Near-Optimal Arm Identification in Continuum-Armed Bandits

Animesh Bohara  
Bachelor of Technology  
Computer Science & Engineering  
160050040

Ritesh Goenka  
Bachelor of Technology  
Computer Science & Engineering  
160050047

Ankit Goyal  
Bachelor of Technology  
Computer Science & Engineering  
160050046

Eashan Gupta  
Bachelor of Technology  
Computer Science & Engineering  
160050045

Abstract

We consider the problem of finding a near optimal arm in continuum armed bandits. The conditions on a metric space under which this problem is solvable are discussed. We present a novel finite horizon probability analysis of the $\epsilon$-optimal arm identification problem assuming that the mean reward function is Lipschitz continuous. The empirical performance of several algorithms is analyzed and compared based on expected distance between true mean and mean of the recommended arm.

1 Introduction

The multi-armed bandits problem (with finite set of arms) has been studied extensively in literature. This problem has been explored in various settings such as regret minimization, optimal arm identification, etc. Such problems often arise in online decision problems where the agent must choose among a set of strategies to minimize the total cost. We consider the case introduced in [1] where the arm space is continuous and therefore infinite. Some applications of this problem are online auction mechanism design [2], in which the strategy space is an interval of feasible prices, and online oblivious routing [3], in which the strategy space is a flow polytope.

While most previous works consider the continuum-armed bandit problem in the regret minimization setting [1] [4] [5], we consider the problem in the optimal arm identification setting. This problem of finding an arm with mean near the optimal mean has been termed the pure exploration problem [6]. We first identify spaces which are explorable in the sense that there exist algorithms which solve the problem of finding a near-optimal arm. Theorem 3.1 gives a class of such spaces. We then derive a novel general result on bounding the finite horizon probability of finding a near optimal arm in Theorem 4.1. This analysis can be applied to any algorithm since it only requires the sampling history for a finite horizon. Finally, we consider a few algorithms and evaluate them empirically.

There are a few approaches which have been used in the literature to solve the problem at hand. One such algorithm for instances with concave mean reward functions is given in [7]. It constructs a sequence of arms which converges to an optimal arm in the limit. A zooming algorithm is proposed in [5] to solve the regret minimization problem. The strategy in this case is to divide the interval $[0, 1]$ into a number of intervals of different sizes with the objective that more points are sampled in the region with higher mean values. We modify this algorithm in a suitable way to solve the near-optimal arm identification problem. Finally, we compare the empirical performance of the two algorithms mentioned above along with the round robin algorithm.

Foundations of Intelligent and Learning Agents course project.
2 Definitions

Definition 2.1. A metric space is an ordered pair \((\mathcal{X}, d)\) where \(\mathcal{X}\) is a set and \(d\) is a metric on \(\mathcal{X}\) i.e. a function
\[
d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}
\]
such that for any \(x, y, z \in \mathcal{X}\), the following conditions hold
1. \(d(x, y) = 0 \iff x = y\)
2. \(d(x, y) = d(y, x)\)
3. \(d(x, y) + d(y, z) \leq d(x, z)\)

Definition 2.2. Let \((\mathcal{X}, d)\) be a metric space. A subset \(\mathcal{Y} \subseteq \mathcal{X}\) is said to be dense in \(\mathcal{X}\) if for any \(x \in \mathcal{X}\) and \(\epsilon > 0\), there exists \(y \in \mathcal{Y}\) such that \(d(x, y) < \epsilon\).

Definition 2.3. A metric space \((\mathcal{X}, d)\) is said to be separable if there exists a countable set \(\mathcal{Y} \subseteq \mathcal{X}\) which is dense in \(\mathcal{X}\).

Definition 2.4. Let \((\mathcal{X}, d)\) be a metric space. A function \(f : \mathcal{X} \to \mathbb{R}\) is said to be Lipschitz continuous if there exists a constant \(L > 0\) such that \(|f(x) - f(y)| \leq Ld(x, y)\) \(\forall x, y \in \mathcal{X}\).

Let us denote the set of arms by \(\mathcal{X}\) and let \(d\) be a metric on the set \(\mathcal{X}\). On pulling any arm \(x \in \mathcal{X}\), the agent obtains a reward from the reward distribution \(f_x\) corresponding to arm \(x\). Let \(\mu_x\) denote the expectation of the reward obtained on pulling arm \(x\). Let us define the mean function \(\mu : \mathcal{X} \to \mathbb{R}\) by \(\mu(x) = \mu_x\). We make the following assumptions on the reward distributions.

1. The reward distributions corresponding to each \(x \in \mathcal{X}\) have a common compact support. Since every compact set in \(\mathbb{R}\) is contained in a closed and bounded interval, it may be assumed without loss of generality that the common compact support is a closed interval in \(\mathbb{R}\). We assume this common support to be the unit interval \([0, 1]\).
2. The mean function \(\mu\) is continuous w.r.t. the topology induced by the metric \(d\) on space \(\mathcal{X}\).

A near-optimal arm identification algorithm \(A\) takes as input the horizon \(T\) and outputs a recommendation (arm). Let the arm returned by the algorithm be denoted by \(x_T^A\). Let \(\mu^* = \max_{x \in \mathcal{X}} \mu(x)\) be the optimal value of mean. Let \(\mathcal{X}^*\) be the set of optimal arms i.e. \(\mathcal{X}^* = \{x \in \mathcal{X} \mid \mu(x) = \mu^*\}\).

Definition 2.5. Simple regret of an algorithm is defined as the difference between the optimal mean and the mean of the recommended arm.

Let us denote the simple regret of algorithm \(A\) by \(r_T^A\). Then, we have \(r_T^A = \mu^* - \mu(x_T^A)\). The objective of the near-optimal arm identification problem is to minimize the simple regret.

Definition 2.6. An arm \(x \in \mathcal{X}\) is said to be \(\epsilon\)-optimal if \(\mu^* - \mu(x) \leq \epsilon\).

Definition 2.7. An algorithm \(A\) is said to be limiting-optimal if it satisfies the following property for all families of reward distributions \(\mathcal{F} = \{f_x : x \in \mathcal{X}\}\) satisfying assumptions 1 and 2.
\[
\lim_{T \to \infty} \mathbb{P}\{\mu^* - \mu(x_T^A) \leq \epsilon\} = 1, \quad \forall \epsilon > 0
\]

3 Initial Results

In this section, we characterize the metric spaces for which a limiting-optimal algorithm exists since these are the spaces where an optimal arm can be identified in countable number of time steps. Theorem 3.1 gives a sufficient condition for a metric space to satisfy the above property. This theorem is a reformulation of one side of Theorem 4 in [6], in which the proof of the converse is also given. We present a slightly more elementary proof as compared to the one given in [6].

Theorem 3.1. Let \((\mathcal{X}, d)\) be a metric space. Then there exists a limiting-optimal algorithm if \((\mathcal{X}, d)\) is separable.

Proof. Let us assume that the metric space \((\mathcal{X}, d)\) is separable. Then there exists a countable dense subset \(\mathcal{Y} \subseteq \mathcal{X}\). Since \(\mathcal{Y}\) is countable, there exists a bijection from \(\mathcal{Y}\) to \(\mathbb{N}\). Henceforth, we may
assume that \( \mathcal{Y} = \{y_n : n \in \mathbb{N}\} \). We now describe an algorithm and show that it is limiting-optimal. Consider an algorithm \( A_0 \) which samples the arms in the following manner.

\[
\begin{array}{ll}
R_1 & y_1 \\
R_2 & y_1, y_2 \\
R_3 & y_1, y_2, y_3 \\
\ldots
\end{array}
\]

After \( T > 0 \) time steps, the algorithm outputs the arm with maximum empirical mean among the set of arms which are pulled at least \( h(T) = (\sqrt{2T} - 2)/2 \) times. We shall now prove that \( A_0 \) is limiting-optimal. Let \( x^* \in \mathcal{X} \) be an optimal arm and \( \epsilon > 0 \) be arbitrary. Since the mean function \( \mu \) is continuous, there exists \( \delta > 0 \) such that

\[
|x - x^*| < \delta \implies |\mu(x) - \mu(x^*)| < \frac{\epsilon}{3}.
\]

Now since \( \mathcal{Y} \) is dense in \( \mathcal{X} \), there exists \( n_0 \in \mathbb{N} \) such that

\[
|y_{n_0} - x^*| < \delta.
\]

For \( T > 0 \), let the number of rounds completed after \( T \) time steps be \( l(T) \), then we have

\[
\frac{l(T)(l(T) + 1)}{2} \leq T < \frac{(l(T) + 1)(l(T) + 2)}{2},
\]

which implies

\[
2h(T) = \sqrt{2T} - 2 < l(T) < \sqrt{2T}.
\]

In this first \( k \) rounds, each of the arms in \( \{y_i : i \in \text{ceil}(k/2)\} \) is pulled at least \( k/2 \) times. Let \( T \geq 2(n_0 + 1)^2 \), then \( l(T) > 2h(T) = 2n_0 \). Therefore, the arm \( y_{n_0} \) is pulled at least \( l(T)/2 \) (> \( h(T) \)) times. We obtain the following probability bound using Hoeffding’s inequality [8].

\[
P\left\{ \mu(y_{n_0}) - \hat{\mu}(y_{n_0}) \leq \frac{\epsilon}{3} \right\} \geq 1 - \exp\left( -\frac{2\epsilon^2 N(y_{n_0})}{9} \right) \geq 1 - \exp\left( -\frac{2\epsilon^2 h(T)}{9} \right)
\]

For simplicity, let us denote the arm recommended by the algorithm by \( x_T \). Then we have the following probability bound using Hoeffding’s inequality [8].

\[
P\left\{ \hat{\mu}(x_T) - \mu(x_T) \leq \frac{\epsilon}{3} \right\} \geq 1 - \exp\left( -\frac{2\epsilon^2 N(x_T)}{9} \right) \geq 1 - \exp\left( -\frac{2\epsilon^2 h(T)}{9} \right)
\]

Since the above events are independent, we have

\[
P\left\{ \mu(y_{n_0}) - \hat{\mu}(y_{n_0}) \leq \frac{\epsilon}{3} \text{ and } \hat{\mu}(x_T) - \mu(x_T) \leq \frac{\epsilon}{3} \right\} \geq \left[ 1 - \exp\left( -\frac{2\epsilon^2 h(T)}{9} \right) \right]^2. \tag{1}
\]

Since \( x_T \) is the arm that is recommended by the algorithm, we have

\[
\hat{\mu}(x_T) \geq \hat{\mu}(y_{n_0}).
\]

Then, we have

\[
P\left\{ \mu(y_{n_0}) - \hat{\mu}(y_{n_0}) \leq \frac{\epsilon}{3} \text{ and } \hat{\mu}(y_{n_0}) - \mu(x_T) \leq \frac{\epsilon}{3} \right\} \geq
\]

\[
P\left\{ \mu(y_{n_0}) - \hat{\mu}(y_{n_0}) \leq \frac{\epsilon}{3} \text{ and } \hat{\mu}(x_T) - \mu(x_T) \leq \frac{\epsilon}{3} \right\}, \tag{2}
\]

since

\[
\hat{\mu}(x_T) \geq \hat{\mu}(y_{n_0}) \text{ and } \hat{\mu}(x_T) - \mu(x_T) \leq \frac{\epsilon}{3} \implies \hat{\mu}(y_{n_0}) - \mu(x_T) \leq \frac{\epsilon}{3}.
\]

Therefore, we have

\[
P\left\{ \mu(y_{n_0}) - \mu(x_T) \leq \frac{2\epsilon}{3} \right\} \geq P\left\{ \mu(y_{n_0}) - \hat{\mu}(y_{n_0}) \leq \frac{\epsilon}{3} \text{ and } \hat{\mu}(x_T) - \mu(x_T) \leq \frac{\epsilon}{3} \right\}, \tag{3}
\]

since

\[
\mu(y_{n_0}) - \hat{\mu}(y_{n_0}) \leq \frac{\epsilon}{3} \text{ and } \hat{\mu}(y_{n_0}) - \mu(x_T) \leq \frac{\epsilon}{3} \implies \mu(y_{n_0}) - \mu(x_T) \leq \frac{2\epsilon}{3}.
\]
Further, we obtain
\[ P\{\mu(x^*) - \mu(x_T) < \epsilon\} \geq P\left\{ \mu(y_{n_0}) - \mu(x_T) \leq \frac{2\epsilon}{3} \right\}, \quad (4) \]
since
\[ \mu(y_{n_0}) - \mu(x_T) \leq \frac{2\epsilon}{3} \] and \[ |\mu(x^*) - \mu(y_{n_0})| < \frac{\epsilon}{3} \implies \mu(x^*) - \mu(x_T) < \epsilon. \]
Finally, combining (1), (2), (3), and (4), we obtain
\[ 1 \geq P\{\mu(x^*) - \mu(x_T) \leq \epsilon\} \geq \left[ 1 - \exp\left( -\frac{2\epsilon^2 h(T)}{9} \right) \right]^2. \]
Now taking limit as \( T \to \infty \) in the above inequality and using sandwich theorem, we obtain
\[ \lim_{T \to \infty} P\{\mu(x^*) - \mu(x_T) \leq \epsilon\} = 1. \]
Hence, we have proved that the algorithm \( A_0 \) is limiting-optimal. Thus, if a metric space is separable then there exists a limiting-optimal algorithm.

It is noteworthy to see that in the above proof, we may choose any function \( h(T) \) which satisfies the following properties.
\[ \lim_{T \to \infty} h(T) = \infty, \quad \forall n_0 \exists T_0 \in \mathbb{N} \text{ such that } y_{n_0} \text{ is pulled atleast } h(T) \text{ times } \forall T \geq T_0 \]
Let us consider the metric space \([0, 1], \rho\), where \( \rho \) is the the euclidean metric \( \rho(x, y) = |x - y| \). Clearly, this metric space is compact and hence separable. Henceforth, let us work with the metric space \( X = [0, 1] \) equipped with the euclidean metric \( \rho(x, y) = |x - y| \).

4 General Analysis

In this section, we obtain a lower bound on the probability of the arm recommended by an algorithm after \( T \) time steps being \( \epsilon \)-optimal. Note that we assume that the algorithm always returns an arm which has been sampled atleast once. This is a "good" assumption since the agent doesn’t have any information about other arms. Infact, a more generalized assumption is that the algorithm recommends the arm with the highest empirical mean among the set of arms which have been sampled atleast \( h(T) \) times in the first \( T \) time steps. A higher value of \( h(T) \) signifies higher confidence in the empirical mean value of the recommended arm.

While continuity of \( \mu \) is sufficient to guarantee the existence of a limiting-optimal algorithm, it must be replaced by a stronger condition in order to obtain lower bounds for probability of finding an \( \epsilon \)-optimal arm at a finite horizon. We assume that the mean function \( \mu \) is Lipschitz continuous.

**Theorem 4.1.** Let \( A \) be a near-optimal arm identification algorithm. Let \( x_1, x_2, \ldots, x_n \) be a sequence of arms which are sampled atleast \( h(T) \) times in the first \( T \) time steps. Let \( k_1, k_2, \ldots, k_n \) be the respective frequencies and \( \mu_1, \mu_2, \ldots, \mu_n \) be the respective means. Then for any \( \epsilon > 0 \), the inequality
\[ P\{\mathcal{E}\} \geq \sum_{i \in [n]} \left[ \prod_{j \in [n] \setminus \{i\}} \left( 1 - \exp\left( -\frac{2k_i k_j}{k_i + k_j} (\mu_i - \mu_j)^2 \right) \right) \right] \chi_{(Ld, \infty)}(\epsilon) \]
holds, where \( \mathcal{E} \) is the event that the recommended arm is \( \epsilon \)-optimal, \( L \) is the Lipschitz constant, \( \chi \) is the indicator function and \( \text{dist}(x_i, X^*) \) for \( i \in [n] \).

**Proof.** Let \( x_i, k_i, \mu_i \) for \( i \in [n] \) be as in the hypothesis of the theorem. Let us denote the empirical means of the arms after \( T \) time steps by \( \hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_n \). The algorithm recommends the arm \( x_T^A = x_{i_0} \), where \( i_0 = \arg\max_{i \in [n]} \{ \hat{\mu}_i \} \) to minimize simple regret. Let \( \mathcal{E} \) be the event that the recommended arm is \( \epsilon \)-optimal. Then we have
\[ P\{\mathcal{E}\} = \sum_{i \in [n]} P\{\hat{\mu}_i \geq \max_{j \in [n]} \hat{\mu}_j \} P\{\mu_i \geq \mu^* - \epsilon\}. \quad (5) \]
A strategy (algorithm) is called a fixed (static) sampling strategy if the probability of each arm being pulled at a finite time step is a function of the input $T$. An algorithm is said to be adaptive has no bearings on it being deterministic or randomized.

The first term in each of the products in the sum on the RHS of (5) can be written as

$$\mathbb{P}\{\hat{\mu}_i \geq \max_{j \in [n]} \hat{\mu}_j\} = \mathbb{P}\{\hat{\mu}_i \geq \hat{\mu}_1, \ldots, \hat{\mu}_i \geq \hat{\mu}_n\}. \tag{6}$$

The events $\hat{\mu}_i \geq \hat{\mu}_j$ for $j \in [n]$ are not independent, but we still have the inequality

$$\mathbb{P}\{\hat{\mu}_i \geq \hat{\mu}_1, \ldots, \hat{\mu}_i \geq \hat{\mu}_n\} \geq \prod_{j \in [n] \setminus \{i\}} \mathbb{P}\{\hat{\mu}_i \geq \hat{\mu}_j\} = \prod_{j \in [n] \setminus \{i\}} \mathbb{P}\{\hat{\mu}_i - \hat{\mu}_j \geq 0\}. \tag{7}$$

Each term in the product on the RHS of the above inequality can be decomposed as

$$\mathbb{P}\{\hat{\mu}_i - \hat{\mu}_j \geq 0\} = \mathbb{P}\{\hat{\mu}_i - \hat{\mu}_j \geq 0 \mid \mu_i > \mu_j\} + \mathbb{P}\{\hat{\mu}_i - \hat{\mu}_j \geq 0 \mid \mu_i \leq \mu_j\}. \tag{8}$$

Using the Corollary to Theorem 2 in [8], we have

$$\mathbb{P}\{\hat{\mu}_i - \hat{\mu}_j \geq 0 \mid \mu_i > \mu_j\} \geq 1 - \exp\left(-\frac{2k_i k_j}{k_i + k_j} (\mu_i - \mu_j)^2\right).$$

Hence, using (9) we have

$$\mathbb{P}\{\hat{\mu}_i - \hat{\mu}_j \geq 0\} \geq 1 - \exp\left(-\frac{2k_i k_j}{k_i + k_j} (\mu_i - \mu_j)^2\right) \tag{9}$$

for $i \neq j$. Let $d_i = dist(x_i, X^*) = \min_{x^* \in X^*} \rho(x_i, x^*)$, denote the distance of the point $x_i$ from the set of optimal arms. Since $\mu : \mathcal{U} \to [0, 1]$ is Lipschitz continuous, we have

$$\mu^* - \mu(x) \leq L \rho(x^*, x) \forall x^*, x \in X^*,$$

which implies

$$\mu^* - \mu_i \leq L d_i, \forall i \in [n]$$

where $L$ is the Lipschitz constant. Now, $\mathbb{P}\{\mu_i \geq \mu^* - \epsilon\}$ can take values only in the set $\{0, 1\}$. This can be guaranteed to be 1 if $\mu^* - \mu_i \leq L d_i \leq \epsilon$, which implies

$$\mathbb{P}\{\mu_i \geq \mu^* - \epsilon\} \geq \chi_{(Ld_i, \infty)}(\epsilon), \tag{10}$$

where $\chi$ is the indicator function. Finally, it follows using (5), (6), (7), (9) and (10) that

$$\mathbb{P}\{E\} \geq \sum_{i \in [n]} \prod_{j \in [n] \setminus \{i\}} \left(1 - \exp\left(-\frac{2k_i k_j}{k_i + k_j} (\mu_i - \mu_j)^2\right)\right) \chi_{(Ld_i, \infty)}(\epsilon).$$

\[\square\]

5 Algorithms and Performance

In this section, we first define two types of sampling strategies (as in [9]) and then consider examples from each category. In each case, we evaluate the algorithms empirically.

A strategy (algorithm) is called a fixed (static) sampling strategy if the probability of each arm being pulled at a finite time step $t$ is a function of the input $T$. Analogously, a strategy is said to be an adaptive sampling strategy if the probability of each arm being pulled at a finite time step $t$ is a function of the input $T$ as well as the history up to that point. Note that an algorithm being fixed or adaptive has no bearings on it being deterministic or randomized.

5.1 Empirical Evaluation of Algorithms

To evaluate the performance of an algorithm, we compute the expected simple regret where expectation is taken on the space of mean reward functions. To generate instances with arbitrary mean functions satisfying assumption given in Section [2], we make use of the Universal approximation theorem [10]. We wish to generate functions $\mu \in C([0, 1])$ with the additional condition that the reward distributions share the common compact support (the unit interval). The latter condition
leads to the restriction of the values of the function $\mu$ to the interval $[0, 1]$. We use functions of the following form to generate the $\mu$ function in accordance with the Universal approximation theorem.

$$
\mu(x) = \sum_{i=1}^{N} w_i \psi(a_i x + b_i)
$$

Here $\psi$ is any activation function. We choose to work with the Gaussian activation function with mean and standard deviation equal to 0 and 110 respectively. Moreover, we consider only positive values of $w_i$ for $i \in [N]$ with $\sum_{i=1}^{N} w_i = 1$ so that the values of the function $\mu$ belong to the interval $[0, 1]$. We use Bernoulli reward distributions corresponding to each arm. To find the expected simple regret $E[r_{A,T}]$, we use the following values: $N = 10$, $a_i \in U[0, 10]$, $b_i \in U[0, 1]$, $w_i \in U[0, 1]$. Figure 1 shows the mean function for two instances generated using the above method. We now study a few algorithms and analyze them empirically.

### 5.1.1 Round Robin Algorithm

We describe the round robin algorithm $A_{rr}$ below. For an input horizon $T$, the agent fixes a set of $g(T)$ arms spread uniformly across the interval $[0, 1]$ and samples these arms in a round robin fashion. The algorithm recommends the arm with highest empirical mean among these $g(T)$ arms. If the function $g(T)$ is unbounded and strictly sub-linear ($o(T)$), then the above algorithm is limiting-optimal.

Although $g$ satisfies the conditions mentioned at the end of Section 3, it alone is not sufficient to prove that $A_{rr}$ is limiting-optimal since this algorithm is not the same as the one constructed in
the proof of Theorem 3.1 (it doesn’t use a fixed dense set); but the proof goes through in a similar way. Figure 2 depicts the performance of the round robin algorithm for various strictly sub-linear unbounded functions \(g\). Figure 2b shows that the expected simple regret initially decreases with increasing exponent of \(T\) and then increases. This is as expected since there is a tradeoff between the confidence of the empirical estimate for each arm (higher for lower exponent values) and the number of arms taken for sampling (higher for higher exponent values).

5.1.2 Kiefer-Wolfowitz Algorithm

The problem of stochastic estimation of the maximum of a regression function was studied in [7]. They propose an iterative algorithm to solve the given problem for a certain family of mean functions (see Section 2 in [7]), with the additional condition of \(\mu\) being concave. The following sequence of iterates which converges to the unique point of maximum (both local as well as global).

\[ x_{n+1} = x_n + a_n \left( \frac{y_{2n} - y_{2n-1}}{a_n} \right) \]

Here, \(x_0\) is arbitrary in \(\mathcal{X}\), \(y_{2n}\) and \(y_{2n-1}\) are samples from the reward distributions \(f(x_n + c_n)\) and \(f(x_n - c_n)\) respectively, and \(a_n, c_n\) are infinite sequences of positive numbers satisfying the following properties.

\[ c_n \to 0 \text{ as } n \to \infty, \sum_{n=0}^{\infty} a_n = \infty, \sum_{n=0}^{\infty} a_n c_n < \infty, \sum_{n=0}^{\infty} a_n^2 c_n^{-2} < \infty \]

We choose the sequences defined by \(a_n = \frac{1}{2^{tn}}, c_n = \frac{1}{20^{tn}}, \forall n \in \mathbb{N}\). It is easy to verify that these sequences satisfy the above properties. The expected simple regret of the algorithm following this sequence of iterates reduces with increase in horizon, as is expected, but the rate of convergence is quite slow (when compared to the round robin algorithm) and no theoretical bounds on the rate of convergence can be found in the present literature. Figure 3a depicts the comparative performance of the above algorithm against the round robin algorithm.

5.1.3 Zooming Algorithm

As described in [5], the zooming algorithm defines a set \(S \subset \mathcal{X}\) of representative arms, each of which covers a confidence ball around it. The confidence radius of an arm at time step \(t\) with input horizon \(T\) is defined as

\[ r_t(x) = \sqrt{\frac{2 \log T}{n_t(x) + 1}} \]

Figure 3: Relative performance of various algorithms considered
where \( n_t(x) \) is the number of pulls of the arm \( x \) up to time \( t \). The confidence ball around a representative arm is defined as the interval

\[
B_t(x) = \{ y \in \mathcal{X} : \| x - y \| \leq r_t(x) \}.
\]

These intervals form a covering of the metric space and all \( y \in B_t(x) \) are approximated to \( x \) when required to be pulled. The confidence ball around an arm shrinks as the arm is sampled more and more times, and this leads to uncovering of some points. The following rules govern the algorithm.

**Activation rule:** Cover any uncovered point \( y \) by adding it to the set \( S \).

**Selection rule:** Use an extension of UCB algorithm and select interval to be pulled according to it.

At any time step \( t \), the agent pulls the arm \( x \) with maximum \( \text{index}_t(x) \) defined by

\[
\text{index}_t(x) = \mu_t(x) + 2r_t(x),
\]

where \( \mu_t(x) \) is the empirical mean for arm \( x \) at time \( t \). UCB is used to exploit and explore appropriately and the activation rule ensures more sampling of arms near frequently chosen arms i.e. arms which are near the optimal arms. The simple regret achieved by the above algorithm at horizon \( t \) satisfies the following bound with probability \( \geq 1 - \frac{1}{t^2} \) (Claim 4.5, Lemma 4.6 in [5], see also [9]).

\[
\mu^* - \mu_t(x) \leq 3r_t(x)
\]

Note that this bound can be used to estimate the input horizon in order to obtain an \( \epsilon \)-optimal arm. This algorithm performs quite well for the cumulative regret minimization problem. But since our objective is to minimize the simple regret, the algorithm requires more exploration. Hence, we modify the algorithm by redefining confidence radius and index function as follows.

\[
r_t(x) = \sqrt{\frac{2\gamma}{n_t(x) + 1}}
\]

\[
\text{index}_t(x) = \mu_t(x) + \alpha r_t(x)
\]

Taking \( \gamma (=0.01) \) to be small, we ensure that more representative arms are required to cover the entire metric space. A larger value is assigned to \( \alpha (=10) \) in order increase the extent of exploration while also improving confidence in the neighbourhood which could be exploited.

As can be seen in Figure [3b], the standard zooming algorithm doesn’t perform very well in the pure exploration setting. On the other hand, the modified zooming algorithm and the round robin algorithm show good performance. For small horizon values, the modified zooming algorithm works better when compared to the round robin algorithm while the case reverses for larger horizon values. This is because for larger horizon values, the round robin algorithm samples a large number of arms at fine granularity and that too with high confidence empirical estimates.

### 6 Conclusion

In this paper, we first gave an elementary proof for the existence of a limiting-optimal algorithm if the metric space of bandit arms is separable. The proof uses the \( 3\epsilon \) argument from mathematical analysis and the Hoeffding’s inequality. Subsequently, we obtained a general lower bound on the finite horizon probability of the arm recommended by an algorithm being \( \epsilon \)-optimal. This was done by using Hoeffding’s inequality to compute confidence values and the Lipschitz condition to compute bounds on mean values in the neighbourhood of a point.

In Section [5] we described a method to evaluate algorithms empirically using simple regret averaged over instances with mean functions generated using the Gaussian activation function. Using this approach, we compared the performance of the following algorithms - round robin, Kiefer-Wolfowitz, zooming and modified zooming. Based on these experiments, we found that the rate of convergence of the sequence of arms in the Kiefer-Wolfowitz algorithm is very low. We finally conclude that the modified zooming algorithm performs the best for small horizon values while the round robin algorithm performs the best for larger horizon values.

A major potential future work in the direction of this paper could be to analyze the discussed algorithms theoretically in a spirit similar to Theorem [4.1]. In this regard, some insights and headway in case of the zooming algorithm are already discussed in [5]. It would also be interesting to see if one can design more Kiefer-Wolfowitz type algorithms with faster convergence rates. From the results obtained in this paper, we speculate that it might be possible to design a better adaptive sampling algorithm which performs atleast as good as the round robin algorithm for larger horizon values.
References


