Optimal Streaming Algorithms for Multi-Armed Bandits

Tianyuan Jin [1] **Keke Huang** [1] **Jing Tang** [2] **Xiaokui Xiao** [1]

Abstract

This paper studies two variants of the best arm identification (BAI) problem under the streaming model, where we have a stream of n arms with reward distributions supported on $[0, 1]$ with unknown means. The arms in the stream are arriving one by one, and the algorithm cannot access an arm unless it is stored in a limited size memory.

We first study the streaming ε-*top*-k arms identification problem, which asks for k arms whose reward means are lower than that of the k-th best arm by at most ε with probability at least $1 - \delta$. For general $\varepsilon \in (0, 1)$, the existing solution for this problem assumes $k = 1$ and achieves the optimal sample complexity $O(\frac{n}{\varepsilon^2} \log \frac{1}{\delta})$ using $O(\log^*(n))$ [1] memory and a single pass of the stream. We propose an algorithm that works for any k and achieves the optimal sample complexity $O(\frac{n}{\varepsilon^2} \log \frac{k}{\delta})$ using a single-arm memory and a single pass of the stream.

Second, we study the streaming BAI problem, where the objective is to identify the arm with the maximum reward mean with at least $1 - \delta$ probability, using a single-arm memory and as few passes of the input stream as possible. We present a single-arm-memory algorithm that achieves a near instance-dependent optimal sample complexity within $O(\log \Delta_2^{-1})$ passes, where Δ_2 is the gap between the mean of the best arm and that of the second best arm.

1. Introduction

Best arm identification (BAI) is a classic decision problem with numerous applications such as medical trials (Thompson, 1933), online advertisement (Bertsimas & Mersereau, 2007), and crowdsourcing (Zhou et al., 2014). It typically considers a bandit with a set of arms, each of which has a reward distribution with an unknown mean. The objective is to identify the best arm with the maximum reward mean.

Due to applications with massive data, the BAI problem has been recently studied under the streaming model in the literature (Assadi & Wang, 2020; Falahatgar et al., 2020; Maiti et al., 2020), where only a limited size of memory is available for storing arms. In addition, BAI under the streaming model also avoids a large amount of time/money on switching alternatives and thus finds numerous applications. For example, in recruitment, employers aim to select the most qualified employee among all candidates with high probability. For this purpose, they could query each candidate with sufficient number of questions to acquire an accurate evaluation with confidence. The more questions they ask, the more confidence they have on the candidate's evaluation. Once the interview ends, usually, the candidate will not be asked for further evaluation. In manufacturing, switching alternatives might require reassembling the production line, which could incur excessive costs.

Motivated by above observations, in this paper, we study two problems, i.e., *streaming ε-top-k arms identification (ε-KAI)* and *streaming BAI*.

(Problem 1) Streaming ε-KAI. In streaming ε-KAI, we have a stream of n arms, such that each *arm*$_i$ is associated with an unknown reward distribution supported on $[0, 1]$ with an unknown mean μ_{arm_i}. The arms in the stream are arriving one by one, and we can pull an arm only when it is stored in the memory. Given parameters $\varepsilon, \delta \in (0, 1)$, the task is to identify k arms whose reward means are lower than that of the k-th best arm by at most ε with probability at least $1 - \delta$. The ultimate goal in this paper is to minimize the sample complexity using a single-arm memory and a single pass over the stream.

(Problem 2) Streaming BAI. In streaming BAI, the task is to identify the *optimal* arm with the largest mean with probability at least $1-\delta$, using a single-arm memory, assuming that there exists a unique optimal arm. Streaming BAI can be regarded as a special case of ε-KAI with $\varepsilon = 0$ and $k = 1$, Again, we aim to minimize the sample complexity using a single-arm memory and as few passes of the input

[1]School of Computing, National University of Singapore, Singapore [2]Data Science and Analytics Thrust, The Hong Kong University of Science and Technology, Guangzhou, China. Correspondence to: Xiaokui Xiao <xkxiao@nus.edu.sg>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1]$\log^*(n)$ equals the number of times that we need to apply the logarithm function on n before the results is no more than 1.

stream as possible.

1.1. State of the Art

Streaming ε-BAI. The ε-BAI problem under the streaming model is pioneered by Assadi & Wang (2020), for which a single-pass streaming algorithm is proposed that can achieve the optimal sample complexity of $O(\frac{n}{\varepsilon^2}\log\frac{1}{\delta})$ using $O(\log^*(n))$ memory. When $\varepsilon \le \Delta_2$, they further devise a single-pass algorithm using memory for pulling 2 arms (i.e., the current arriving arm in the stream, and the candidate arm currently stored) while achieving the same sample complexity, where Δ_i is the difference between the expected rewards of k-th best and the i-th best arms for any $i > k$ and $k = 1$ for ε-BAI. Maiti et al. (2020) reveal that if the arms arrive in a random order, the requirement of $\varepsilon \le \Delta_2$ can be discarded experimentally. However, if the arms arrive in some specific sequences, the correctness of the algorithm is not guaranteed. Moreover, both algorithms may revisit a candidate arm tested previously based on the sample outcomes of other arriving arms, which are often undesirable in practice. For example, during an interview process, an employer cannot repeatedly test a candidate based on the outcomes of other applicants, since a candidate is usually waiting at home for the final result after attending an interview. Falahatgar et al. (2020) propose an algorithm that tests each arm in a strictly first-in-first-out (FIFO) order but still assumes a random-order arrival of the arms.

Streaming ε-KAI. To the best of our knowledge, the work by Assadi & Wang (2020) is the only one that studies the general streaming ε-KAI problem. Under the assumption that $\varepsilon \le \Delta_{k+1}$, Assadi & Wang (2020) propose an algorithm that achieves the optimal sample complexity using $O(k)$ memory. Again, their algorithm suffers from two major deficiencies that it (i) does not test each arm in a FIFO order and (ii) requires Δ_{k+1} to be known in advance, which are unrealistic in many practical applications.

1.2. Our Contributions

As our main result, we address the aforementioned shortcomings of existing algorithms for the general streaming ε-KAI problem and also study the streaming BAI problem which aims to identify the best arm strictly. The results are summarized in Table 1.

Streaming ε-KAI. We propose a single-pass algorithm for ε-KAI that achieves the optimal sample complexity using a *single-arm memory*, i.e., we pull an arm only at the time that it arrives and never revisit it after we pull other arms.[2] Our solution significantly improves upon the algorithms by Assadi & Wang (2020) in the way that (i) it does not rely on any assumption on ε, and (ii) it requires only a single-arm memory for the general ε-KAI problem.

[2]Note that we ignore the memory cost of storing the IDs of arms; otherwise, any algorithm for ε-KAI requires $\Omega(k)$ memory for recording the IDs of the arms to be returned.

Streaming BAI. We present an algorithm for streaming BAI that optimizes the sample complexity. Given any constant δ, it achieves a near instance-dependent optimal sample complexity of $O\big(\sum_{i=2}^{n}\frac{1}{\Delta_i^2}\log\big(\frac{1}{\delta}\log\frac{1}{\Delta_i}\big)\big)$ using a single-arm memory and $O(\log\frac{1}{\Delta_2})$ passes in expectation.

2. Single-Arm Memory Algorithm for ε-BAI

In this section, we present our solution for the streaming ε-BAI problem, i.e., ε-KAI with $k = 1$. We then extend our solution to address the ε-KAI problem for the general case of k in Section 3.

2.1. High Level Overview

Let *arm*^{o} be the selected arm, and *arm*$_i$ be the i-th arm in the stream where $i \in \{1, 2, \ldots, n\}$. Let μ_{arm} and $\widehat{\mu}_{arm}$ be *arm*'s true mean and estimated mean respectively, and *arm** be the best arm. In the first step, we initialize *arm*^{o} with *arm*$_1$. When *arm*$_i$ arrives, we compare *arm*$_i$ with *arm*^{o} and decide whether *arm*$_i$ should be the new *arm*^{o}. In particular, our algorithm mainly consists of the following two operations.

1. **Sampling.** We pull each arrived arm $\Theta(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ times to estimate its true mean. This number of pull is sufficient to ensure that $\widehat{\mu}_{arm^*}$ approaches μ_{arm^*} within $O(\varepsilon)$ with high probability[3], i.e., $|\widehat{\mu}_{arm^*} - \mu_{arm^*}| \le O(\varepsilon)$.
2. **Comparison.** We replace *arm*^{o} with *arm*$_i$ if $\widehat{\mu}_{arm_i} \ge \widehat{\mu}_{arm^o} + \alpha$, where $\alpha = \Theta(\varepsilon)$ is a random variable following a predefined distribution.

Our algorithm maintains the following property.

Property 1. *Whenever we update arm^o, we always ensure that $|\widehat{\mu}_{arm^o} - \mu_{arm^o}| \le O(\varepsilon)$.*

Based on the above two operations and Property 1, we could prove that $|\mu_{arm^o(T)} - \mu_{arm^*}| \le O(\varepsilon)$ holds where *arm*$^o(T)$ is the final returned arm. The basic idea is as follows. Operation one ensures that $|\widehat{\mu}_{arm^*} - \mu_{arm^*}| \le O(\varepsilon)$. Operation two guarantees that $\widehat{\mu}_{arm^o(T)} \ge \widehat{\mu}_{arm^*} + \alpha$ if *arm*$^o(T)$ is not *arm**. In the meantime, $|\widehat{\mu}_{arm^o(T)} - \mu_{arm^o(T)}| \le O(\varepsilon)$ holds according to Property 1. As a consequence, $|\mu_{arm^o(T)} - \mu_{arm^*}| \le O(\varepsilon)$ is established.

As indicated above, the correctness of our algorithm lies in Property 1. When each *arm*$_i$ is pulled $\Theta(\frac{1}{\varepsilon^2}\log\frac{cj^2}{\delta})$ times, we have $|\widehat{\mu}_{arm_i} - \mu_{arm_i}| \le O(\varepsilon)$ with probability at least $1-\frac{\delta}{cj^2}$ according to Hoeffding bound, where j is the number of arms that current *arm*^{o} beats and c is a constant. Then by *union bound*, Property 1 holds with probability at least

[3]We omit *with high probability* in the following for expression simplicity.

Problem	Algorithm	Requirement	Memory	#passes
ε-BAI	Assadi & Wang (2020)	No	$O(\log^*(n))$	1
	Assadi & Wang (2020)	$\varepsilon \le \Delta_2$	2	1
	Falahatgar et al. (2020)	Random-order arrival	1	1
	This paper (Algorithm 1)	No	1	1
ε-KAI	Assadi & Wang (2020)	$\varepsilon \le \Delta_{k+1}$	$O(k)$	1
	This paper (Algorithm 2)	No	1	1
BAI	**This paper (Algorithm 3)**	No	1	$O(\log \Delta_2^{-1})$

Table 1. Comparisons of streaming algorithms for ε-KAI with the optimal sample complexity of $O(\frac{n}{\varepsilon^2}\log\frac{k}{\delta})$ and BAI with sample complexity of $O\big(\sum_{i=2}^{n}\frac{1}{\Delta_i^2}\log\big(\frac{1}{\delta}\log\frac{1}{\Delta_i}\big)\big)$.

Algorithm 1: Streaming ε-BAI

```
Input: $\varepsilon$, $\delta$, and a stream of $n$ arms.
Output: The index of an arm.
initialize $arm^o \leftarrow arm_1$ and $j \leftarrow 1$;
pull $arm^o$ $s_1$ times;
foreach arriving $arm_i$ $(i > 1)$ do
    set $\alpha$ as $\frac{\varepsilon}{4}$ with probability $\frac{1}{\log j+1}$, and as $\frac{\varepsilon}{2}$ with other probability $1-\frac{1}{\log j+1}$;
    $\ell \leftarrow 1$;
    while true do
        pull $arm_i$ for $s_\ell - s_{\ell-1}$ times;
        if $\widehat{\mu}_{arm_i} \ge \widehat{\mu}_{arm^o} + \alpha$ and $s_\ell > \tau_j$ then
            $arm^o \leftarrow arm_i$;
            $j \leftarrow 1$;
            break;
        else if $\widehat{\mu}_{arm_i} < \widehat{\mu}_{arm^o} + \alpha$ then
            $j \leftarrow j+1$;
            break;
        else
            $\ell \leftarrow \ell + 1$;
return the index of $arm^o$;
```

$1-\sum_{j=1}^{\infty}\frac{\delta}{cj^2} \ge 1-\delta$.

In what follows, we highlight how our algorithm achieves the optimal sample complexity when Property 1 is maintained. As mentioned, each arm_i would be pulled $\Theta(\frac{1}{\varepsilon^2}\log\frac{j^2}{\delta})$ times before it could replace current arm^o. However, for arms with relatively small means, pulling such number of times is inefficient since we could identify and remove them with less pulls. In this regard, we pull arm_i through multiple rounds. In the ℓ-th round, it is pulled $\Theta(\frac{2^\ell}{\varepsilon^2}\log\frac{1}{\delta})$ times. This round loop terminates immediately once either the number of pulls reaches $\Theta(\frac{1}{\varepsilon^2}\log\frac{j^2}{\delta})$ or we are able to decide arm_i is not the best arm and then eliminate it.

Another aspect to optimize sample complexity lies in the design of α. Actually, this part is the main *hardness* of our algorithm. One conventional method is to set a fixed value to α. However, this would lead to suboptimal sample complexity. To explain, suppose we set $\alpha = \frac{\varepsilon}{2}$. If some arm_i has $\mu_{arm_i} = \widehat{\mu}_{arm^o} + \frac{\varepsilon}{2}$, it is inappropriate to bound the probability $\Pr(\widehat{\mu}_{arm_i} \ge \widehat{\mu}_{arm^o} + \frac{\varepsilon}{2})$ according to Hoeffding inequality. To fix this, a straightforward method is to bound the number of pulls of arm_i by $\Theta(\frac{1}{\varepsilon^2}\log(\frac{j^2}{\delta}))$. However, when j grows to $\Theta(n)$, this method would incur the total sample complexity of $O\big(\frac{n}{\varepsilon^2}\log\frac{n}{\delta}\big)$, which is suboptimial. To bypass this intractable issue, we leverage the *power of randomization*. That is, we set $\alpha = \frac{\varepsilon}{4}$ with probability $\frac{1}{\log j+1}$ and $\alpha = \frac{\varepsilon}{2}$ with probability $1-\frac{1}{\log j+1}$. We elaborate the details later.

The proof of the optimal sample complexity is highly nontrivial and is also one of our *main technical* contributions. We refer readers to Appendix A for details.

2.2. The Algorithm

We first introduce two parameters used in our algorithm.

$$\{s_\ell\}_{\ell=1}^{\infty}: \quad s_\ell = \frac{16}{\varepsilon^2}\cdot\log\left(\frac{C}{\delta}\right)\cdot 2^\ell, \text{ and } s_0 = 0, \qquad (1)$$

$$\{\tau_j\}_{j=1}^{\infty}: \quad \tau_j := \frac{32}{\varepsilon^2}\cdot\log\left(\frac{C\cdot j^2}{\delta}\right), \qquad (2)$$

where ℓ indicates the ℓ-th round and $C \ge 100$ is a universal constant.

Algorithm 1 presents the pseudo-code of our algorithm. In the beginning, we initialize $arm^o = arm_1$ with the first arm arm_1, and then pull arm^o s_1 times to obtain its estimated mean $\widehat{\mu}_{arm^o}$. In what follows, for each arrived arm_i in the stream, we sample α from the distribution defined as

$$\Pr\left(\alpha = \frac{\varepsilon}{4}\right) = \frac{1}{\log j + 1} \text{ and } \Pr\left(\alpha = \frac{\varepsilon}{2}\right) = 1 - \frac{1}{\log j + 1},$$

where j is the number of arms beaten by arm^o. Next, arm_i is compared with arm^o in multiple rounds. In the ℓ-th round, *arm*$_i$ will be pulled $s_\ell - s_{\ell-1}$ times to obtain its estimated mean $\widehat{\mu}_{\mathit{arm}_i}$. If both conditions $\widehat{\mu}_{\mathit{arm}_i} \geq \widehat{\mu}_{\mathit{arm}^o} + \alpha$ and $s_\ell > \tau_j$ hold, (i) *arm*^{o} is replaced by *arm*$_i$, (ii) j is reset to 1, and (iii) current round terminates. Otherwise, we would check whether the condition $\widehat{\mu}_{\mathit{arm}_i} < \widehat{\mu}_{\mathit{arm}^o} + \alpha$ meets. If it is true, *arm*$_i$ will be removed immediately. Meanwhile, j is increased by 1 and the round ends. If none of the two events on round termination happen, we increase index ℓ by 1 and then enter the next round. The above procedure is repeated for each arriving arm until all arms in the stream have been scrutinized. Eventually, we return the index of the final *arm*^{o}.

2.3. The Analysis

We say that an arm is ε*-best* arm if its mean is smaller than that of the best arm *arm** by at most ε, i.e., $\mu_{\mathit{arm}^*} - \mu_{\mathit{arm}} \leq \varepsilon$. We formalize our main result for ε-BAI problem as follows.

Theorem 1. *Given a stream of n arms, approximation parameter ε and confidence parameter δ in $(0,1)$, Algorithm 1 finds the ε-best arm with probability at least $1-\delta$ using expected $O(\frac{n}{\varepsilon^2}\log\frac{1}{\delta})$ pulls and a single-arm memory.*

Let *best arm change* be the event that *arm*^{o} is replaced by another arm, and *arm*$^o(t)$ denote the resulting arm after best arm change happens exactly t times (Note that *arm*$^o(1) =$ *arm*$_1$). We denote *arm*$^o(T)$ as the final returned arm^o. In what follows, we focus on the correctness proof of Algorithm 1, i.e., $\mu_{\mathit{arm}^o(T)} \geq \mu_{\mathit{arm}^*} - \varepsilon$ holds with probability at least $1-\delta$. The proof consists of two parts. In the first part, we establish the relation between all *arm*^{o} and *arm** in Lemma 1. In the second part, we then complete the correctness proof based on the result of Lemma 1.

Proposition 1 (Hoeffding Inequality)**.** *Let $X_1, \ldots, X_m$ be m independent random variables with support in $[0,1]$. Define $X := \sum_{i=1}^m X_i$. Then, for $x > 0$,*

$$\Pr(X - \mathbb{E}[X] > x) \leq 2 \cdot \exp\left(-\frac{2x^2}{m}\right).$$

Lemma 1. *For any $\varepsilon, \delta \in (0,1)$, it holds in Algorithm 1 that*

$$\Pr\left(\bigcap_{t\geq 1}\left\{\left\{\widehat{\mu}_{\mathit{arm}^o(t)} < \mu_{\mathit{arm}^*} - \frac{5\varepsilon}{8}\right\} \bigcup \left\{\mu_{\mathit{arm}^o(t)} \geq \mu_{\mathit{arm}^*} - \varepsilon\right\}\right\}\right) \geq 1 - \frac{3\delta}{4}.$$

The proof for Lemma 1 is conducted in three steps. In *Step I*, for one specific *arm*^{o}, we prove that its estimated mean differs from its true mean by at most $\Theta(\varepsilon)$. In *Step II*, we extend this result for all *arm*^{o} in general (Property 1) and bound the failure probability within δ. In *Step III*, we then derive Lemma 1 based on the results in previous steps.

Proof of Lemma 1. We prove the lemma by three steps.

Step I. For $t = 1$, Algorithm 1 pulls *arm*$^o(1)$ s_1 times. From Proposition 1, for $r \in \mathbb{N}^+$, we have

$$\begin{aligned}
&\Pr\left(|\widehat{\mu}_{\mathit{arm}^o(1)} - \mu_{\mathit{arm}^o(1)}| \geq \frac{r\varepsilon}{8}\right) \leq 2\exp\left(-\frac{s_1 r^2 \varepsilon^2}{32}\right) \\
&= 2\exp\left(-r^2 \log\left(\frac{C}{\delta}\right)\right) = \frac{2\delta^{r^2}}{C^{r^2}} \leq \frac{2\delta}{C^r}. \qquad (3)
\end{aligned}$$

Let $Q_{t,p}$ be the p-th passed arm after t-th best arm change, and $s(p) := s_\ell$ such that $s_{\ell-1} < \tau_p \leq s_\ell$. For ease of analysis, we design a virtual sampling process for a better illustration. Notably, if $Q_{t,p}$ is pulled less than τ_p times when Algorithm 1 ends, we pull $Q_{t,p}$ again to $s(p)$ times (a virtual process). Therefore, for all $p \geq 1$, $Q_{t,p}$ will be pulled $s(p)$ times to obtain its estimated mean, denoted as $\widehat{\mu}'_{Q_{t,p}}$. If *arm*$^o(t+1) = Q_{t,p}$, then $\widehat{\mu}'_{Q_{t,p}} = \widehat{\mu}_{Q_{t,p}}$ holds according to the definition. Let $F^o(t)$ be the union of history till the t-th best arm change. Then, conditioned on any $F^o(t)$, we have

$$\begin{aligned}
&\left\{|\widehat{\mu}_{Q_{t,p}} - \mu_{Q_{t,p}}| \geq \frac{r\varepsilon}{8}, \mathit{arm}^o(t+1) = Q_{t,p}\right\} \\
&\subseteq \left\{|\widehat{\mu}'_{Q_{t,p}} - \mu_{Q_{t,p}}| \geq \frac{r\varepsilon}{8}\right\}. \qquad (4)
\end{aligned}$$

Based on equation (4), for all $p \geq 1$, we have

$$\begin{aligned}
&\Pr\left(|\widehat{\mu}_{\mathit{arm}^o(t+1)} - \mu_{\mathit{arm}^o(t+1)}| \geq \frac{r\varepsilon}{8} \,\middle|\, F^o(t)\right) \\
&= \sum_{p=1}^{\infty} \Pr\left(\left\{|\widehat{\mu}_{Q_{t,p}} - \mu_{Q_{t,p}}| \geq \frac{r\varepsilon}{8}\right\} \bigcap \left\{\mathit{arm}^o(t+1) = Q_{t,p}\right\} \,\middle|\, F^o(t)\right) \\
&\leq \sum_{p=1}^{\infty} \Pr\left(|\widehat{\mu}'_{Q_{t,p}} - \mu_{Q_{t,p}}| \geq \frac{r\varepsilon}{8} \,\middle|\, F^o(t)\right) \\
&\leq \sum_{p=1}^{\infty} 2\exp\left(-\frac{\tau_p r^2 \varepsilon^2}{32}\right) \leq \sum_{p=1}^{\infty} \frac{2\delta}{p^2 \cdot C^r} \\
&\leq \frac{4\delta}{C^r}. \qquad (5)
\end{aligned}$$

Step II. Next we extend the above result for all $\mu_{\mathit{arm}^o(t)}$. Let

$$\mathcal{S}_r(t) = \left\{\mu_{\mathit{arm}^o(q)} : \mu_{\mathit{arm}^o(q)} \in \left(\mu_{\mathit{arm}^*} - \frac{r\varepsilon}{8}, \mu_{\mathit{arm}^*} - \frac{(r-1)\varepsilon}{8}\right], \text{ and } q \in [t]\right\},$$

where r is an integer and $r \geq 1$. Let r_t be the index associated with $\mu_{\mathit{arm}^o(t)}$ such that $\mu_{\mathit{arm}^o(t)} \in \mathcal{S}_{r_t}(T)$. Let E_t be the event

$$\left\{|\widehat{\mu}_{arm^o(t)} - \mu_{arm^o(t)}| \le \frac{(r_t-8)\varepsilon}{8}, r_t \ge 9\right\} \bigcap \left\{|\widehat{\mu}_{arm^o(t)} - \mu_{arm^o(t)}| \le \frac{r_t\varepsilon}{8}, r_t < 9\right\}.$$

Let E_t^c be the complement of E_t. As indicated from (5), for $r_t \ge 9$, we have

$$\begin{aligned}&\Pr(E_t^c \mid \cap_{q=1}^{t-1} E_q)\\ &= \Pr\left(|\widehat{\mu}_{arm^o(t)} - \mu_{arm^o(t)}| \ge \frac{(r_t-8)\varepsilon}{8} \;\middle|\; \bigcap_{q=1}^{t-1} E_q\right) \quad (6)\\ &\le \frac{4\delta}{C^{r_t-8}}. \quad (7)\end{aligned}$$

Similarly, for $r_t < 9$, we have

$$\begin{aligned}&\Pr(E_t^c \mid \cap_{q=1}^{t-1} E_q)\\ &= \Pr\left(|\widehat{\mu}_{arm^o(t)} - \mu_{arm^o(t)}| \ge \frac{r_t\varepsilon}{8} \;\middle|\; \bigcap_{q=1}^{t-1} E_q\right) \quad (8)\\ &\le \frac{4\delta}{C^{r_t}}. \quad (9)\end{aligned}$$

Define event $E = \bigcap_{t=1}^{T} E_t$. Therefore, by chain rule we have

$$\Pr(E) = \prod_{t=1}^{T} \Pr\left(E_t \mid \cap_{q=1}^{t-1} E_q\right). \quad (10)$$

Conditioned on $\cap_{q=1}^{t} E_q$, we will prove that the number of arms in $\mathcal{S}_r(t)$ is at most $r+2$. Conditioned on $\cap_{q=1}^{t} E_q$ and $arm^o(t) \in \mathcal{S}_r(t)$, we have

$$\begin{aligned}\widehat{\mu}_{arm^o(t)} &\le \mu_{arm^o(t)} + \frac{r\varepsilon}{8}\\ &\le \mu_{arm^*} + \frac{r\varepsilon}{8} - \frac{(r-1)\varepsilon}{8}\\ &= \mu_{arm^*} + \frac{\varepsilon}{8}, \quad (11)\end{aligned}$$

and

$$\begin{aligned}\widehat{\mu}_{arm^o(t)} &\ge \mu_{arm^o(t)} - \frac{r\varepsilon}{8}\\ &\ge \mu_{arm^*} - \frac{r\varepsilon}{8} - \frac{r\varepsilon}{8}\\ &= \mu_{arm^*} - \frac{r\varepsilon}{4}, \quad (12)\end{aligned}$$

where the second inequalities of (11) and (12) are from the definition of $\mathcal{S}_r(t)$, respectively. Let $U = \mu_{arm^*} + \frac{\varepsilon}{8}$ and $L = \mu_{arm^*} - \frac{r\varepsilon}{4}$. On the one hand, conditioned on $\cap_{q=1}^{t} E_q$,

$$\sum_{t_i : arm^o(t_i) \in \mathcal{S}_r(t)} \widehat{\mu}_{arm^o(t_i)} - \widehat{\mu}_{arm^o(t_{i-1})} \le U - L.$$

On the other hand, since $\alpha \ge \frac{\varepsilon}{4}$, the update rule in Algorithm 1 indicates $\widehat{\mu}_{arm^o(t)} \ge \widehat{\mu}_{arm^o(t-1)} + \frac{\varepsilon}{4}$. Thus, we have

$$\sum_{t_i : arm^o(t_i) \in \mathcal{S}_r(t)} \widehat{\mu}_{arm^o(t_i)} - \widehat{\mu}_{arm^o(t_{i-1})} \ge \frac{(|\mathcal{S}_r(t)|-1)\varepsilon}{4}.$$

Hence, conditioned on event $\cap_{q=1}^{t} E_q$, we have

$$\frac{(|\mathcal{S}_r(t)|-1)\varepsilon}{4} \le U - L = \frac{r\varepsilon}{4} + \frac{\varepsilon}{8}. \quad (13)$$

Therefore if $\cap_{q=1}^{t} E_q$ holds, we get $|\mathcal{S}_r(t)| \le r+2$. Applying union bound, we have

$$\begin{aligned}\Pr(E) &= \prod_{t=1}^{T} \Pr\left(E_t \mid \cap_{q=1}^{t-1} E_q\right)\\ &= \prod_{r=1}^{\infty} \prod_{t: arm^o(t) \in \mathcal{S}_r(T)} \Pr\left(E_t \mid \cap_{q=1}^{t-1} E_q\right)\\ &= \prod_{r=1}^{8} \prod_{t: arm^o(t) \in \mathcal{S}_r(T)} \Pr\left(E_t \mid \cap_{q=1}^{t-1} E_q\right)\\ &\quad \cdot \prod_{r=9}^{\infty} \prod_{t: arm^o(t) \in \mathcal{S}_r(T)} \Pr\left(E_t \mid \cap_{q=1}^{t-1} E_q\right)\\ &\ge \prod_{r=1}^{8} \left(1 - \frac{4(r+2)\delta}{C^r}\right) \prod_{r=9}^{\infty} \left(1 - \frac{4(r+2)\delta}{C^{r-8}}\right)\\ &\ge 1 - \sum_{r=1}^{8} \frac{4(r+2)\delta}{C^r} - \sum_{r=9}^{\infty} \frac{4(r+2)\delta}{C^{r-8}}\\ &\ge 1 - \frac{3\delta}{4}. \quad (14)\end{aligned}$$

where first and second inequalities are due to Weierstrass product inequality and the last inequality is due to $C \ge 100$.

Step III. Based on the definition of E_t and $\mathcal{S}_r(t)$, we have

$$\begin{aligned}E &\subseteq \left\{ \bigcap_{t \ge 1} \left\{ \left\{ |\widehat{\mu}_{arm^o(t)} - \mu_{arm^o(t)}| \le \frac{(r_t-8)\varepsilon}{8} \right\} \bigcup \left\{ \mu_{arm^o(t)} \ge \mu_{arm^*} - \varepsilon \right\} \right\} \right\}\\ &\subseteq \left\{ \bigcap_{t \ge 1} \left\{ \left\{ \widehat{\mu}_{arm^o(t)} < \mu_{arm^*} - \frac{5\varepsilon}{8} \right\} \bigcup \left\{ \mu_{arm^o(t)} \ge \mu_{arm^*} - \varepsilon \right\} \right\} \right\}, \quad (15)\end{aligned}$$

where the second formula follows since if $|\widehat{\mu}_{arm^o(t)} - \mu_{arm^o(t)}| \le \frac{(r_t-8)\varepsilon}{8}$, we have

$$\begin{aligned}&\widehat{\mu}_{arm^o(t)} \le \mu_{arm^o(t)} + \frac{(r_t-8)\varepsilon}{8}\\ &\le \mu_{arm^*} - \frac{(r_t-1)\varepsilon}{8} + \frac{(r_t-8)\varepsilon}{8} < \mu_{arm^*} - \frac{5\varepsilon}{8}. \quad (16)\end{aligned}$$

This completes the proof. □

Based on Lemma 1, we are then ready to accomplish the correctness of Algorithm 1.

Proof of Correctness of Algorithm 1. Since $\alpha \leq \frac{\varepsilon}{2}$, from Algorithm 1, there exists an *arm*$^o(t)$ such that

$$\widehat{\mu}_{arm^*} - \frac{\varepsilon}{2} \leq \widehat{\mu}_{arm^o(t)}. \tag{17}$$

From Proposition 1, we know

$$\Pr\left(\widehat{\mu}_{arm^*} \geq \mu_{arm^*} - \frac{\varepsilon}{8}\right) \geq 1 - 2\exp\left(-\frac{s_1\varepsilon^2}{8}\right) \geq 1 - \frac{\delta}{4}. \tag{18}$$

Since $\alpha \geq 0$, from Algorithm 1, we have

$$\widehat{\mu}_{arm^o(t)} \leq \widehat{\mu}_{arm^o(T)}. \tag{19}$$

Combining (17) (18) (19) together, we have

$$\Pr\left(\widehat{\mu}_{arm^o(T)} \geq \mu_{arm^*} - \frac{5\varepsilon}{8}\right) \geq 1 - \frac{\delta}{4}. \tag{20}$$

From Lemma 1, we obtain

$$\Pr\left(\left\{\widehat{\mu}_{arm^o(T)} < \mu_{arm^*} - \frac{5\varepsilon}{8}\right\} \bigcup \left\{\mu_{arm^o(T)} \geq \mu_{arm^*} - \varepsilon\right\}\right) \geq 1 - \frac{3\delta}{4}. \tag{21}$$

Let

$$A = \left\{\widehat{\mu}_{arm^o(T)} < \mu_{arm^*} - \frac{5\varepsilon}{8}\right\},$$
$$\text{and} \quad B = \left\{\mu_{arm^o(T)} \geq \mu_{arm^*} - \varepsilon\right\}.$$

Then from (20), $\Pr(A) \leq \frac{\delta}{4}$. Therefore $\Pr(B) \geq \Pr(A \cup B) - \Pr(A) \geq 1 - \delta$, which completes the proof. □

Due to the space constraint, we provide a sketch of proof for the optimal sample complexity, and we refer interested readers to Appendix A for details.

Proof of Sample Complexity of Algorithm 1 (Sketch). The key idea is to bound the total expected number of pulls during the life cycle of each selected *arm*^{o}, i.e., the period from *arm*^{o} replacing its predecessor to *arm*^{o} being replaced by its successor. Given current *arm*^{o}, we divide the arriving arms during the life cycle of *arm*^{o} into two sets, i.e.,

$$S_1 := \{arm_i \colon \mu_{arm_i} \leq \widehat{\mu}_{arm^o} + \frac{3\varepsilon}{8}\},$$
$$\text{and} \quad S_2 := \{arm_i \colon \mu_{arm_i} > \widehat{\mu}_{arm^o} + \frac{3\varepsilon}{8}\}.$$

We show that (i) for each $arm_i \in S_1$, the expected number of pulls of arm_i is $O\big(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\big)$, and (ii) the total expected number of pulls of all arms in S_2 is $O(\tau_j|S_2|)$ where $|S_2|$ is $O(\text{polylog}(j))$ with high probability. Consequently, the total expected number of pulls for j arriving arms during the life cycle of *arm*^{o} is $O\big(\frac{j}{\varepsilon^2}\log\frac{1}{\delta}\big)$. In the following, we provide some intuitive analyses and the formal analysis is far more challenging and interested readers are referred to the appendix for more details.

Case I. Consider $arm_i \in S_1$ with $\mu_{arm_i} \leq \widehat{\mu}_{arm^o} + \frac{3\varepsilon}{8}$. By Hoeffding inequality, after $\Theta(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ pulls of arm_i, $\widehat{\mu}_{arm_i} \leq \mu_{arm_i} + \frac{\varepsilon}{8}$ holds with high probability. Thus, $\widehat{\mu}_{arm_i} \leq \widehat{\mu}_{arm^o} + \frac{\varepsilon}{2}$. If $\alpha = \frac{\varepsilon}{2}$, arm_i will be dropped. On the other hand, if $\alpha = \frac{\varepsilon}{4}$, arm_i will be pulled at most $2\tau_j$ times. As a result, the expected number of pulls of arm_i is $O\big(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\cdot(1-\frac{1}{\log j+1}) + \frac{2\tau_j}{\log j+1}\big) = O\big(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\big)$.

Case II. Consider $arm_i \in S_2$ with $\mu_{arm_i} > \widehat{\mu}_{arm^o} + \frac{3\varepsilon}{8}$. Again, by Hoeffding inequality, after $\Theta(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ pulls of arm_i, $\widehat{\mu}_{arm_i} \geq \mu_{arm_i} - \frac{\varepsilon}{8}$ holds with high probability. Thus, $\widehat{\mu}_{arm_i} > \widehat{\mu}_{arm^o} + \frac{\varepsilon}{4}$. If $\alpha = \frac{\varepsilon}{4}$, arm_i will replace *arm*^{o} and the life of *arm*^{o} ends. When $|S_2| = \Theta(\text{polylog}(j))$, $\alpha = \frac{\varepsilon}{4}$ will happen at least once with high probability.

Putting it together, we have the total expected number of pulls for all the j arms $O\big(j\cdot\frac{1}{\varepsilon^2}\log\frac{1}{\delta} + \text{polylog}(j)\cdot\tau_j\big) = O\big(\frac{j}{\varepsilon^2}\log\frac{1}{\delta}\big)$, since $j = \Omega(\text{polylog}(j))$. □

3. Single-Arm-Memory Algorithm for ε-KAI

In this section, we extend ε-BAI into its general version, i.e., ε-KAI that aims to find the ε-*top*-k arms using a *single-arm memory*. That is, we aim to find k arms such that each of which has the mean no smaller than $\mu_{arm^*(k)} - \varepsilon$, where *arm*$^*(k)$ is the k-th largest value in $\{\mu_{arm_1}, \ldots, \mu_{arm_n}\}$.

3.1. High Level Overview

First, we maintain the first k arms in a set[4], denoted as $\mathcal{A}$. Let *top*^{o} be the arm in $\mathcal{A}$ with the minimum estimated mean. When the following arm_i arrives in the stream, we compare $\widehat{\mu}_{arm_i}$ with $\widehat{\mu}_{top^o}$ to update *top*^{o}. Similarly, we perform the following two operations.

1. **Sampling.** We pull arm_i $\Theta(\frac{1}{\varepsilon^2}\log\frac{k}{\delta})$ times to get $\widehat{\mu}_{arm_i}$.
2. **Comparison.** We replace *top*^{o} with arm_i if $\widehat{\mu}_{arm_i} \geq \widehat{\mu}_{top^o} + \alpha$ holds, where $\alpha = \Theta(\varepsilon)$ follows the same setting in Algorithm 1.

In addition, our algorithm retains the following property.

Property 2. *For each arm in $\mathcal{A}$, $|\widehat{\mu}_{arm} - \mu_{arm}| \leq O(\varepsilon)$.*

Let *top*$^o(T)$ be the arm with the minimum estimated mean in the final returned set $\mathcal{A}$. Following the similar logic flow

[4]We store the IDs of these k arms and ignore the memory cost.

in section 2.1, the two operations and Property 2 guarantee that $|\mu_{top^o(T)} - \mu_{arm^*(k)}| \le O(\varepsilon)$ with high probability. The main idea is as follows. By applying union bound for the k arms $\{arm^*(1), \dots, arm^*(k)\}$, operation one ensures that for all $s \in [k]$, $|\widehat{\mu}_{arm^*(s)} - \mu_{arm^*(s)}| \le O(\varepsilon)$. As operation two indicates, $\widehat{\mu}_{top^o(T)}$ is the k-th largest estimated mean among those of arms in $\mathcal{A}$, which means there are at most $k-1$ values in $\{\widehat{\mu}_{arm^*(1)}, \cdots, \widehat{\mu}_{arm^*(k)}\}$ larger than $\widehat{\mu}_{top^o(T)}$ by $\Theta(\varepsilon)$. Therefore, based on operation one and operation two, we have $\widehat{\mu}_{top^o(T)} - \widehat{\mu}_{arm^*(k)} \le O(\varepsilon)$. Meanwhile, $|\widehat{\mu}_{top^o(T)} - \mu_{top^o(T)}| \le O(\varepsilon)$ holds according to Property 2. In consequence, we acquire $|\mu_{top^o(T)} - \mu_{arm^*(k)}| \le O(\varepsilon)$ and the k arms in $\mathcal{A}$ are *ε-top-k* arms with high probability.

How to implement the two operations and Property 2 within optimal number of pulls remains the main challenge in ε-KAI. However, techniques adopted in ε-BAI could basically tackle this issue. Hence we omit the details here.

3.2. The Algorithm

First we define the following two parameters of our algorithm.

$$\{s_\ell\}_{\ell=1}^{\infty}: \quad s_\ell = \frac{16}{\varepsilon^2} \cdot \log\left(\frac{C \cdot k}{\delta}\right) \cdot 2^\ell, \text{ and } s_0 = 0;$$

$$\{\tau_j\}_{j=1}^{\infty}: \quad \tau_j := \frac{32}{\varepsilon^2} \cdot \log\left(\frac{C \cdot k \cdot j^2}{\delta}\right),$$

where $C \ge 100$ is a universal constant.

Algorithm 2 presents the pseudo-code for ε-KAI. As noticed, Algorithm 2 is tailored based on Algorithm 1 to identify the *ε-top-k* arms. Specifically, to initialize $\mathcal{A}$, we pull the first k arrived arms s_1 times, and store them in $\mathcal{A}$. *top^o* denotes the arm with the minimum estimated mean in $\mathcal{A}$. Then, we compare each arriving *arm$_i$* with *top^o* through multiple rounds. This part is conducted similarly as the procedure of Algorithm 1 in section 2.2. Eventually, the indexes of *ε-top-k* arms in $\mathcal{A}$ are returned.

Our main result is formalized in the following theorem.

Theorem 2. *Given a stream of n arms, approximation parameter ε and confidence parameter δ in $(0,1)$, Algorithm 2 finds ε-top-k arms with probability at least $1-\delta$ using expected $O(\frac{n}{\varepsilon^2} \cdot \log(\frac{k}{\delta}))$ pulls and a single-arm memory.*

Theorem 2 summarizes the main results of Algorithm 2. Detailed proofs are in Appendix B.

Compared with Assadi & Wang (2020), our algorithm 2 is fundamentally different. Assadi & Wang (2020) require the assumption, e.g., $\Delta_{k+1} < \varepsilon$ and does not test each arm in FIFO order, which leads to $O(k)$ memory costs. In comparison, our algorithm does not make any explicit assumption and uses a single-arm memory.

Remark: This paper adopts the wildly used Explore-k metric (Kalyanakrishnan & Stone, 2010) which asks for k arms whose reward means are lower than that of the k-th best arm by at most ε. By using the similar technique (Cao et al., 2015; Jin et al., 2019), our algorithm can return the top-k arms such that the mean of i-th returned arm is at most ε lower than that of the i-th best arm.

Algorithm 2: Streaming ε-KAI

```
Input: $k$, $\varepsilon$, $\delta$, and a stream of $n$ arms.
Output: The indexes of $k$ arms.
1  initialize: $j \leftarrow 1, i \leftarrow 1, \mathcal{A} = \emptyset$;
2  for each arriving $arm_i$ ($i \le k$) do
3      pull $arm_i$ $s_1$ times to obtain $\widehat{\mu}_{arm_i}$;
4      insert $arm_i$ to $\mathcal{A}$;
5  $top^o \leftarrow \arg\min_{arm \in \mathcal{A}} \widehat{\mu}_{arm}$;
6  for each arriving $arm_i$ ($i > k$) do
7      set $\alpha$ as $\frac{\varepsilon}{4}$ with probability $\frac{1}{\log j+1}$, and as $\frac{\varepsilon}{2}$ with
         other probability $1 - \frac{1}{\log j+1}$;
8      $\ell \leftarrow 1$;
9      while true do
10         pull $arm_i$ for $s_\ell - s_{\ell-1}$ times;
11         if $\widehat{\mu}_{arm_i} \ge \widehat{\mu}_{top^o} + \alpha$ and $s_\ell > \tau_j$ then
12             $top^o \leftarrow arm_i$;
13             insert $arm_i$ into $\mathcal{A}$ and update $top^o$;
14             $j \leftarrow 1$;
15             break;
16         else if $\widehat{\mu}_{arm_i} < \widehat{\mu}_{top^o} + \alpha$ then
17             $j \leftarrow j + 1$;
18             break;
19         else
20             $\ell \leftarrow \ell + 1$;
21 return the indexes of the arms in $\mathcal{A}$;
```

4. Streaming BAI

Previous sections study the problems with *instance independent* sample complexity. However, for particular problem instances, the sample complexity could be highly optimized. In this section, we consider the streaming BAI problem and investigate the *instance dependent* sample complexity.

4.1. The Algorithm

Algorithm 3 presents the pseudo-code to address the streaming BAI problem. Notice that our algorithm borrows the existing Exponential-Gap-Eliminaion algorithm (Karnin et al., 2013; Chen et al., 2017c) as a framework. We substitute the selection component in this framework for Algorithm 1 (Line 4 in Algorithm 3), which is the major modification. Algorithm 3 runs in multiple rounds. Suboptimal arms in the stream are eliminated round by round until only one arm remains. In the r-th round, we maintain a set S_r of the total n arms in the stream, and ε-BAI algorithm is adopted as a subroutine to return an ε-best arm arm_r^o from S_r. We then compute its estimated mean I_r by pulling arm_r^o for $\frac{2}{\varepsilon_r^2}\log(\frac{1}{\delta_r})$ times. At the r-th round, we set the additional budget $B_r = \frac{6|S_r|}{\varepsilon_r^2}\log\left(\frac{40}{\delta_r}\right)$. If $B_r > 0$, for each arriving arm_i in $S_r \setminus \{arm_r^o\}$, we pull it in multiple iterations. At the ℓ-th iteration, we pull it $\frac{2^\ell}{\varepsilon_r^2}\log(\frac{40h^2}{\delta_r})$ times to get its estimated mean $\widehat{p}_i^\ell(r)$. Budget B_r is decreased by $\frac{2^\ell}{\varepsilon_r^2}\log(\frac{40h^2}{\delta_r})$ and s_i is increased by $\frac{2^\ell}{\varepsilon_r^2}\log(\frac{40}{\delta_r})$ accordingly. If $\widehat{p}_i^\ell(r) < I_r - \varepsilon_r$ holds ($\varepsilon_r = 2^{-r}/4$), we remove arm_i from S_r and consider the next arm in the stream till the number of pulls exceeds $\frac{2}{\varepsilon_r^2}\log\left(\frac{40h^2}{\delta_r}\right)$. If $B_r \le 0$, we pull it $\frac{2}{\varepsilon_r^2}\log(\frac{40h^2}{\delta_r})$ times to get its estimated mean $\widehat{p}_i(r)$. If $\widehat{p}_i(r) < I_r - \varepsilon_r$ holds, arm_i is removed from S_r. We continue to check following arriving arms till the end of the stream. This procedure is repeated until S_r contains only one arm whose index is returned as our final output. Notice that for each round, the stream is only visited $O(1)$ times.

Our main result for Algorithm 3 is formalized in the following theorem.

Theorem 3. *Given a stream of n arms and confidence parameter $\delta \in (0,1)$, Algorithm 3 identifies the optimal arm with probability at least $1-\delta$, in which case it takes expected $O\big(\sum_{i=2}^n \frac{1}{\Delta_i^2}\log\left(\frac{1}{\delta}\log\frac{1}{\Delta_i}\right)\big)$ arm pulls and $O(\log\Delta_2^{-1})$ passes using a single-arm memory.*

Here, we present some high level ideas why $O(\log\Delta_2^{-1})$ passes would suffice for the correctness of Algorithm 3. We consider ε_r in the following two cases.

Case 1. $\varepsilon_r \in (\Delta_2/3, 1]$. In each round, Algorithm 3 costs at most 2 passes. Therefore, the number of total passes is $O(\log\Delta_2^{-1})$.

Case 2. $\varepsilon_r \le \Delta_2/3$. According to Theorem 1, arm_r^o is an ε_r-best arm. From Hoeffding bound, we have $I_r \ge \mu_{arm_r^o} - \frac{\varepsilon_r}{2}$. Besides, the budget B_r ensures that with high probability $\widehat{\mu}_{arm_i} \le \mu_{arm_i} + \frac{\varepsilon_r}{2}$ (see the proofs for details). Therefore,

$$\begin{aligned} I_r - \widehat{\mu}_{arm_i} &\ge \mu_{arm_r^o} - \frac{\varepsilon_r}{2} - \widehat{\mu}_{arm_i} \ge \mu_{arm_r^o} - \varepsilon_r - \mu_{arm_i} \\ &\ge \mu_{arm^*} - \mu_{arm_i} - 2\varepsilon_r \ge \varepsilon_r, \end{aligned}$$

which indicates that with high probability, all suboptimal arms will be eliminated in the current round when $\varepsilon_r \le \Delta_2/3$. Therefore, it takes $O(1)$ passes in such case. The detail proofs are in Appendix C.

Note that previous BAI algorithms (Karnin et al., 2013; Jin et al., 2019) run in R rounds. In each round, the number of pulls of each arm is fixed. As a comparison, one can convert those algorithms into streaming algorithms in R passes. In this regard, the previous best known algorithm (Jin et al., 2019) will run in $\log^*(n)\cdot\log(1/\Delta_2)$ passes, which is inferior to our $\log(1/\Delta_2)$ passes. For sample complexity, our algorithm achieves the optimal instance-dependent sample complexity up to a $\log\log(1/\Delta_2)$ term, compared with the lower bound in Chen et al. (2017c).

Algorithm 3: ID-BAI

Input: Parameter δ and a stream of arms.
Output: The index of an arm.

```
1  Initialize $r \leftarrow 1$, $S_r = \{arm_1, arm_2, \cdots, arm_n\}$;
2  while $|S_r| > 1$ do
3      $\varepsilon_r \leftarrow 2^{-r}/4$, $\delta_r \leftarrow \delta/(40\cdot r^2)$, $h \leftarrow 1$;
4      $arm_r^o \leftarrow \varepsilon$-BAI$(\varepsilon_r, \delta_r, S_r)$;
5      pull $arm_r^o$ $\frac{2}{\varepsilon_r^2}\log(\frac{1}{\delta_r})$ times and let $I_r$ be the
         estimated mean;
6      $B_r \leftarrow \frac{6|S_r|}{\varepsilon_r^2}\log\left(\frac{40}{\delta_r}\right)$;
7      for each arriving $arm_i \in S_r \setminus \{arm_r^o\}$ do
8          if $B_r > 0$ then
9              $s_i \leftarrow 0$, $\ell \leftarrow 1$;
10             while $s_i \le \frac{2}{\varepsilon_r^2}\log(\frac{40h^2}{\delta_r})$ do
11                 pull $arm_i$ for $\frac{2^\ell}{\varepsilon_r^2}\log(\frac{40}{\delta_r})$ times, and let
                     $\widehat{p}_i^\ell(r)$ be the estimated mean;
12                 $B_r \leftarrow B_r - \frac{2^\ell}{\varepsilon_r^2}\log(\frac{40}{\delta_r})$;
13                 $s_i \leftarrow s_i + \frac{2^\ell}{\varepsilon_r^2}\log(\frac{40}{\delta_r})$;
14                 if $\widehat{p}_i^\ell(r) < I_r - \varepsilon_r$ then
15                     remove $arm_i$ from $S_r$;
16                     $h \leftarrow h + 1$;
17                     break;
18                 $\ell \leftarrow \ell + 1$;
19         else
20             pull $arm_i$ for $\frac{2}{\varepsilon_r^2}\log(\frac{40}{\delta_r})$ times, and let $\widehat{p}_i(r)$
                 be the estimated mean;
21             if $\widehat{p}_i(r) < I_r - \varepsilon_r$ then
22                 remove $arm_i$ from $S_r$;
23     $r \leftarrow r + 1$;
24 return the index of the arm in $S_r$;
```

5. Additional Related Work

We review the related work, excluding those (Assadi & Wang, 2020; Maiti et al., 2020; Falahatgar et al., 2020) discussed in Section 1.1. The problem of best arm identification is mostly considered in a non-streaming setting in the literature. Normally, two types of sample complexity are considered: *instance-dependent complexity* and *instance-*

independent complexity.

Instance-independent arm selection. Existing work (Even-Dar et al., 2002; Kalyanakrishnan & Stone, 2010; Cao et al., 2015; Jin et al., 2019) achieves the optimal worst sample complexity $\Omega(\frac{n}{\varepsilon^2}\log\frac{k}{\delta})$, matching the lower bound in (Kalyanakrishnan et al., 2012). In addition, the recent work (Hassidim et al., 2020) shows that the complexity of identifying an ε-best arm can be reduced to $\frac{n}{2\varepsilon^2}\log\frac{1}{\delta}$. However, all these algorithms require $\Theta(n)$ memory, which is inferior to ours. Recently, Assadi & Wang (2020) show that one can modify an r-round algorithm to an r-memory algorithm. In this way, the previous best known algorithm can be modified to a $O(\log^*(n))$ memory algorithm, which is also inferior to ours.

Instance-dependent arm selection. The instance-dependent sample complexity is closely tied to the bandit instance and is superior to the instance-independent complexity for 'easy' bandit instances. Existing work (Karnin et al., 2013; Jamieson et al., 2014; Chen et al., 2017c;b; Jin et al., 2019; Tao et al., 2019; Chen et al., 2017a) focuses on achieving the optimal instance-dependent sample complexity. The algorithm in (Karnin et al., 2013; Jamieson et al., 2014) achieves the sample complexity $O\big(\sum_{i=2}^{n}\frac{1}{\Delta_i^2}\log\big(\frac{1}{\delta}\log\frac{1}{\Delta_i}\big)\big)$. Furthermore, Jamieson et al. (2014) prove a lower bound such that for some instances the BAI problem needs at least $\Omega\big(\sum_{i=2}^{n}\frac{1}{\Delta_i^2}\log\big(\frac{1}{\delta}\log\frac{1}{\Delta_i}\big)\big)$ samples. Recently, Chen et al. (2017c) propose an instance-wise lower bound and almost optimal upper bound for the BAI problem. In another line of the work, Garivier & Kaufmann (2016) present a constant optimal algorithm under the assumption $\delta \to 0$.

Regret Minimization in Streaming Model. (Liau et al., 2018; Chaudhuri & Kalyanakrishnan, 2019) study the streaming bandits for regret minimization. Both algorithms consume $O(1)$ memory and visit the stream multiple passes. Since we achieve different objectives , their regret bound is not directly comparable to our sample complexity bound. It is interesting to see whether our algorithm can help to improve their results.

6. Conclusion

We study the streaming ε-*top*-k arms identification (ε-KAI) problem and the streaming BAI problem. For ε-KAI, we propose the first algorithm that applies to any k and achieves the optimal sample complexity using a single-arm memory without any explicit assumptions. For streaming BAI, we present a single-arm memory algorithm that achieves a near instance-dependent optimal sample complexity within $O(\log\Delta_2^{-1})$ passes.

Acknowledgement

We would like to thank Pan Xu for helpful discussions. This work is supported by the Ministry of Education, Singapore, under Tier-2 Grant R-252-000-A70-112, and by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-01-004[T]). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Assadi, S. and Wang, C. Exploration with limited memory: streaming algorithms for coin tossing, noisy comparisons, and multi-armed bandits. In *Proc. STOC*, pp. 1237–1250, 2020.

Bertsimas, D. and Mersereau, A. J. A learning approach for interactive marketing to a customer segment. *Operations Research*, 55(6):1120–1135, 2007.

Cao, W., Li, J., Tao, Y., and Li, Z. On top-k selection in multi-armed bandits and hidden bipartite graphs. In *Proc. NeurIPS*, pp. 1036–1044, 2015.

Chaudhuri, A. R. and Kalyanakrishnan, S. Regret minimisation in multi-armed bandits using bounded arm memory. In *Proc. AAAI*, pp. 10085–10092, 2019.

Chen, J., Chen, X., Zhang, Q., and Zhou, Y. Adaptive multiple-arm identification. In *International Conference on Machine Learning*, pp. 722–730. PMLR, 2017a.

Chen, L., Li, J., and Qiao, M. Nearly instance optimal sample complexity bounds for top-k arm selection. In *Proc. AISTATS*, pp. 101–110, 2017b.

Chen, L., Li, J., and Qiao, M. Towards instance optimal bounds for best arm identification. In *Proc. COLT*, pp. 535–592, 2017c.

Even-Dar, E., Mannor, S., and Mansour, Y. Pac bounds for multi-armed bandit and markov decision processes. In *Proc. COLT*, pp. 255–270, 2002.

Falahatgar, M., Orlitsky, A., and Pichapati, V. Optimal sequential maximization one interview is enough! In *Proc. ICML*, pp. 2975–2984, 2020.

Garivier, A. and Kaufmann, E. Optimal best arm identification with fixed confidence. In *Proc. COLT*, pp. 998–1027, 2016.

Hassidim, A., Kupfer, R., and Singer, Y. An optimal elimination algorithm for learning a best arm. In *Proc. NeurIPS*, 2020.

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

Jin, T., Shi, J., Xiao, X., and Chen, E. Efficient pure exploration in adaptive round model. In *Proc. NeurIPS*, pp. 6605–6614, 2019.

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

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

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

Liau, D., Song, Z., Price, E., and Yang, G. Stochastic multi-armed bandits in constant space. In *Proc. AISTATS*, pp. 386–394, 2018.

Maiti, A., Patil, V., and Khan, A. Streaming algorithms for stochastic multi-armed bandits. *arXiv preprint arXiv:2012.05142*, 2020.

Tao, C., Zhang, Q., and Zhou, Y. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *Proc. IEEE FOCS*, pp. 126–146, 2019.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Zhou, Y., Chen, X., and Li, J. Optimal pac multiple arm identification with applications to crowdsourcing. In *Proc. ICML*, pp. 217–225, 2014.