

Optimal Algorithms for Range Searching over Multi-Armed Bandits

Siddharth Barman^{1*}, Ramakrishnan Krishnamurthy¹ and Saladi Rahul¹

¹Indian Institute of Science, Bengaluru
{barman, kramakrishna, saladi}@iisc.ac.in

Abstract

This paper studies a multi-armed bandit (MAB) version of the range-searching problem. In its basic form, range searching considers as input a set of points (on the real line) and a collection of (real) intervals. Here, with each specified point, we have an associated weight, and the problem objective is to find a maximum-weight point within every given interval. The current work addresses range searching with stochastic weights: each point corresponds to an arm (that admits sample access) and the point’s weight is the (unknown) mean of the underlying distribution. In this MAB setup, we develop sample-efficient algorithms that find, with high probability, near-optimal arms within the given intervals, i.e., we obtain PAC (probably approximately correct) guarantees. We also provide an algorithm for a generalization wherein the weight of each point is a multi-dimensional vector. The sample complexities of our algorithms depend, in particular, on the size of the optimal hitting set of the given intervals. Finally, we establish lower bounds proving that the obtained sample complexities are essentially tight. Our results highlight the significance of geometric constructs (specifically, hitting sets) in our MAB setting.

1 Introduction

Range searching is a fundamental problem in computational geometry and database theory; see, e.g., [Agarwal and Erickson, 1999; Agarwal, 2017; Gupta *et al.*, 2018; Rahul and Tao, 2019]. This problem has been extensively studied over the past few decades, and applications of range searching (along with its variants) arise in numerous real-world domains, such as spatial databases [Samet, 1990], temporal databases [Agarwal *et al.*, 2003], and networking [Li *et al.*, 2003].

In its basic form, range searching considers as input a set of points in \mathbb{R} —each with an associated weight—and a collection of (real) intervals.¹ The problem’s objective is to efficiently find, for every given interval I , an input point within I with maximum weight.

Note that this classic formulation of range searching addresses fixed (deterministic) weights, i.e., one assumes that the weight of each point is known a priori. By contrast, many modern applications require queries to be processed on uncertain (stochastic) data. As a stylized example of such a setting, consider a paid crowdsourcing platform wherein the competency of the participants (say, in performing a type of task) is not known beforehand. Here, each participant a can be represented as a point $p_a \in \mathbb{R}$ —which denotes the fixed monetary payment for a —and the competency (weight) of a can be modeled as a stochastic quantity. In this setup, range searching corresponds to the natural problem of finding, for each queried price range (i.e., within each given interval), the most proficient crowdworker.

Motivated, in part, by such applications, we study range searching in the multi-armed bandit (MAB) framework. In particular, we address a natural formulation wherein each input point $p_a \in \mathbb{R}$ is associated with a random variable (drawn from an unknown distribution) and the weight of the point corresponds to the (a priori unknown) mean, μ_a , of the random variable. That is, each given point is an arm and the objective is to identify an optimal (with respect to μ_a) arm within each given interval. Here, as is standard in the MAB literature, we assume sample access to the random variables associated with the arms/points. The current work develops novel, sample-efficient algorithms for this stochastic version of range searching.

We also design algorithms for a pertinent generalization wherein the weight of each point is a d -dimensional vector, i.e., the random variables associated with every arm/point and, hence, the means μ_a s are d -dimensional vectors. For the crowdsourcing example mentioned above, this generalization models settings in which the proficiency of each crowdworker needs to be assessed across d different types of tasks.² In this multi-dimensional setting, the problem objective extends, from finding an optimal arm per interval, to computing a set of *Pareto optimal* arms within each input interval.

Our Contributions and Techniques. Given the MAB nature of the problems, we obtain PAC-style (probably approximately correct) results, i.e., the developed algorithms compute, with high probability, near-optimal arms for every given

*Contact Author

¹The given intervals are not necessarily disjoint.

²As before, the monetary payment for each crowdworker is a fixed and known number.

interval.

The sample complexities of our algorithms depend, in particular, on the size of the optimal *hitting set* of the given collection of intervals. Recall that a (finite) set of points, $\text{HS} \subset \mathbb{R}$,³ is said to be a *hitting set* for a collection of intervals $\mathcal{I} = \{I_1, I_2, \dots, I_q\}$ iff HS contains a point from the interior of each interval $I_i \in \mathcal{I}$. Let τ be the size of a minimum-cardinality hitting set for the queried collection of intervals \mathcal{I} and n be the number of input points (arms). We show that⁴

- (i) For range searching with single-dimensional weights ($\mu_a \in \mathbb{R}$), i.e., for finding (near) optimal arms for every interval, the number of samples required by our algorithm is $O\left(\frac{n}{\varepsilon^2} \log\left(\frac{\tau}{\varepsilon\delta}\right)\right)$ (Theorem 1). Here, $\varepsilon > 0$ is the approximation parameter and $\delta > 0$ is the confidence parameter in the PAC guarantee.
- (ii) For range searching with multi-dimensional weights ($\mu_a \in \mathbb{R}^d$), i.e., for finding (near) Pareto optimal arms for every interval, the number of samples required by our algorithm is $O\left(\frac{nd}{\varepsilon^2} \log\left(\frac{\tau^d}{\varepsilon\delta}\right)\right)$ (Theorem 3).
- (iii) Finally, we establish lower bounds proving that the sample upper bounds obtained in our algorithms are essentially tight (Theorem 4). Specifically, these results show that a sample complexity dependence on $\log \tau$ is unavoidable, in general.

Notably the above-mentioned upper bounds do not explicitly depend on the number of input intervals, q . Indeed, the size of an optimal hitting set, τ , can be significantly smaller than q , e.g., $\tau = 1$ for a collection of intervals with a common intersection. Even in the worst case (with pairwise-disjoint intervals), we have $\tau = q$; one can always select the midpoint of each given interval to obtain a hitting set. Notably, a guarantee in terms of τ provides refined (and matching) upper and lower bounds. It also highlights a novel application of this geometric parameter in the current MAB setting.

At a high level, our algorithm for single-dimensional weights proceeds by using the τ points in the optimal hitting set to construct $\tau + 1$ pairwise-disjoint intervals (referred to as *slabs*) that cover all the input intervals (which themselves might have multiple intersections). A key idea then is to identify—through relevant subroutines—a candidate set of arms within each of the $\tau + 1$ constructed slabs and judiciously combine them to find the desired set of near-optimal arms for the input intervals. The subroutines employed to populate the candidate set of arms are based on variations of a result of Cheu et al. [2018] that addresses the *skyline problem* (with scalar weights).

We build upon this design template, with additional technical insights, to obtain an algorithm for d -dimensional weights. Towards a subroutine, we develop a PAC algorithm for the d -dimensional version of the skyline problem (see Subsection 4.1 for details). Note that the algorithm provided in [Cheu et al., 2018] solely address the single-dimensional

version of the skyline problem. Furthermore, even the scalar instantiation (obtained by setting $d = 1$) of our skyline algorithm is distinct from that of Cheu et al. [2018].

Related Work. For a broad survey of MAB literature see, e.g., [Slivkins, 2019]. Even-Dar et al. [2006] develop a PAC algorithm for the problem of identifying a best arm among all the n given ones; see also [Domingo et al., 2002]. The algorithm of Even-Dar et al. has a sample complexity of $O\left(\frac{n}{\varepsilon^2} \log \frac{1}{\delta}\right)$ and a matching lower bound was obtained in [Mannor and Tsitsiklis, 2004]. Multiple variants of best arm identification have also been studied in the literature; see, e.g., [Audibert and Bubeck, 2010; Bubeck et al., 2013; Kalyanakrishnan and Stone, 2010; Garivier and Kaufmann, 2016; Russo, 2016; Yu et al., 2018]. Note that best arm identification can be viewed as a special case of bandit range searching (by considering a single input interval that contains all the arms).

Addressing settings in which each arm is associated with a d -dimensional random variable, the work of Auer et al. [2016] provides a PAC algorithm—with sample complexity $O\left(\frac{n}{\varepsilon^2} \log \frac{nd}{\delta}\right)$ —for finding Pareto optimal arms. This setting is generalized in our range-searching framework with d -dimensional weights.

Cheu et al. [2018] study a setup wherein the arms are positioned on the real line and they provide a PAC algorithm to find a *skyline*. Bandit range searching generalizes this skyline problem as well. Complementarily, in the case of single-dimensional weights, one can use the algorithm of Cheu et al. [2018] as a subroutine for range searching. For multi-dimensional weights, we in fact develop a novel algorithm for computing skylines. This result generalizes the work of Cheu et al. [2018] (to multi-dimensional weights) and is potentially interesting in its own right. The technical connections between range searching and skylines are detailed in subsequent sections.

Multiple recent results in computational geometry have addressed the (classic) range searching problem over uncertain data [Tao et al., 2007; Agarwal et al., 2009; Agarwal et al., 2018; Li and Wang, 2016]. These works primarily address uncertainty in the point locations. By contrast, the current setting studies fixed point locations, but stochastic weights.

Naive Algorithms: A direct approach for the problem at hand is to first compute the empirical estimates of the weights (means) by sampling each arm $O\left(\frac{1}{\varepsilon^2} \log \frac{n}{\delta}\right)$ times and then report the optimal arms with respect to the estimates. This method requires $O\left(\frac{n}{\varepsilon^2} \log \frac{n}{\delta}\right)$ samples overall; see, e.g., the NAIVE algorithm in [Even-Dar et al., 2006] for a PAC analysis. One can also show (by considering a union bound over the intervals, instead of the arms) that $O\left(\frac{n}{\varepsilon^2} \log \frac{q}{\delta}\right)$ samples suffice; here q is the number of input intervals. Notably, the current work goes beyond these generic approaches and invokes problem-specific insights (e.g., the use of hitting sets) to develop refined upper bounds. Our tight lower bounds further substantiate the relevance of the obtained refinement.

2 Notation and Preliminaries

The paper studies settings in which one has access to independent draws from n unknown distributions, each supported

³The hitting set is not required to be a subset of the input points.

⁴Note that, even though the single-dimensional-weights setting ($d = 1$) is a special case of the multi-dimensional one, we present the settings separately to highlight the layered development of the algorithmic ideas.

on $[0, 1]$, i.e., we have sample access to n arms. Throughout, $\mu_a \in [0, 1]$ will be used to denote the (unknown) mean under the a th distribution and we will refer to μ_a as the *weight* of arm $a \in [n] := \{1, 2, \dots, n\}$.

Furthermore, in our framework, each arm $a \in [n]$ is associated with a (fixed) point $p_a \in \mathbb{R}$. The set of points $\mathcal{P} := \{p_1, \dots, p_n\}$ is given to us as input, along with a collection of q real intervals $\mathcal{I} := \{I_i = [\ell_i, r_i]\}_{i=1}^q$. Without loss of generality, we will assume that each given interval has a non-empty interior, i.e., for each interval $I_i = [\ell_i, r_i]$ we have $\ell_i < r_i$. For any interval $J = [\ell, r]$, write $\mathcal{A}(J)$ to denote the set of arms belonging to J , i.e., $\mathcal{A}(J) := \{a \in [n] \mid \ell \leq p_a \leq r\}$.

The current work studies the *bandit range searching* problem: given a set of points $\mathcal{P} = \{p_a\}_{a=1}^n$, a collection of intervals $\mathcal{I} = \{I_i\}_{i=1}^q$, and sample-access to n arms, find—for each given interval I_i —an arm $a \in \mathcal{A}(I_i)$ with maximum weight μ_a .

Given the multi-armed bandit nature of this problem, we obtain PAC (probably approximately correct) guarantees; in particular, we develop sample-efficient algorithms that find, with high-probability, near-optimal arms for each given interval. We next define the PAC constructs, with approximation parameter $\varepsilon > 0$ and confidence parameter $\delta > 0$.

Definition 1 (ε -optimality). *For any interval I and parameter $\varepsilon > 0$, an arm $a \in \mathcal{A}(I)$ is said to be ε -optimal for I iff $\mu_a \geq \mu_b - \varepsilon$ for all $b \in \mathcal{A}(I)$.*

Definition 2 ((ε, δ) -PAC guarantee). *An algorithm ALG is said to achieve the (ε, δ) -PAC guarantee for the bandit range searching problem, iff—given any problem instance $(\mathcal{P}, \mathcal{I})$ —ALG finds, with probability at least $(1 - \delta)$, an ε -optimal arm for every interval $I \in \mathcal{I}$.*

Range Searching with Multi-Dimensional Weights. We will also develop algorithms for a generalization of bandit range searching wherein the weights are multi-dimensional vectors. In this generalization, for every arm, the underlying random variable is d -dimensional and component-wise supported on $[0, 1]$. That is, for each arm $a \in [n]$ the (unknown) weight $\mu_a \in [0, 1]^d$. The range aspects of the problem remain unchanged: each arm a is associated with a (given) point $p_a \in \mathbb{R}$ and, as before, we are given a collection of intervals $\mathcal{I} = \{I_i = [\ell_i, r_i]\}_{i=1}^q$. In this setup, since the weights are d -dimensional vectors (instead of scalars), the problem objective extends to finding ε -Pareto optimal arms (instead of ε -optimal arms).

Note that in the single-dimensional case, if a was an ε -optimal arm in set \mathcal{A} , then (hypothetically) adding ε to the weight μ_a , and keeping the weight of all the other arms unchanged, ensured proper optimality of a . The notion of ε -Pareto optimality (defined next) essentially builds upon this perspective. It deems a set of arms T to be ε -Pareto optimal if (hypothetically) adding $\varepsilon \mathbf{1}$ to the d -dimensional weight of each arm in T (and keeping the weight of the other arms unchanged) gives us a legitimately Pareto optimal set. Formally,

Definition 3 (ε -Pareto optimality). *For any interval I and parameter $\varepsilon > 0$, a subset $T \subseteq \mathcal{A}(I)$ is said to be ε -Pareto optimal for interval I iff (a) for all arms $b \in \mathcal{A}(I)$ there exists an arm $a \in T$ such that $\mu_a \geq \mu_b - \varepsilon \mathbf{1}$, and (b) for all arms*

$a \in T$ there does not exist an arm $b \in \mathcal{A}(I)$ such that $\mu_b \geq \mu_a + \varepsilon \mathbf{1}$.

Here, the inequalities between weights are enforced component-wise and $\mathbf{1}$ denotes the all-ones vector. We note that, up to pruning, a set T that satisfies Definition 3 also upholds the approximate-optimality criterion (specifically, Success Condition 2) studied in [Auer *et al.*, 2016].

Analogous to the single-dimensional case, we define the PAC guarantee under d -dimensional weights: an algorithm ALG is said to be (ε, δ) -correct for bandit range searching with multi-dimensional weights, iff—given any problem instance $(\mathcal{P}, \mathcal{I})$ —ALG finds, with probability at least $(1 - \delta)$, an ε -Pareto optimal set of arms for every interval $I \in \mathcal{I}$.

Hitting Set and Slabs. A finite set of points $\text{HS} \subset \mathbb{R}$ is said to be a hitting set for a collection of intervals $\mathcal{I} = \{I_i = [\ell_i, r_i]\}_{i=1}^q$ iff for each $I_i = [\ell_i, r_i] \in \mathcal{I}$ there exists a point $e \in \text{HS}$ such that $\ell_i < e < r_i$.

Let HS^* denote a minimum-cardinality hitting set for the given collection of intervals \mathcal{I} and write $\tau := |\text{HS}^*|$.⁵ Note that the size of an optimal hitting set, τ , is at most the number of intervals in \mathcal{I} , i.e., $\tau \leq q$.

Our algorithms use the τ points in an optimal hitting set HS^* and construct $\tau + 1$ intervals that intersect only at their endpoints and partition the real line. In particular, with $e_1 < e_2 < \dots < e_\tau$ denoting the set of points in HS^* , define the following collection of $\tau + 1$ intervals: $S_0 := (-\infty, e_1]$, $S_\tau := [e_\tau, \infty)$, and $S_j := [e_j, e_{j+1}]$ for $1 \leq j < \tau$.

We will refer to these $\tau + 1$ intervals as *slabs*. Note that, by construction, the slabs are indexed in increasing order of their left (or, equivalently, right) endpoints. Furthermore, the slabs have pairwise-disjoint interiors and their union covers all the input intervals, $\cup_{I \in \mathcal{I}} I \subseteq \cup_{j=0}^{\tau} S_j$. We will also utilize the following property of slabs, which follows from the fact that their endpoints constitute a hitting set: for every input interval $I \in \mathcal{I}$ there exists a sequence of *at least two* slabs, S_x, S_{x+1}, \dots, S_y , that intersect with I , i.e.,

$$I \cap S_t \neq \emptyset \quad \text{for } x \leq t \leq y \quad (\text{P})$$

Property (P) implies that every input interval gets partitioned among two or more slabs.

3 Single-Dimensional Weights

We next state our main result for bandit range searching under single-dimensional weights. Since the following theorem can be obtained by instantiating the multi-dimensional generalization (Theorem 3 with $d = 1$); a detailed description of this scalar setting is deferred to the full version of this paper.

Theorem 1. *Given any problem instance $(\mathcal{P}, \mathcal{I})$ with n arms, algorithm ALG-RS draws $O\left(\frac{n}{\varepsilon^2} \log\left(\frac{\tau}{\varepsilon\delta}\right)\right)$ samples and achieves the (ε, δ) -PAC guarantee for bandit range searching (with single-dimensional weights); here τ denotes the size of the minimum-cardinality hitting set for \mathcal{I} .*

⁵Given a collection of intervals \mathcal{I} , an optimal hitting set HS^* (and, hence, τ) can be computed in polynomial-time, via a greedy algorithm.

4 Multi-Dimensional Weights

This subsection develops our PAC algorithm for range-searching with d -dimensional weights. Specifically, for each arm $a \in [n]$, the (unknown) weight/mean is a d -dimensional vector, $\mu_a \in [0, 1]^d$, and the problem objective is to find ε -Pareto optimal arms (Definition 3) within each input interval.

The range aspects of the setup are as mentioned previously: each arm $a \in [n]$ is associated with a (given) point $p_a \in \mathbb{R}$ and we are also given a collection of intervals $\mathcal{I} = \{I_i = [\ell_i, r_i]\}_{i=1}^q$.

Our algorithm, ALG-D-RS (Algorithm 2 in Section 4.2), begins by constructing slabs $\{S_j\}_j$ from an optimal hitting set of the input collection intervals \mathcal{I} . Then, ALG-D-RS executes the following two subroutines for each slab S_j : D-LSKY(\cdot) and D-RSKY(\cdot). The subset of arms collected through these executions serve as a candidate set C and the algorithm computes the desired set of arms by selecting Pareto optimal (with respect to the empirically estimated weights) *candidate* arms within each input interval I_i .

The subroutines D-LSKY and D-RSKY solve the *skyline problem*, which in fact can be viewed as a special case of bandit range searching. Specifically, in the skyline problem the points associated with the arms, $\{p_a \in \mathbb{R}\}_a$, themselves generate the collection of query intervals \mathcal{I} : in the left-skyline problems we have $\mathcal{I} = \{[p_a, p_\infty]\}_a$ and in the right-skyline problems $\mathcal{I} = \{[p_{-\infty}, p_a]\}_a$, where $p_\infty := \max_a p_a$ and $p_{-\infty} := \min_a p_a$.

Prior work has only focused on the single-dimensional version of the skyline problem (see [Cheu *et al.*, 2018]). Hence, towards implementations of the subroutines D-LSKY and D-RSKY, in Section 4.1 we develop a novel algorithm for computing ε -skylines with d -dimensional weights.

4.1 Skyline with Multi-Dimensional Weights

In this section we develop a PAC algorithm for the skyline problem with weights in $[0, 1]^d$.

Definition 4 (ε -skyline for d -dimensional weights). *For an interval $J = [\ell, r]$ and parameter $\varepsilon > 0$, a set of arms $L \subseteq \mathcal{A}(J)$ is said to be an ε -left-skyline within interval J iff (i) for each arm $b \in \mathcal{A}(J)$, the set L contains an ε -Pareto optimal set T for the interval $[p_b, r]$, and (ii) each arm in L is ε -Pareto optimal for some interval $[p_b, r]$, with $b \in \mathcal{A}(J)$.*

Analogously, an ε -right-skyline, R , is defined considering intervals $\{[\ell, p_b]\}_{b \in \mathcal{A}(J)}$. The following characterization of ε -skylines will be used in the analysis; all the missing proofs appear in the full version of this paper.

Proposition 1. *A set of arms $L \subseteq \mathcal{A}(J)$ is an ε -left-skyline within interval $J = [\ell, r]$ iff (i) for every arm $b \in \mathcal{A}(J)$, there exists an arm $\beta \in L$ that satisfies $p_\beta \geq p_b$ and $\mu_\beta \geq \mu_b - \varepsilon \mathbf{1}$, and (ii) for each arm $\beta \in L$, there does not exist an arm $x \in \mathcal{A}(J)$ such that $p_x \geq p_\beta$ and $\mu_x \geq \mu_\beta + \varepsilon \mathbf{1}$.*

In Proposition 1—and in the remainder of this section—the inequalities are enforced component-wise. At a high level, the first condition of the proposition requires that for each arm b in the given interval, L contains a dominating (componentwise and within a factor of ε) point to the right of b . The second condition mandates that any arm in L is itself not

dominated (componentwise and beyond a factor of ε) by arms to its right.

Our algorithm, D-LSKY (Algorithm 1), starts by setting A_1 as the set of all arms within the given interval and ε_1 as an initial approximation parameter. The algorithm then considers a partition of $[0, 1]^d$ into hypercubes of side length $\varepsilon_1/4$. Drawing a conservative number of samples, the algorithm finds an estimate $\hat{\mu}_a \in [0, 1]^d$ for each arm $a \in A_1$ and partitions the set of arms A_1 itself based on how the estimates fall across different hypercubes.

For any hypercube B , let $B(A_1)$ be the set of arms (in A_1) whose estimates are within B . We will establish that, if the cardinality of $B(A_1)$ is sufficiently large, then we can drop half of the arms in $B(A_1)$ from consideration, and still not lose ε -Pareto dominating arms. In particular, from every large-sized set $B(A_1)$, the algorithm removes from consideration half of the arms $x \in B(A_1)$ whose p_x value is less than the median of $\{p_y\}_{y \in B(A_1)}$. This update ensures that the number of arms under consideration decreases geometrically, and at the same time, the set of arms that remain, say A_2 , continues to include an approximate skyline. The algorithm repeats analogous steps over A_2 , with an updated approximation parameter $\varepsilon_2 = 3\varepsilon_1/4$. Iteratively, the algorithm continues until the number of remaining arms is sufficiently small. At this point, the algorithm concludes by sampling the final set of arms A_T (to obtain accurate-enough estimates of their weights) and returning a skyline (based on the final estimates) within A_T .

We will now define constructs that will be useful in the design and analysis of our algorithm. For $\eta \in (0, 1]$, define a lattice $L(\eta)$ as the set of vectors in $[0, 1]^d$ whose components are integer multiples of η , i.e., $L(\eta) := \{(\kappa_1\eta, \kappa_2\eta, \dots, \kappa_d\eta) \in [0, 1]^d \mid \kappa_1, \dots, \kappa_d \in \mathbb{Z}\}$. For each vector $g \in L(\eta)$, define a (hyper) cube around it as $B(\eta, g) := \{\nu \in [0, 1]^d \mid \|\nu - g\|_\infty \leq \frac{\eta}{2}\}$. Note that the collection of cubes $\{B(\eta, g)\}_{g \in L(\eta)}$ forms a cover of $[0, 1]^d$. Faces of these cubes can intersect. However, for ease of presentation and given that the interiors of the cubes are pairwise disjoint, we will consider them to constitute a partition of $[0, 1]^d$; in particular, by breaking ties arbitrarily, we will assume that each $\nu \in [0, 1]^d$ belongs to exactly one cube in $\{B(\eta, g)\}_{g \in L(\eta)}$.

In each iteration t , with approximation parameter $\varepsilon_t > 0$, the algorithm considers the collection of cubes obtained by setting $\eta = \varepsilon_t/4$, i.e., it considers $\mathcal{B}_t := \{B(\frac{\varepsilon_t}{4}, g)\}_{g \in L(\frac{\varepsilon_t}{4})}$.

Note that the cardinality of \mathcal{B}_t is at most $\left(\frac{4}{\varepsilon_t}\right)^d$.⁶

In the analysis, we will consider a *representative arm* $r(B)$ for each cube $B \in \mathcal{B}_t$: among all arms b whose weight μ_b is in B , arm $r(B)$ is the one with the maximum $p_{r(B)}$ value.

Definition 5 (Representative Arms). *In any iteration t —with set of arms A_t under consideration—an arm $r(B) \in A_t$ is said to be the representative of cube $B \in \mathcal{B}_t$ iff $\mu_{r(B)} \in B$ and $p_{r(B)} \geq p_b$ for all arms $b \in A_t$ whose weight $\mu_b \in B$.*

⁶The number of hypercubes is actually $(\lceil \frac{4}{\varepsilon_t} \rceil)^d$, but we assume $\frac{4}{\varepsilon_t}$ is an integer for ease of analysis.

Algorithm 1 D-LSKY: finds ε -left-skyline with probability at least $1 - \delta$

Input: Interval J , with sample access to arms in $\mathcal{A}(J)$, points $\{p_a\}_{a \in \mathcal{A}(J)}$, and parameters $\varepsilon, \delta > 0$. **Output:** Set of arms $S \subseteq \mathcal{A}(J)$

- 1: Initialize set $A_1 = \mathcal{A}(J)$, parameters $\varepsilon_1 = \varepsilon/5$, $\delta_1 = \delta/2$, and iteration count $t = 1$
- 2: **while** $|A_t| > 120 \left(\frac{4}{\varepsilon_t}\right)^d$ **do**
- 3: For each arm $a \in A_t$, find an empirical estimate, $\hat{\mu}_a \in [0, 1]^d$, of its weight, by sampling the arm $\frac{8}{\varepsilon_t^2} \log \left(\left(\frac{4}{\varepsilon_t}\right)^d \frac{50d}{\delta_t} \right)$ times
- 4: Initialize the set of arms to drop, $D_t = \emptyset$
- 5: **for all** cubes $B \in \mathcal{B}_t$ **do**
- 6: Write $B(A_t) := \{x \in A_t \mid \hat{\mu}_x \in B\}$ $\{B(A_t)$ is defined using the *estimated* weights}
- 7: If $|B(A_t)| > \frac{|A_t|}{10} \left(\frac{\varepsilon_t}{4}\right)^d$, then $D_t \leftarrow D_t \cup \{x \in B(A_t) : p_x < \text{the median of } \{p_y\}_{y \in B(A_t)}\}$
- 8: **end for**
- 9: Set $A_{t+1} = A_t \setminus D_t$, $\varepsilon_{t+1} = 3\varepsilon_t/4$, $\delta_{t+1} = \delta_t/2$, and update $t \leftarrow t + 1$
- 10: **end while**
- 11: For each arm $a \in A_t$, find an estimate, $\hat{\mu}_a \in [0, 1]^d$, of its weight, by sampling the arm $\frac{1}{\varepsilon_t^2} \log \left(\frac{|A_t|d}{\delta_t} \right)$ times
- 12: Set $\tilde{D} := \{z \in A_t \mid p_z \leq p_x \text{ and } \hat{\mu}_z < \hat{\mu}_x \text{ for some } x \in A_t\}$ $\{\text{With respect to the estimates, condition (ii) of Proposition 1 does not hold for arms in } \tilde{D}\}$
- 13: **return** $A_t \setminus \tilde{D}$

Write $R_t := \{r(B)\}_{B \in \mathcal{B}_t}$ to denote the set of all representatives in iteration t . Note that representative arms are defined considering the exact (but unknown) weights $\mu_b \in [0, 1]^d$. Still the definition is relevant, since we use these arms solely for the purposes of analysis, and not in the algorithm (which only has access to estimated weights $\hat{\mu}_b$ s). Also, note that the size of R_t is at most that of \mathcal{B}_t and, hence, $|R_t| \leq \left(\frac{4}{\varepsilon_t}\right)^d$.

The next lemma (Lemma 2) highlights a significance of representative arms—it asserts that R_1 (the set of representative arms in the first iteration) satisfies condition (i) of Proposition 1, with an (absolute) approximation factor of $\varepsilon/20$. Then, Lemma 3 (stated below) guarantees that, with high probability, the algorithm essentially maintains condition (i) between consecutive R_t s.

Lemma 2. For every arm $b \in \mathcal{A}(J)$, there exists an arm $r \in R_1$ that satisfies $p_r \geq p_b$ and $\mu_r \geq \mu_b - \frac{\varepsilon}{20}$.

Lemma 3. In any iteration t , with probability $1 - \delta_t$, for every representative arm $r \in R_t$, there exists an arm $\gamma \in R_{t+1}$ such that $\mu_\gamma \geq \mu_r - \varepsilon_t$ and $p_\gamma \geq p_r$.

The following lemma bounds the number of samples required by D-LSKY.

Lemma 4. For any given interval J with m arms ($m = |\mathcal{A}(J)|$), D-LSKY draws $O\left(\frac{md}{\varepsilon^2} \log \frac{d}{\varepsilon\delta}\right)$ samples.

We next state and establish the main result of this section.

Theorem 2. Given any interval J with m arms, D-LSKY draws $O\left(\frac{md}{\varepsilon^2} \log \frac{d}{\varepsilon\delta}\right)$ samples and returns an ε -left-skyline within J with probability at least $1 - \delta$.

Proof Sketch. In the interest of space, we defer the technical details to the full version of this paper and provide a proof sketch here. The stated sample complexity of D-LSKY follows directly from Lemma 4. Hence, we establish the theorem by showing that the returned set $A_T \setminus \tilde{D}$ satisfies both the conditions in Proposition 1 and, hence, is an ε -left-skyline; here T denotes the total number of iteration of the while loop in D-LSKY.

Towards this, first note that the representative set $R_T \subseteq A_T$ satisfies condition (i) of Proposition 1, with probability at least $1 - \sum_{t=1}^{T-1} \delta_t$. This follows from Lemma 2 and a repeated application of Lemma 3. Furthermore, the sampling in Line 11 ensures that, with probability $1 - \delta_T$, we have $\|\hat{\mu}_a - \mu_a\|_\infty \leq \varepsilon_T \leq \varepsilon/5$ for every arm $a \in A_T$. For the rest of the proof we will assume that these bounds hold and A_T satisfies condition (i). It suffices to prove the stated claim under this assumption, since it holds with probability at least $1 - \sum_{t=1}^{T-1} \delta_t - \delta_T \geq 1 - \delta$.

We note that $A_T \setminus \tilde{D}$ continues to satisfy condition (i) and it also bears conditions (ii) of Proposition 1. Condition (i) holds (with an additional approximation loss of at most $\varepsilon/20$), since in \tilde{D} we are only removing arms that are themselves Pareto dominated, under accurate-enough estimates. Similarly, conditions (ii) follows from the fact that in $A_T \setminus \tilde{D}$ we are retaining arms that are *not* Pareto dominated with respect to sufficiently-accurate estimates $\hat{\mu}_a$ s, i.e., the retained arms could not have been Pareto dominated under μ_a s beyond an absolute approximation factor of ε . Overall, we get that the returned set $A_T \setminus \tilde{D}$ bears both the conditions and, hence, it is an ε -left-skyline. \square

Analogous to Theorem 2, one can achieve a PAC guarantee for ε -right-skylines by executing D-LSKY on negated p_a values.

4.2 Algorithm for Multi-Dimensional Weights

This subsection develops an algorithm for bandit range searching with d -dimensional weights. As mentioned previously, this result is obtained by reducing range searching to finding ε -skylines over slabs.

In particular, given interval J and parameters $\varepsilon', \delta' > 0$, the subroutine D-LSKY(J, ε', δ') finds, with probability at least $1 - \delta'$, an ε' -left-skyline within J . Similarly, D-RSKY(J, ε', δ') finds, with probability at least $1 - \delta'$, an ε' -right-skyline within J . These subroutines—in addition to identifying the desired set of arms—provide an empirical estimate $\hat{\mu}_a \in [0, 1]^d$ of the true weight (mean) $\mu_a \in [0, 1]^d$ of each returned arm a . Specifically, subsumed within the same success probability as before, for every returned arm a we have an empirical estimate $\hat{\mu}_a$ which satisfies $\|\mu_a - \hat{\mu}_a\|_\infty \leq \varepsilon'$. This property is satisfied by the algorithms in Section 4.1.

By executing these two subroutines over each slab, ALG-D-RS populates a set of candidate arms \mathcal{C} along with

Algorithm 2 ALG-D-RS: (ε, δ) -PAC algorithm for range searching d -dimensional weights

Input: Set of points $\mathcal{P} = \{p_a \in \mathbb{R}\}_{a=1}^n$, collection of intervals $\mathcal{I} = \{I_i = [\ell_i, r_i]\}_{i=1}^q$, and sample access to the n arms, along with parameters $\varepsilon > 0$ and $\delta > 0$.

Output: Subsets of arms $\{T_i\}_{i=1}^q \subseteq [n]$

- 1: Let $e_1 < e_2 < \dots < e_\tau$ constitute a minimum-size hitting set for \mathcal{I} , and write $e_0 = -\infty$ and $e_{\tau+1} = +\infty$
- 2: Define slabs $S_j := [e_j, e_{j+1}]$, for $0 \leq j \leq \tau$
- 3: For every slab S_j , with $0 \leq j \leq \tau$, set left skyline $L_j = \text{D-LSKY}\left(S_j, \frac{\varepsilon}{3}, \frac{\delta}{2(\tau+1)}\right)$ and right skyline $R_j = \text{D-RSKY}\left(S_j, \frac{\varepsilon}{3}, \frac{\delta}{2(\tau+1)}\right)$
- 4: Set of candidate arms $C = \bigcup_{j=0}^{\tau} (L_j \cup R_j)$
- 5: For each interval $I_i \in \mathcal{I}$, set $D_i = \{x \in C \cap \mathcal{A}(I_i) \mid \hat{\mu}_x < \hat{\mu}_y \text{ for some } y \in C \cap \mathcal{A}(I_i)\}$ and $T_i = (C \cap \mathcal{A}(I_i)) \setminus D_i$ {With respect to the estimates, D_i is the set of Pareto dominated arms}
- 6: **return** $\{T_1, \dots, T_q\}$

their weight estimates. Finally, for each input interval I_i , the algorithm considers the set of candidate arms within I_i and returns the ones that form a Pareto optimal set with respect to the estimates.

The following theorem asserts that ALG-D-RS achieves the desired PAC guarantee in a sample-efficient manner.

Theorem 3. *Given a problem instance $(\mathcal{P}, \mathcal{I})$ with n arms, algorithm ALG-D-RS draws $O\left(\frac{nd}{\varepsilon^2} \log\left(\frac{\tau d}{\varepsilon \delta}\right)\right)$ samples and achieves the (ε, δ) -PAC guarantee for bandit range searching with d -dimensional weights; here τ denotes the size of the minimum cardinality hitting set for \mathcal{I} .*

Proof Sketch. In the interest of space, we defer the technical details to the full version of this paper and provide a proof sketch here. For any slab $S_j = [e_j, e_{j+1}]$, the subroutines $\text{D-LSKY}\left(S_j, \frac{\varepsilon}{3}, \frac{\delta}{2(\tau+1)}\right)$ and $\text{D-RSKY}\left(S_j, \frac{\varepsilon}{3}, \frac{\delta}{2(\tau+1)}\right)$ require $O\left(\frac{n_j d}{\varepsilon^2} \log\left(\frac{2d(\tau+1)}{\varepsilon \delta}\right)\right)$ samples each; here, $n_j := \mathcal{A}(S_j)$. Note that $\sum_{j=0}^{\tau} n_j \leq 2n$; this inequality follows from the fact that any arm can be a part of at most two successive slabs. Now, summing over the sample complexities of the subroutines, across all the slabs, we get that $O\left(\frac{nd}{\varepsilon^2} \log\left(\frac{\tau d}{\varepsilon \delta}\right)\right)$ samples are drawn in the algorithm.

Next, we complete the proof by showing that the ALG-D-RS achieves the (ε, δ) -PAC guarantee. Write \mathcal{E} to denote the event that, for all the slabs, the subroutines find the desired set of arms along with accurate-enough estimates. Given that the success probability of each subroutine is at least $\left(1 - \frac{\delta}{2(\tau+1)}\right)$ and there are $2(\tau+1)$ subroutine instantiations, we get (via the union bound) that $\mathbb{P}\{\mathcal{E}\} \geq 1 - \delta$.

We will prove that, under event \mathcal{E} , for every input interval $I_i \in \mathcal{I}$, the set of arms $T_i \subseteq \mathcal{A}(I_i)$ (selected in Line 5) satisfies both part (a) and (b) of Definition 3, i.e., T_i is ε -Pareto

optimal for I_i . Hence, the desired PAC guarantee holds.

Fix an interval $I_i \in \mathcal{I}$ and consider any arm $b \in \mathcal{A}(I_i)$. We will first note that there necessarily exists an arm $a \in T_i$ that satisfies condition (a) (of Definition 3) for arm b . Towards this, note that Property (P) of our slab construction (see Section 2) ensures that interval I_i is partitioned among two or more slabs, S_x, S_{x+1}, \dots, S_y ; in particular, $x < y$. We will perform a case analysis based on whether arm b is contained in S_x and one of the intermediate slabs, or in S_y .

Case I: $b \in \mathcal{A}(S_t)$ with $x \leq t < y$. Here, under event \mathcal{E} , the set of arms L_t (obtained by executing D-LSKY on S_t) is an $(\varepsilon/3)$ -left-skyline. Therefore, using condition (i) in Proposition 1, one can show that there exists a candidate arm in $C \supseteq L_t$ and, hence, an arm in T_i that satisfies condition (a) with respect to b . **Case II:** $b \in \mathcal{A}(S_y)$. The analysis here relies on the $(\varepsilon/3)$ -right-skyline, R_y , of slab S_y and is otherwise identical to the previous case.

It only remains to prove that T_i satisfies part (b) of Definition 3. This follows from the observation that T_i retains arms that are *not* Pareto dominated with respect to sufficiently-accurate estimates $\hat{\mu}_a$ s, i.e., the retained arms could not have been Pareto dominated under μ_a s beyond an absolute approximation factor of ε . Therefore, parts (a) and (b) of Definition 3 hold for T_i , proving that T_i is indeed ε -Pareto optimal for I_i . This completes the proof. \square

5 Lower Bound

This section presents a lower bound which shows that our algorithms are essentially optimal in terms of sample complexity. The following result is obtained via a nontrivial reduction from a problem studied in [Cheu *et al.*, 2018]; details of the reduction are deferred to the full version of this paper.

Theorem 4. *For each $\tau \in \mathbb{Z}_+$, there exists a bandit range searching instance, with n arms and interval collection \mathcal{I} , such that the size of optimal hitting set for \mathcal{I} is τ , and any algorithm that achieves the (ε, δ) -PAC guarantee for the instance necessarily draws $\Omega\left(\frac{n}{\varepsilon^2} \log\left(\frac{\tau}{\varepsilon \delta}\right)\right)$ samples. Here, d is the ambient dimension of the weights.*

6 Conclusion and Future Work

This work establishes essentially tight sample-complexity bounds for the problem of identifying optimal arms within a given collection of intervals. Complementing the current focus on sample-efficient algorithms, one can also consider a *regret* version of the range searching.

Extending the range aspects to higher dimensions (e.g., finding optimal arms within rectangles) would also be interesting. More generally, the interplay of computational geometry and multi-armed bandits stands as a rich source of geometric problems over uncertain data.

Acknowledgements

Siddharth Barman gratefully acknowledges the support of a Ramanujan Fellowship (SERB - SB/S2/RJN-128/2015). Saladi Rahul's research is generously supported by an IISc Startup Grant.

References

- [Agarwal and Erickson, 1999] Pankaj K. Agarwal and Jeff Erickson. Geometric range searching and its relatives. *Advances in Discrete and Computational Geometry*, pages 1–56, 1999.
- [Agarwal *et al.*, 2003] Pankaj K. Agarwal, Lars Arge, and Jeff Erickson. Indexing moving points. *Journal of Computer and System Sciences (JCSS)*, 66(1):207–243, 2003.
- [Agarwal *et al.*, 2009] Pankaj K. Agarwal, Siu-Wing Cheng, Yufei Tao, and Ke Yi. Indexing uncertain data. pages 137–146, 2009.
- [Agarwal *et al.*, 2018] Pankaj K. Agarwal, Nirman Kumar, Stavros Sintos, and Subhash Suri. Range-max queries on uncertain data. *Journal of Computer and System Sciences (JCSS)*, 94:118–134, 2018.
- [Agarwal, 2017] Pankaj K. Agarwal. Simplex range searching and its variants: A review. *A Journey Through Discrete Mathematics: A Tribute to Jiří Matoušek*, pages 1–30, 2017.
- [Audibert and Bubeck, 2010] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. 2010.
- [Auer *et al.*, 2016] Peter Auer, Chao-Kai Chiang, Ronald Ortner, and Madalina Drugan. Pareto front identification from stochastic bandit feedback. In *Artificial intelligence and statistics*, pages 939–947, 2016.
- [Bubeck *et al.*, 2013] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, pages 258–265, 2013.
- [Cheu *et al.*, 2018] Albert Cheu, Ravi Sundaram, and Jonathan Ullman. Skyline identification in multi-arm bandits. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 1006–1010. IEEE, 2018.
- [Domingo *et al.*, 2002] Carlos Domingo, Ricard Gavaldà, and Osamu Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery*, 6(2):131–152, 2002.
- [Even-Dar *et al.*, 2006] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.
- [Garivier and Kaufmann, 2016] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027, 2016.
- [Gupta *et al.*, 2018] Prosenjit Gupta, Ravi Janardan, Saladi Rahul, and Michiel Smid. Computational geometry: Generalized (or colored) intersection searching. In *Handbook of Data Structures and Applications*, chapter 67:1042–1057, 2018.
- [Kalyanakrishnan and Stone, 2010] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, volume 10, pages 511–518, 2010.
- [Li and Wang, 2016] Jian Li and Haitao Wang. Range queries on uncertain data. *Theoretical Computer Science*, 609:32–48, 2016.
- [Li *et al.*, 2003] Xin Li, Young-Jin Kim, Ramesh Govindan, and Wei Hong. Multi-dimensional range queries in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 63–75. ACM, 2003.
- [Mannor and Tsitsiklis, 2004] Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.
- [Rahul and Tao, 2019] Saladi Rahul and Yufei Tao. A guide to designing top-k indexes. *SIGMOD Record*, 48(2):6–17, 2019.
- [Russo, 2016] Daniel Russo. Simple bayesian algorithms for best arm identification. In *Conference on Learning Theory*, pages 1417–1418, 2016.
- [Samet, 1990] Hanan Samet. *Applications of spatial data structures - computer graphics, image processing, and GIS*. Addison-Wesley, 1990.
- [Slivkins, 2019] Aleksandrs Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.
- [Tao *et al.*, 2007] Yufei Tao, Xiaokui Xiao, and Reynold Cheng. Range search on multidimensional uncertain data. *ACM Transactions on Database Systems (TODS)*, 32(3):15, 2007.
- [Yu *et al.*, 2018] Xiaotian Yu, Han Shao, Michael R Lyu, and Irwin King. Pure exploration of multi-armed bandits with heavy-tailed payoffs. In *UAI*, pages 937–946, 2018.