Project Report : Best Arm Identification in Bandits with Limited Precision Sampling

Aditya Mallick
EE21B005

Ameya Bagal
EE21B014

***Abstract*—This paper analyses best arm identification in a multi-armed bandit problem where the learner can only pull arms through "boxes" with specific probability distributions over arms, which is not known. The learner's goal is to identify the best arm with minimal expected stopping time, in a fixed confidence setting. It derives an asymptotic lower bound on stopping time and addresses the challenge of non-unique optimal allocations. A tracking-based algorithm is proposed to handle this non-uniqueness, proving it asymptotically optimal, and bounds are provided on stopping time for disjoint box-arm settings.**

I. Introduction

The paper addresses the problem of best arm identification in a multi-armed bandit setup where the learner can only sample arms indirectly through "boxes." Each box represents a probability distribution over arms, and selecting a box triggers a random pull of an arm, whose reward is observed. The learner's goal is to identify the best arm, defined as the arm with the highest mean reward, by minimizing the expected stopping time under a fixed confidence constraint.

This problem scenario can be interpreted in many ways, such as limited access to arms due to external constraints, noisy arm selection (e.g. the trembling hand model), or privacy considerations that obscure the learner's preferences.

Unlike classical best arm identification problems like SEA and LUCB, the learner doesn't have full control over the arm to pull at each time instant. Another challenge faced in this setting is that the optimal allocation is non-unique, hence the existing tracking based algorithms need to be improvised to accommodate it.

II. Related works

The work in [1] examines a setup similar to the paper in the context of community mode estimation; however, a key distinction is that their analysis and results apply to the *fixed-budget* regime, while the paper focuses on the *fixed-confidence* regime. Other similar models include the *trembling arm model* which is studied in [2] and the model with arm erasure mentioned in [3].

III. Problem Setup

We consider a K-armed bandit with arms $1, 2, \ldots, K$. Each arm is associated with a reward distribution ν_k and mean μ_k. The goal is to identify the optimal arm (with the highest mean reward) via sequential sampling. However, instead of accessing arms directly, the learner selects from M boxes, each with a probability distribution over the arms. Selecting box m results in arm k being pulled with probability $q_{m,k}$. It is important to note that $q := \{q_{m,k} : k \in \mathcal{A}_m, m \in [M]\}$ is unknown to the learner apriori.

The problem instance is specified completely by the tuple $C = (q, \nu)$ (where ν is the vector of arm distributions), with the best arm denoted by $a^*(C) = \arg\max_{k\in[K]} \mu_k$. For a given confidence threshold $\delta \in (0, 1)$, the goal is to find the best arm with minimal expected stopping time while keeping the probability of incorrect identification below δ.

Let B_t denote the box selected at time t. Upon selecting a box $B_t = m$, the arm $A_t = k$ is pulled with probability $q_{m,k}$, generating a reward X_t. The policy $\pi = \{\pi_t\}_{t=1}^{\infty}$ defines the learner's actions, where at each time t, it selects a box based on the history of observations, or it stops and declares an estimated best arm. We denote by τ_π the stopping time under policy π and by $\hat{a}$ the estimated best arm at stoppage. A policy is defined to be δ-probably correct (δ-PC) if $\mathbb{P}(\hat{a} \neq a^*(C)) \leq \delta$ for all problem instances C.

The paper proposes two algorithms for the problem setup. The first one is a track and stop based algorithm, where the class of arm distributions $\mathcal{G}$ is taken to be a family of distributions with known variance. The algorithm is shown to be δ-PC with the expected stopping time achieving optimal value as $\delta \downarrow 0$. The second algorithm is a successive elimination algorithm, for the special case where arms are partitioned across boxes. $\mathcal{G}$ is taken to be a family of 1 - sub Gaussian distributions.

IV. Algorithms

A. Track & Stop Based Algorithm

This algorithm is designed for a general setting, where each arm may lie in more than one box. The results are presented for the extreme setting where each arm is mapped to each box. The underlying problem instance is denoted by $q_0 = \{q_{m,k} : m \in M, k \in K\}$ and $\mu_0 = \{\mu_k^0 : k \in K\}$.

1) Asymptotic Lower Bound: First, an asymptotic lower bound is established for the stopping time of a $\delta - PC$ algorithm as δ approaches 0. For a given q_0 and μ_0 the lower bound is given as

$$\liminf_{\delta\downarrow 0} \inf_{\pi\in\Pi(\delta)} \frac{\mathbb{E}[\tau_\pi]}{\log(1/\delta)} \geq \frac{1}{T^*(\boldsymbol{q}_0, \boldsymbol{\mu}_0)}, \tag{1}$$

Algorithm 1 Boxed-Bandit Modified Track-and-Stop

Input: $\delta \in (0,1)$, $\rho > 0$, $K \in \mathbb{N}$, and $M \in \mathbb{N}$
Output: $\hat{a} \in [K]$ – best arm
Initialisation: $t = 0$, $\hat{\mu}_a(t) = 0 \, \forall a \in [K]$, $Z(0) = 0$.
1: Compute $\hat{a}(t) = \arg\max_a \hat{\mu}_a(t)$.
2: **if** $Z(t) \geq \zeta(t, \delta, \rho)$ and $\min_k N_k(t) > 0$ **then**
3: Stop box selections.
4: **return** $\hat{a} = \arg\max_{a \in [K]} \hat{\mu}_a(t)$
5: **else**
6: Select box B_{t+1} as per *modified D-tracking rule* (**??**).
7: Update $\hat{\boldsymbol{q}}(t)$, $\hat{\boldsymbol{\mu}}(t)$, and $Z(t)$. Go to step 1.
8: **end if**

where $T^*(\boldsymbol{q}_0, \boldsymbol{\mu}_0)$ in (1) is given by

$$T^*(\boldsymbol{q}_0, \boldsymbol{\mu}_0) = \sup_{w \in \Sigma_M} \inf_{\boldsymbol{\lambda} \in \text{ALT}(\boldsymbol{\mu}_0)} \sum_{m=1}^{M} \sum_{k=1}^{K} w_m \, q_{m,k}^0 \, \frac{(\mu_k^0 - \lambda_k)^2}{2}, \tag{2}$$

Here w is the allocation strategy for the boxes, where w_m denotes the probability with which the learner selects box $m \in M$.

2) Non-unique Optimal Allocations: One of the main challenges faced in the non-partitioned setting is the non-uniqueness of the optimal allocation w. We take $\mathcal{W}^*(q, \mu)$ to be the set of all optimal allocations for a certain instance (q, μ). The paper further establishes the following,

- The mapping $(\boldsymbol{q}, \boldsymbol{\mu}) \mapsto \mathcal{W}^\star(\boldsymbol{q}, \boldsymbol{\mu})$ is upper-hemicontinuous and compact-valued.
- The set $\mathcal{W}^*(q, \mu)$ is convex for all instances (q, μ).

A key feature of best arm identification problems is tracking, i.e. the empirical frequencies of arm pulls (in this case box allocations) should approach the optimal allocation asymptotically. However, in the provided scenario, when multiple allocations are equally optimal, the selection frequencies may oscillate among these options rather than converging to a single choice. This presents challenges in demonstrating effective tracking. The paper presents a technique that handles the non-unique optimal allocations and develops a framework to demonstrate tracking like behaviour.

3) The Algorithm: The algorithm takes two inputs, $\delta \in (0,1)$ and $\rho > 0$. It also maintains an estimate of the best arm $\hat{a}(t) \in \arg\max_a \hat{\mu}_a(t)$ at each time t.

The boxes are chosen in a round robin fashion initially, such that each box is sampled sufficiently. After this, the boxes which are under-sampled are prioritized, till $N(t, m) = \Omega(\sqrt{t})$ $\forall m \in M$, where $N(t, m)$ denotes the number of times box m has been sampled till time t.

After the the initial exploration, the boxes are sampled based on the allocation $\{w(s) : 1 \leq s \leq t\}$, where $w(s)$ is an arbitrary element of $\mathcal{W}^*(\hat{q}(s-1), \hat{\mu}(s-1))$. This is called the *modified D-tracking rule*.

The stopping condition for the algorithm is given by,

$$\min_{b \neq \hat{a}(t)} Z_{\hat{a}(t),b}(t) \geq \zeta(t, \delta, \rho) \tag{3}$$

where, $\zeta(t, \delta, \rho) := \log(C \, t^{1+\rho}/\delta)$, where C is a constant that satisfies $\sum_{t=1}^{\infty} \frac{e^{K+1}}{K^K} \frac{(\log^2(C \, t^{1+\rho}) \, \log t)^K}{t^{1+\rho}} \leq C$. Further, $Z_{a,b}(t)$ denotes the *generalised likelihood ratio test statistic* between arms $a, b \in [K]$ up to time t. The paper provides an explicit expression for $Z_{a,b}(t)$ as,

$$Z_{a,b}^{\pi}(t) = N_a(t) \frac{(\hat{\mu}_a(t) - \hat{\mu}_{a,b}(t))^2}{2} + N_b(t) \frac{(\hat{\mu}_b(t) - \hat{\mu}_{a,b}(t))^2}{2}, \tag{4}$$

For any $a, b \in [K]$ for a policy π such that all arms have been sampled at least once and $\hat{\mu}_a(t) \geq \hat{\mu}_b(t)$. Further, $\hat{\mu}_{a,b}(t)$ is defined as,

$$\hat{\mu}_{a,b}(t) = \frac{N_a(t)}{N_a(t) + N_b(t)} \hat{\mu}_a(t) + \frac{N_b(t)}{N_a(t) + N_b(t)} \hat{\mu}_b(t). \tag{5}$$

B. BBSEA Algorithm

A simplified version of the original problem setting, we consider the case where each arm is only present in a single box. In other words, the arms are partitioned into the boxes. To deal with this scenario, the authors propose a modified version of successive elimination, termed the Boxed-Bandit Successive Elimination Algorithm. The principle is very similar to that of SEA, where each arm is sampled a fixed number of times in every round. The only difference, however, is the fact that we cannot ensure that we pick each arm a fixed number of times every round, since we do not know which arm we will pick from a given box.

Like SEA, BBSEA maintains empirical reward estimates to create a confidence interval, which is used to decide the fate of an arm. BBSEA works on fixed confidence, i.e, for a confidence $\delta \in (0,1)$, we have $\alpha_\delta(x) = \sqrt{\frac{2 \log(8Kx^2/\delta)}{x}}$. Then, the $\text{UCB}_{m,k}(n)$ and $\text{LCB}_{m,k}(n)$ at time n are written as:

$$\text{UCB}_{m,k}(n) = \hat{\mu}_{m,k}(n) + \alpha_\delta(t_{m,k}(n)) \tag{6}$$

$$\text{LCB}_{m,k}(n) = \hat{\mu}_{m,k}(n) - \alpha_\delta(t_{m,k}(n)) \tag{7}$$

for arm k belonging to box m.

All arms having an upper confidence bound (UCB) that is smaller than the lower confidence bound (LCB) of any other arm are eliminated at the end of a round. In this way, the algorithm successively eliminates sub-optimal arms, outputting the best arm with a reasonable guarantee.

Alluding to an earlier made point, the boxed-bandit has the added uncertainty of not knowing apriori the arm being sampled. Intuitively, this may become a problem in the case of skewed boxed-bandit distributions, where one arm is picked more frequently than the rest inside a box. Since the algorithm needs to ensure that all arms are picked a minimum fixed number of times within a round, this condition may have an adverse effect on the number of pulls required.

V. Main Results

A. Performance of the BBMTS algorithm

The algorithm satisfies the following performance criteria

Algorithm 2 Boxed-Bandit Successive Elimination Algorithm

Input: K, M, $\delta > 0$, $\mathcal{A}_m$ for $m \in [M]$
Output: $\hat{a}_{\text{BBSEA}} \in [K]$ (best arm).
Initialization: $B = [M]$, $n = 0$, $t = 0$,
$S_m = \mathcal{A}_m \ \forall m$, $S = \bigcup_m S_m$, $\hat{\mu}_{m,k}(0) = 0 \ \forall m, k$.

while $|S| > 1$ **do**
  $n \leftarrow n + 1$
  For each $m \in B$, select box m until every active arm $A_{m,k}$ in box m is pulled at least n times. For every box selection, increment t by 1.
  Update $t_{m,k}(n)$, $\hat{\mu}_{m,k}(n)$, $\text{UCB}_{m,k}(n)$ and $\text{LCB}_{m,k}(n)$ for all the active arms.
  if $\exists A_{m',k'} \in S$ such that $\text{UCB}_{m,k}(n) < \text{LCB}_{m',k'}(n)$ **then**
    $S_m \leftarrow S_m \backslash A_{m,k}, \quad S \leftarrow \bigcup_{m \in [M]} S_m,$
    $B \leftarrow \{m : S_m \neq \emptyset\}$.
  end if
  if $|S| = 1$ **then**
    $\hat{a}_{\text{BBSEA}} \leftarrow a \in S, \quad S \leftarrow \emptyset, \quad B \leftarrow \emptyset$.
  end if
end while
return $\hat{a}_{\text{BBSEA}}$.

1) $\pi_{\text{BBMTS}}(\delta, \rho) \in \Pi(\delta)$ for each $\delta \in (0, 1)$ and $\rho > 0$.
2) For each $\rho > 0$, the stopping time of $\pi_{\text{BBMTS}}(\delta, \rho)$ satisfies

$$\limsup_{\delta \downarrow 0} \frac{\tau_{\pi_{\text{BBMTS}}(\delta,\rho)}}{\log(1/\delta)} \leq \frac{1+\rho}{T^*(\boldsymbol{q}_0, \boldsymbol{\mu}_0)} \quad \text{a.s..} \tag{8}$$

3) For each $\rho > 0$, the quantity $\mathbb{E}[\tau_{\pi_{\text{BBMTS}}(\delta,\rho)}]$ satisfies

$$\limsup_{\delta \downarrow 0} \frac{\mathbb{E}[\tau_{\pi_{\text{BBMTS}}(\delta,\rho)}]}{\log(1/\delta)} \leq \frac{1+\rho}{T^*(\boldsymbol{q}_0, \boldsymbol{\mu}_0)}. \tag{9}$$

ρ therefore serves as a tuneable parameter that may be set to make the upper bound in (9) as close to (1) as desired. We can see that the proposed upper bound and lower bound match as $\rho \downarrow 0$. Equations (9) and (1) together imply that $1/T^*(\boldsymbol{q}_0, \boldsymbol{\mu}_0)$ is the optimal asymptotic growth rate of the expected stopping time.

B. Performance of the BBSEA algorithm

The stopping time of the BBSEA algorithm is equivalent to the number of arm pulls. For a fixed confidence $\delta \in (0, 1)$, and a K-armed, M-boxed partitioned bandit case having boxwise probability distribution q and 1-subgaussian rewards, the paper provides the following results regarding the stopping time:

1) Define $\alpha_{m,k} = 1 + \frac{102}{\Delta^2_{m,k}} \log\left(\frac{64\sqrt{\frac{8K}{\delta}}}{\Delta^2_{m,k}}\right)$ for arm k, in box m. Moreover, let $\beta_{m,k} = \frac{1}{q^0_{m,k}}\left[\alpha_{m,k} + 2\log\frac{2K}{\delta} + 2\sqrt{\log\frac{2K}{\delta}\left(\log\frac{2K}{\delta} + \alpha_{m,k}\right)}\right]$, and let β_m be the largest $\beta_{m,k}$ for box m.
2) BBSEA outputs the best arm correctly.
3) The stopping time of BBSEA is $\leq \sum_{m=1}^{M} \beta_m$.

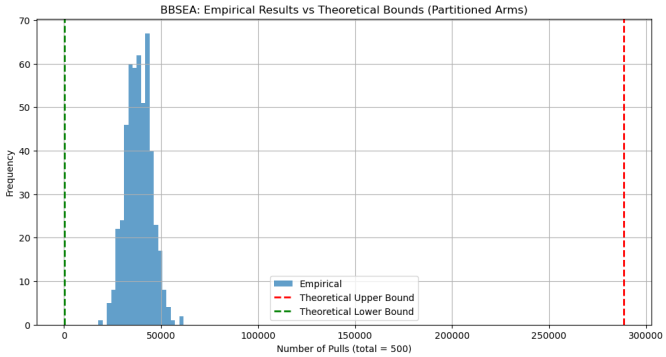

Fig. 1. Gaussian Rewards - 500 runs

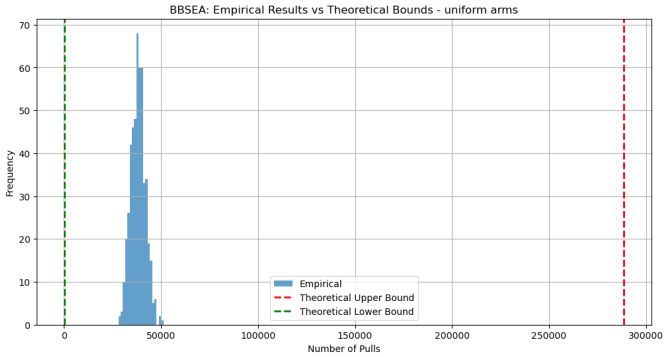

Fig. 2. Uniform Rewards - 500 runs

4) Furthermore, for any policy π, a lower bound on the stopping time is given by:

$$\mathbb{E}[\tau_\pi] \geq \log\left(\frac{1}{2.4\,\delta}\right) \cdot \sum_{m=1}^{M} \max_{k \in \mathcal{A}_m} \frac{1}{q^0_{m,k}\,\Delta^2_{m,k}}. \tag{10}$$

VI. Simulations

To evaluate the performance of BBSEA, we perform a number of experiments by varying key parameters associated with the algorithm and visualizing the effects.

A. Upper Bound Analysis

Assuming a general setup, we use a histogram to plot the results of BBSEA for a boxed-bandit setup with Gaussian rewards (variance = 1). We use the bound presented earlier to compute the upper bound for our problem instance. Additionally, we also plot the lower bound for any general BAI policy, given earlier. We observe that the algorithm performs quite well, with the average number of pulls being well below the upper bound. We also explore a few other 1-subgaussian instances, namely uniform and bernoulli bandits, and observe a similar trend. We maintain the same mean for all arms, as well as other parameters, to ensure uniformity.

B. Parameter Variation

We would like to know which parameters have a significant effect on the average stopping time of BBSEA.

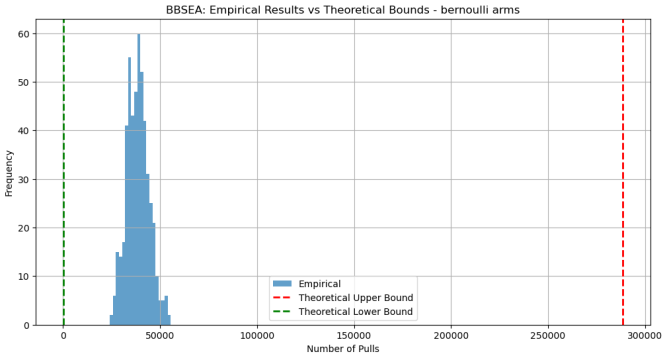

Fig. 3. Bernoulli Rewards - 500 runs

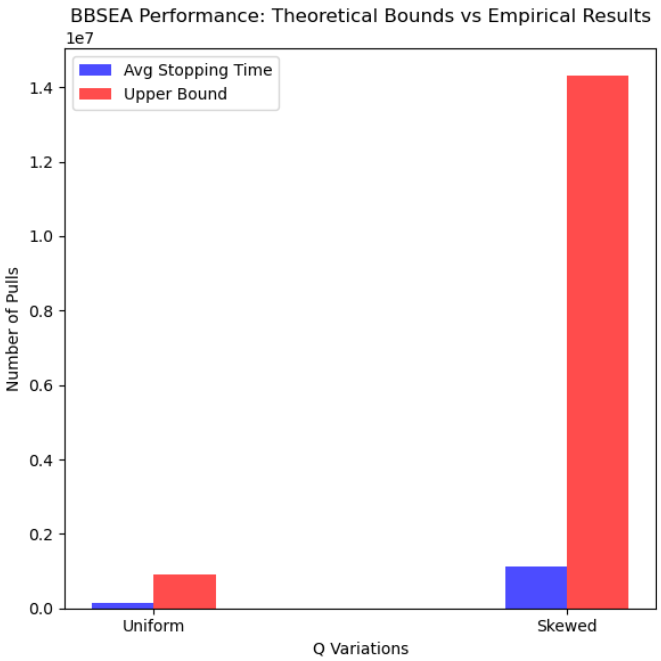

Fig. 4. Variation of Boxwise Arm Distribution

- As expected, a highly skewed distribution results in an increased stopping time, as well as an increased upper bound, since it is instance-dependent. This can also be viewed as low entropy within a box.
- Varying the gap between the best and second-best arm is also a very important parameter that decides the number of pulls required. We observe that a low gaps lead to a larger stopping time, as expected from any SEA.
- Finally, we vary the number of arms in our problem setting, keeping the number of boxes constant. As expected, increasing the number of arms leads to more arms needing to be eliminated, thus increasing the stopping time.

VII. Conclusion and Future Directions

The main challenge addressed by the paper is the non-uniqueness of optimal allocations in the problem setting. The

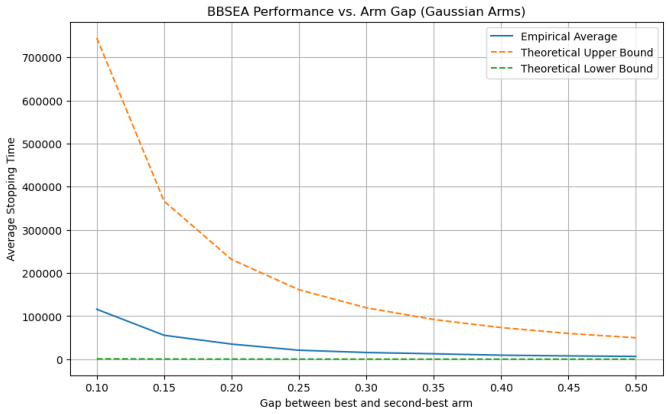

Fig. 5. Gap Variation

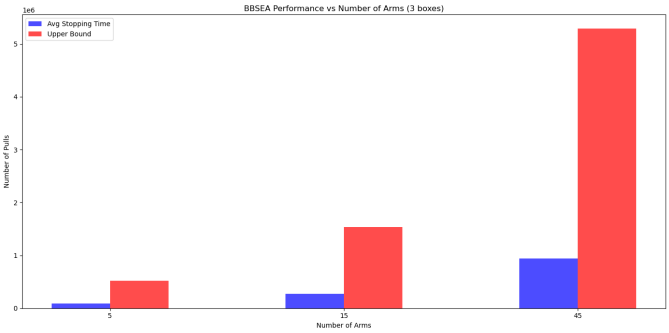

Fig. 6. Varying the Number of Arms

paper showed that asymptotic optimality can be achieved by tracking the empirical average of arbitrarily chosen optimal allocations. The paper also proposes future work to develop algorithms that admit non asymptotic upper bounds.

Additionally, the paper presents an algorithm to deal with the special case of partitioned arms, providing an upper bound on the number of pulls required to identify the best arm with fixed confidence. The results of the simulations show that BBSEA indeed performs unto and well beyond the limits imposed by the upper bound. However, it fails to handle skewed boxed-bandit settings, and also faces the drawbacks of any SEA-based algorithm. We hope to eliminate these drawbacks by drawing inspiration from better techniques like SHA [4], and improvising them for this problem setting.

References

[1] S. A. Jain, S. Goenka, D. Bapna, N. Karamchandani, and J. Nair, "Sequential community mode estimation," Performance Evaluation, vol. 152, p. 102247, 2021.

[2] P. N. Karthik and R. Sundaresan, "Detecting an odd restless Markov arm with a trembling hand," IEEE Transactions on Information Theory, vol. 67, no. 8, pp. 5230–5258, 2021.

[3] K. S. Reddy, P. N. Karthik and V. Y. F. Tan, "Best Arm Identification with Arm Erasures," 2024 IEEE International Symposium on Information Theory (ISIT), Athens, Greece, 2024

[4] Almost Optimal Exploration in Multi-Armed Bandits Zohar Karnin, Tomer Koren, Oren Somekh Proceedings of the 30th International Conference on Machine Learning, PMLR 28(3):1238-1246, 2013.