PAC setup. We demonstrate the tightness of our result in Section 3.3 with a lower bound for the required budget of arm pulls in the one round setting.

3.1. Best-arm Identification Algorithm

We now describe a one-round, multi-player MAB algorithm that needs only a $\tilde{O}(\sqrt{k})$ times more arm pulls than a single-player algorithm, thus improving upon the trivial $\tilde{O}(k)$ factor of the naive approach. For simplicity, we present a version matching $\delta = 1/3$, meaning that the algorithm produces the correct arm with probability at least $2/3$. We explain later how to expand it to deal with arbitrary values of δ. Our algorithm is akin to a majority vote among the multiple players, where each player pulls arms in two stages. In the first stage, each player independently solves a "smaller" MAB instance on a random subset of the arms using Successive Elimination[2]. In the second stage, each player exploits the arm identified as best in the first stage, and communicates that arm and its observed reward. See Algorithm 1 below for a precise description. An appealing feature of our algorithm is that it requires each player to communicate merely 2 numerical values to the other players.

Although our goal is pure exploration, each player in the algorithm follows an explore-exploit strategy. The idea here is that a player should sample his recommended arm as much as his budget permits, even if it was easy to identify in his small-sized problem (i.e., did not require many pulls). This way we can guarantee that the top arms are sampled sufficiently-many times by the time the players have to agree on a single best arm.

Algorithm 1 One-round Best-arm

input time horizon T
output an arm

1: **for** player $j = 1$ to k **do**
2: choose a subset A_j of $6n/\sqrt{k}$ arms uniformly at random
3: `explore`: execute $i_j \leftarrow \texttt{SE}(A_j, 0, \frac{1}{3})$ using at most $\frac{1}{2}T$ pulls (and halting the algorithm early if necessary); if the algorithm fails to identify any arm or does not terminate gracefully, let i_j be an arbitrary arm
4: `exploit`: pull arm i_j for $\frac{1}{2}T$ times and let $\hat{q}_j$ be its average reward
5: communicate the numbers $i_j, \hat{q}_j$
6: **end for**
7: let k_i be the number of players j with $i_j = i$, and define $A = \{i : k_i > \sqrt{k}\}$
8: let $\hat{p}_i = (1/k_i) \sum_{\{j\,:\,i_j = i\}} \hat{q}_j$ for all i
9: output $\arg\max_{i \in A} \hat{p}_i$; if the set A is empty, return an arbitrary arm.

In Theorem 3.1 we prove that Algorithm 1 indeed achieves the promised upper bound.

Theorem 3.1. *Algorithm 1 identifies the best arm correctly with probability at least $2/3$ using no more than*

$$O\left(\sqrt{k} \cdot \sum_{i \neq i^\star} \frac{1}{\Delta_i^2} \log \frac{n}{\Delta_i}\right) \tag{3}$$

arm pulls, provided that $6 \leqslant \sqrt{k} \leqslant n$. The algorithm uses a single communication round, in which each player communicates 2 numerical values.

By repeating the algorithm $O(\log(1/\delta))$ times and taking the majority vote of the independent instances, we can amplify the success probability to $1 - \delta$ for any given $\delta > 0$. Note that we can still do that with one

[2] We note that any (optimal) best-arm identification strategy can be used as a building-block in our algorithm, and the choice of Successive Elimination here is rather arbitrary.

communication round (at the end of all executions), but each player now has to communicate $O(\log(1/\delta))$ values[3].

Theorem 3.2. *There exists a k-player algorithm that given*

$$O\left(\sqrt{k}\cdot\sum_{i\neq i^\star}\frac{1}{\Delta_i^2}\log\frac{n}{\Delta_i}\log\frac{1}{\delta}\right)$$

arm pulls, identifies the best arm correctly with probability at least $1-\delta$. The algorithm uses a single communication round, in which each player communicates $O(\log(1/\delta))$ numbers.

3.1.1. Analysis

We will show that a budget T of samples (arm pulls) per player, where

$$T \geqslant \frac{24c}{\sqrt{k}}\cdot\sum_{i\neq i^\star}\frac{1}{\Delta_i^2}\ln\frac{3n}{\Delta_i}\,, \tag{4}$$

suffices for the players to jointly identify the best arm $i^\star$ with the desired probability. Here, c denotes the constant under the big O notation in eq. (1), and without loss of generality we assume that $c \geqslant 1$. Clearly, this would imply the sample complexity bound stated in Theorem 3.1. Note that we did not optimize the constants in the above expression.

We begin by analyzing the `explore` phase of the algorithm. Our first lemma shows that each player chooses the global best arm and identifies it as the local best arm with sufficiently large probability.

Lemma 3.3. *When (4) holds, each player identifies the (global) best arm correctly after the* `explore` *phase with probability at least $2/\sqrt{k}$.*

Proof. Let $H = \sum_{i\neq i^\star}(1/\Delta_i^2)\ln(3n/\Delta_i)$ and $H_j = \sum_{i^\star\neq i\in A_j}(1/\Delta_i^2)\ln(3n/\Delta_i)$ for all j. Then $\mathbf{E}[H_j \mid i^\star \in A_j] \leqslant 6H/\sqrt{k}$ by the linearity of expectation, and Markov's inequality thus gives that $\Pr[H_j \leqslant 12H/\sqrt{k} \mid i^\star \in A_j] \geqslant 1/2$. Clearly, we also have $\Pr[i^\star \in A_j] = 6/\sqrt{k}$ which implies that

$$\Pr[i^\star \in A_j \text{ and } H_j \leqslant 12H/\sqrt{k}] \geqslant 3/\sqrt{k}\,. \tag{5}$$

Now consider the "local" MAB problem facing player j, over the subset of arms A_j. If the (global) best arm $i^\star$ is amongst the arms in A_j, then by Theorem 2.1 the `SE` algorithm player j executes needs no more than

$$T_j := c\sum_{i^\star\neq i\in A_j}\frac{1}{\Delta_i^2}\ln\frac{3n}{\Delta_i} = cH_j$$

pulls in order to identify $i^\star$ successfully with probability $2/3$. In case that $H_j \leqslant 12H/\sqrt{k}$, we have $T_j \leqslant 12cH/\sqrt{k} \leqslant T/2$, which means that the pulls budget of player j suffices for identifying the best arm. Together with (5), we conclude that with probability at least $2/\sqrt{k}$ player j identifies the best arm correctly. □

We next address the `exploit` phase. The next simple lemma shows that the popular arms (i.e. those selected by many players) are estimated to a sufficient precision.

Lemma 3.4. *Provided that (4) holds, with probability at least $5/6$ we have $|\hat{p}_i - p_i| \leqslant \frac{1}{2}\Delta_\star$ for all arms $i \in A$.*

Proof. Consider some arm $i \in A$. The estimate $\hat{p}_i$ is the average reward of $k_iT/2 \geqslant \sqrt{k}T/2 \geqslant (2/\Delta_\star^2)\ln(12n)$ arm pulls (of the `exploit` phase). Hoeffding's inequality now gives that

$$\Pr[|\hat{p}_i - p_i| > \tfrac{1}{2}\Delta_\star] \leqslant 2\exp(-\tfrac{1}{2}\Delta_\star^2\cdot k_iT/2) \leqslant 1/6n\,,$$

and the lemma follows via a union bound. □

We can now prove Theorem 3.1.

Proof. Let us first show that with probability at least $5/6$, the best arm i is contained in the set A. To this end, notice that $k_{i^\star}$ is the sum of k i.i.d. Bernoulli random variables $\{I_j\}_j$ where I_j is the indicator of whether player j chooses arm $i^\star$ after the `explore` phase. By Lemma 3.3 we have that $\mathbf{E}[I_j] \geqslant 2/\sqrt{k}$ for all j, hence by Hoeffding's inequality,

$$\begin{aligned}\Pr\left[k_{i^\star} \leqslant \sqrt{k}\right] &\leqslant \Pr\left[\frac{1}{k}\sum_{j=1}^{k}(I_j - \mathbf{E}[I_j]) \leqslant \frac{-1}{\sqrt{k}}\right]\\ &\leqslant \exp(-2k/k) \leqslant 1/6\end{aligned}$$

which implies that $i^\star \in A$ with probability at least $5/6$.

Next, note that with probability at least $5/6$ the arm $i \in A$ having the highest empirical reward $\hat{p}_i$ is the one with the highest expected reward p_i. Indeed, this follows directly from Lemma 3.4 that shows that with probability at least $5/6$, for all arms $i \in A$ the estimate $\hat{p}_i$ is within $\frac{1}{2}\Delta$ of the true bias p_i.

[3] In fact, by letting each player pick a slightly larger subset of $O(\sqrt{\log(1/\delta)}\cdot n/\sqrt{k})$ arms, we can amplify the success probability to $1-\delta$ without needing to communicate more than 2 values per player. However, this approach only works when $k = \Omega(\log(1/\delta))$.

Hence, via a union bound we conclude that with probability at least 2/3, the best arm is in A and has the highest empirical reward. In other words, with probability at least 2/3 the algorithm outputs the best arm $i^\star$. □

3.2. (ε, δ)-PAC Algorithm

We now present Algorithm 2 whose purpose is to recover an ε-optimal arm. The algorithm is similar to Algorithm 1 of the previous section, but now each player attempts to find an $O(\varepsilon)$-best arm in a subsampled problem. Note, however, that there may be more than one ε-best arm, so each "successful" player might come up with a different ε-best arm. Nevertheless, our analysis shows that, with high probability, a subset of the players can still arrive to a consensus on a single ε-best arm.

In Theorem 3.5 we prove that our algorithm needs only $\tilde{O}(\sqrt{k})$ times more arm pulls than a single-player algorithm for identifying an $O(\varepsilon)$-best arm.

Algorithm 2 One-round ε-arm

input time horizon T, accuracy ε
output an arm

1: **for** player $j = 1$ to k **do**
2: choose a subset A_j of $12n/\sqrt{k}$ arms uniformly at random
3: `explore`: execute $i_j \leftarrow \mathtt{SE}(A_j, \varepsilon, \frac{1}{3})$ using at most $\frac{1}{2}T$ pulls (and halting the algorithm early if necessary); if the algorithm fails to identify any arm or does not terminate gracefully, let i_j be an arbitrary arm
4: `exploit`: pull arm i_j for $\frac{1}{2}T$ times, and let $\hat{q}_j$ be the average reward
5: communicate the numbers $i_j, \hat{q}_j$
6: **end for**
7: let k_i be the number of players j with $i_j = i$
8: let $t_i = \frac{1}{2}k_i T$ and $\hat{p}_i = (1/k_i)\sum_{\{j\,:\,i_j=i\}} \hat{q}_j$ for all i
9: define $A = \{i \in [n] \,:\, t_i \geqslant (4/\varepsilon^2)\ln(12n)\}$
10: output $\arg\max_{i\in A} \hat{p}_i$; if the set A is empty, return an arbitrary arm.

Theorem 3.5. *Algorithm 2 identifies a 2ε-best arm with probability at least 2/3 using no more than*

$$O\left(\sqrt{k}\cdot\sum_{i\neq i^\star}\frac{1}{(\Delta_i^\varepsilon)^2}\log\frac{n}{\Delta_i^\varepsilon}\right) \tag{6}$$

arm pulls, provided that $24 \leqslant \sqrt{k} \leqslant n$. The algorithm uses a single communication round, in which each player communicates 2 numerical values.

3.2.1. Analysis

Before proving the theorem, we first state several key lemmas. In the following, let n_ε and $n_{2\varepsilon}$ denote the number of ε-best and 2ε-best arms respectively. Our analysis considers two different regimes: $n_{2\varepsilon} \leqslant \frac{1}{50}\sqrt{k}$ and $n_{2\varepsilon} > \frac{1}{50}\sqrt{k}$, and shows that in any case,

$$T \geqslant \frac{400c}{\sqrt{k}}\sum_{i\neq i^\star}\frac{1}{(\Delta_i^\varepsilon)^2}\ln\frac{24n}{\Delta_i^\varepsilon} \tag{7}$$

suffices for identifying a 2ε-best arm with the desired probability. Again, we use c to denote the constant under the big O notation in eq. (1) and assume that $c \geqslant 1$. Clearly, this implies the sample complexity bound stated in eq. (6).

The first lemma shows that at least one of the players is able to find an ε-best arm. As we later show, this is sufficient for the success of the algorithm in case there are many 2ε-best arms.

Lemma 3.6. *When (7) holds, at least one player successfully identifies an ε-best arm in the* `explore` *phase, with probability at least 5/6.*

The next lemma is more refined and states that in case there are few 2ε-best arms, the probability of each player to successfully identify an ε-best arm grows linearly with n_ε.

Lemma 3.7. *Assume that $n_{2\varepsilon} \leqslant \frac{1}{50}\sqrt{k}$. When (7) holds, each player identifies an ε-best arm in the* `explore` *phase, with probability at least $2n_\varepsilon/\sqrt{k}$.*

The last lemma we need analyzes the accuracy of the estimated rewards of arms in the set A.

Lemma 3.8. *With probability at least 5/6, we have $|\hat{p}_i - p_i| \leqslant \varepsilon/2$ for all arms $i \in A$.*

For the proofs of the above lemmas, refer to the supplementary material. We now turn to prove Theorem 3.5.

Proof. We shall prove that with probability 5/6 the set A contains at least one ε-best arm. This would complete the proof, since Lemma 3.8 assures that with probability 5/6, the estimates $\hat{p}_i$ of all arms $i \in A$ are at most $\varepsilon/2$-away from the true reward p_i, and in turn implies (via a union bound) that with probability 2/3 the arm $i \in A$ having the maximal empirical reward $\hat{p}_i$ must be a 2ε-best arm.

First, consider the case $n_{2\varepsilon} > \frac{1}{50}\sqrt{k}$. Lemma 3.6 shows that with probability 5/6 there exists a player j that identifies an ε-best arm i_j. Since for at least $n_{2\varepsilon}$ arms $\Delta_i \leqslant 2\varepsilon$, we have

$$t_{i_j} \geqslant \tfrac{1}{2}T \geqslant \frac{400}{2\sqrt{k}}\cdot\frac{n_{2\varepsilon}-1}{(2\varepsilon)^2}\ln\frac{24n}{2\varepsilon} \geqslant \frac{1}{\varepsilon^2}\ln(12n)\,,$$

that is, $i_j \in A$.

Next, consider the case $n_{2\varepsilon} \leqslant \frac{1}{50}\sqrt{k}$. Let N denote the number of players that identified some ε-best arm. The random variable N is a sum of Bernoulli random variables $\{I_j\}_j$ where I_j is an indicator to whether player j identified some ε-best arm. By Lemma 3.7, $\mathbf{E}[I_j] \geqslant 2n_\varepsilon/\sqrt{k}$ and thus by Hoeffding's inequality,

$$\Pr\left[N < n_\varepsilon\sqrt{k}\right] = \Pr\left[\frac{1}{k}\sum_{j=1}^{k}(I_j - \mathbf{E}[I_j]) \leqslant -\frac{n_\varepsilon}{\sqrt{k}}\right] \leqslant \exp(-2n_\varepsilon^2) \leqslant \frac{1}{6}\,.$$

That is, with probability 5/6, at least $n_\varepsilon\sqrt{k}$ players found an ε-best arm. A pigeon-hole argument shows that in this case there exists an ε-best arm $i^\star$ selected by at least $\sqrt{k}$ players. Hence, with probability 5/6 the number of samples of this arm collected in the `exploit` phase is at least $t_{i^\star} \geqslant \sqrt{k}T/2 > (1/\varepsilon^2)\ln(12n)$, which means that $i^\star \in A$. □

3.3. Lower Bound

The following theorem suggests that in general, for identifying the best arm k players need at least $\tilde{\Omega}(\sqrt{k})$ times the budget required for a single player to do so, when only allowed one round of communication (at the end of the game). Clearly, this also implies that a similar lower bound holds in the PAC setup, and proves that our algorithmic results for the one-round case are essentially tight.

Theorem 3.9. *there exist rewards $p_1, \ldots, p_n$ and integer T such that*

- *each player in the algorithm must use at least $T/\sqrt{k}$ arm pulls for them to collectively identify the best arm with probability at least 2/3;*
- *there exist a single-player algorithm that needs at most $\tilde{O}(T)$ pulls for identifying the best arm with probability at least 2/3.*

The proof of the theorem is deferred to the supplementary material.

4. Multiple Communication Rounds

This section establishes a general dependence of communication on $1/\varepsilon$, and shows a tradeoff between the sample complexity of a multi-player algorithm and the number of communication rounds it uses, in terms of $1/\varepsilon$.

4.1. $O(\log(1/\varepsilon))$-rounds Algorithm

We first show that by allowing $O(\log(1/\varepsilon))$ rounds of communication, we are able to attain the optimal sample complexity as in the single-player scenario. That is, we do not gain any improvement in learning performace from allowing more than $O(\log(1/\varepsilon))$ rounds.

Our algorithm is based on a variant of Successive Elimination. The idea is to eliminate in each round r (i.e., right after the rth communication round) all 2^{-r}-suboptimal arms. We accomplish that by letting each player sample uniformly all remaining arms and communicate the results to other players. Then, players are able to eliminate suboptimal arms with high confidence. If each such round is successful, after $\log_2(1/\varepsilon)$ rounds only the best arm survives. The algorithm is given in detail as Algorithm 3 below, and in Theorem 4.1 we state its sample complexity guarantee.

Algorithm 3 MULTI-PLAYER SE

input (ε, δ)
output an arm
define $\varepsilon_r = 2^{-r}$
let $t_0 = 0$ and $t_r = (2/k\varepsilon_r^2)\ln(4nr^2/\delta)$ for $r \geqslant 1$
initialize $S_0 \leftarrow [n]$, $r \leftarrow 0$
repeat
  set $r \leftarrow r + 1$
  for player $j = 1$ to k **do**
    sample each arm $i \in S_{r-1}$ for $t_r - t_{r-1}$ times
    let $\hat{p}^r_{j,i}$ be the average reward of arm i (in all rounds so far of player j)
    communicate the numbers $\hat{p}^r_{j,1}, \ldots, \hat{p}^r_{j,n}$
  end for
  let $\hat{p}^r_i = (1/k)\sum_{j=1}^{k}\hat{p}^r_{j,i}$ for all $i \in S_{r-1}$, and let $\hat{p}^r_\star = \max_{i\in S_{r-1}}\hat{p}^r_i$
  set $S_r \leftarrow S_{r-1} \setminus \{i \in S_{r-1} : \hat{p}^r_i < \hat{p}^r_\star - \varepsilon_r\}$
until $\varepsilon_r \leqslant \varepsilon/2$ or $|S_r| = 1$
output S_r

Theorem 4.1. *With probability at least $1-\delta$, Algorithm 3*

- *identifies the optimal arm using*

$$O\left(\sum_{i=2}^{n}\frac{1}{(\Delta_i^\varepsilon)^2}\log\left(\frac{n}{\delta}\log\frac{1}{\Delta_i^\varepsilon}\right)\right)$$

  arm pulls;
- *terminates after at most $1+\lceil\log_2(1/\varepsilon)\rceil$ rounds of communication (or after $1+\lceil\log_2(1/\Delta)\rceil$ rounds for $\varepsilon = 0$).*

Proof. Without loss of generality, we may assume that the rewards of all arms are drawn before the algorithm

is executed, so that the empirical averages $\hat{p}_i^r$ are defined for all arms at all rounds (even for arms that were eliminated prior to some round). Since $\hat{p}_i^r$ is the empirical average of kt_r samples of arm i (aggregated from all players), for any round r and arm i we have by Hoeffding's inequality,

$$\Pr[|\hat{p}_i^r - p_i| \geqslant \varepsilon_r/2] \leqslant 2\exp(-\varepsilon_r^2 k t_r/2) = \delta/2nr^2\,.$$

Hence, a union bound gives that $|\hat{p}_i^r - p_i| < \varepsilon_r/2$ for all i and r with probability at least

$$1 - \sum_{r=1}^{\infty}\sum_{i=1}^{n}\frac{\delta}{2nr^2} = 1 - \sum_{r=1}^{\infty}\frac{\delta}{2r^2} \geqslant 1-\delta\,.$$

That is, with probability at least $1-\delta$, an ε-optimal arm i is never eliminated by the algorithm, as the event $\hat{p}_i^r < \hat{p}_\star^r - \varepsilon_r$ implies that either $\hat{p}_i^r < p_i - \varepsilon_r/2$ or $\hat{p}_j^r > p_j + \varepsilon_r/2$ for some arm j. In addition, any suboptimal arm i does not survive round $r_i = \lceil \log_2(1/\Delta_i^\varepsilon)\rceil + 1$, since $\Delta_i^\varepsilon \geqslant 2\varepsilon_{r_i}$ and so for $r = r_i$,

$$\begin{aligned}\hat{p}_i^r &< p_i + \varepsilon_r/2 = p_1 + \varepsilon_r/2 - \Delta_i \leqslant \hat{p}_1^r + \varepsilon_r - \Delta_i \\ &\leqslant \hat{p}_1^r - \varepsilon_r \leqslant \hat{p}_\star^r - \varepsilon_r\,.\end{aligned}$$

That is, with probability at least $1-\delta$, after $\lceil\log_2(1/\varepsilon)\rceil + 1$ rounds (when the algorithm terminates) all remaining arms are ε-optimal. When $\varepsilon = 0$, the algorithm terminates once only a single arm survives, and with high probability this occurs after at most $\lceil\log_2(1/\Delta)\rceil + 1$ rounds.

We conclude by computing the sample complexity. Let T_i be the total number of times arm $i \neq 1$ is pulled. Since $r_i \leqslant \log_2(4/\Delta_i^\varepsilon)$, we have

$$\begin{aligned}T_i &\leqslant 2(2^{r_i})^2 \ln\frac{4nr_i^2}{\delta} \\ &\leqslant 2\left(\frac{4}{\Delta_i^\varepsilon}\right)^2 \ln\left(\frac{4n}{\delta}\log_2^2\frac{4}{\Delta_i^\varepsilon}\right) \\ &= O\left(\frac{1}{(\Delta_i^\varepsilon)^2}\log\left(\frac{n}{\delta}\log\frac{1}{\Delta_i^\varepsilon}\right)\right).\end{aligned}$$

Consequently, the total number of arm pulls is $T_2 + \sum_{i=2}^n T_i$, which gives the theorem. □

4.2. R-rounds Algorithm

By properly tuning the elimination thresholds ε_r of Algorithm 3 in accordance with the target accuracy ε, we can trade off between the number of communication rounds and the sample complexity of the algorithm. In particular, we can design a multi-player algorithm that terminates after at most R communication rounds, for any given parameter $R > 0$. This, however, comes at the cost of a compromise in sample complexity as quantified in the following theorem.

Theorem 4.2. *Given a parameter $R > 0$, set $\varepsilon_r = \varepsilon^{r/R}$ for all $r \geqslant 0$ in Algorithm 3. With probability at least $1-\delta$, the modified algorithm*[4]

- *identifies an ε-optimal arm using*

$$O\left(\frac{1}{\varepsilon^{2/R}}\cdot\sum_{i=2}^{n}\frac{1}{(\Delta_i^\varepsilon)^2}\log\frac{nR}{\delta}\right)$$

  arm pulls;

- *terminates after at most R rounds of communication.*

Proof. For all arms $i \in [n]$, let

$$r_i = \left\lceil R\,\frac{1+\log_2(1/\Delta_i^\varepsilon)}{\log_2(1/\varepsilon)}\right\rceil.$$

Since $\Delta_i \geqslant 2\varepsilon_{r_i}$, if the algorithm is successful any arm i which is not ε_{r_i}-optimal is eliminated after at most r_i rounds. Clearly, after R rounds only ε-optimal arms survive. This happens with probability at least $1-\delta$.

It remains to calculate the sample complexity. It is easy to verify that $2\varepsilon_{r_i} \geqslant \varepsilon^{1/R}\cdot\Delta_i^\varepsilon$, thus the number of times arm i was pulled is

$$T_i = \frac{4}{\varepsilon_{r_i}^2}\ln\frac{2nr_i^2}{\delta} = O\left(\frac{1}{\varepsilon^{2/R}}\cdot\frac{1}{(\Delta_i^\varepsilon)^2}\log\frac{nR}{\delta}\right),$$

and the theorem follows. □

5. Conclusions and Future Work

We have considered a multi-player multi-armed bandits problem, where arms are pulled, and rewards are observed, by several independent players with a common goal. In order to learn efficiently, the players must communicate. We show an inherent tradeoff between the communication among the players, and the overall number of arm pulls required to solve the problem (less communication implies more arm pulls).

We leave the following for future work. Most importantly, we wish to translate our findings to the regret minimization setting. Furthermore, it would be interesting to extend our single-round results to the case where r communication rounds are allowed, for any constant $r > 1$. Specifically, we would like to quantify the communication-pulls tradeoff in terms of the number of players k (and independently of ε). Finally, we wish to investigate lower bounds on arm pulls for the case of multiple rounds of communication.

[4] In contrast to other algorithms discussed in this paper, this algorithm does not apply to $\varepsilon = 0$. In order to apply it in a best-arm identification setting, one should set $0 < \varepsilon \leqslant \Delta$, but this would require knowing the value of Δ (or some lower bound thereof) beforehand.

References

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Nitin Motgi, Seung-Taek Park, Raghu Ramakrishnan, Scott Roy, and Joe Zachariah. Online models for content optimization. In *Proc. 22nd Annual Conference on Neural Information Processing Systems (NIPS'2008)*, pages 17–24, December 2008.

Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Explore/exploit schemes for web content optimization. In *Proc. Ninth IEEE International Conference on Data Mining (ICDM'2009)*, pages 1–10, 2009.

M. Anthony and P. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 1999.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53, 2010.

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

M.F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. *Arxiv preprint arXiv:1204.3514*, 2012.

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pages 23–37. Springer, 2009.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Proc. 22nd Annual Conference on Neural Information Processing Systems (NIPS'2008)*, pages 273–280, 2008.

E. Even-Dar, S. Mannor, and Y. Mansour. PAC bounds for multi-armed bandit and markov decision processes. In *COLT*, pages 193–209. Springer, 2002.

E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105, 2006.

Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sebastien Bubeck. Multi-bandit best arm identification. In *Advances in Neural Information Processing Systems 24*, pages 2222–2230. 2011.

Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, Nov. 2010.

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

Taesup Moon, Lihong Li, Wei Chu, Ciya Liao, Zhaohui Zheng, and Yi Chang. Online learning for recency search ranking using real-time user feedback. In *Proc. 9th ACM Conference on Information and Knowledge Management (CIKM'2010)*, pages 1501–1504, October 2010.

Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proc. 17th ACM Conference on Information and Knowledge Management (CIKM'2008)*, pages 43–52, October 2008.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proc. 26th Annual International Conference on Machine Learning (ICML'2009)*, page 151, June 2009.