Journal of Machine Learning Research 24 (2023) 1-29 Submitted 5/22; Revised 12/22; Published 4/23

Adaptive Data Depth via Multi-Armed Bandits

Tavor Z. Baharav TAVORB@STANFORD.EDU
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

Tze Leung Lai LAIT@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Editor: Chris Wiggins

Abstract

Data depth, introduced by Tukey (1975), is an important tool in data science, robust statistics, and computational geometry. One chief barrier to its broader practical utility is that many common measures of depth are computationally intensive, requiring on the order of n^d operations to exactly compute the depth of a single point within a data set of n points in d-dimensional space. Often however, we are not directly interested in the absolute depths of the points, but rather in their *relative ordering*. For example, we may want to find the most central point in a data set (a generalized median), or to identify and remove all outliers (points on the fringe of the data set with low depth). With this observation, we develop a novel instance-adaptive algorithm for adaptive data depth computation by reducing the problem of exactly computing n depths to an n-armed stochastic multi-armed bandit problem which we can efficiently solve. We focus our exposition on simplicial depth, developed by Liu (1990), which has emerged as a promising notion of depth due to its interpretability and asymptotic properties. We provide general data-dependent theoretical guarantees for our proposed algorithms, which readily extend to many other common measures of data depth including majority depth, Oja depth, and likelihood depth. When specialized to the case where the gaps in the data follow a power law distribution with parameter $\alpha < 2$, we reduce the complexity of identifying the deepest point in the data set (the simplicial median) from $O(n^d)$ to $\tilde{O}(n^{d-(d-1)\alpha/2})$, where $\tilde{O}$ suppresses a logarithmic factor. We corroborate our theoretical results with numerical experiments on synthetic data, showing the practical utility of our proposed methods.

Keywords: multi-armed bandits, data depth, adaptive computation, simplicial depth

1. Introduction and Background

For a set $\mathcal{D}$ of n points in d-dimensional space, the simplicial depth of a point x is defined as the fraction of closed d simplices with vertices in $\mathcal{D}$ which contain x. This empirical measure of depth was formulated by Liu (1990), and satisfies many desired properties of depth including affine invariance, and in the continuous limit satisfies maximality in the center and monotonicity relative to the deepest point. This leads to its usage in robust statistics (Maronna et al., 2019), computational geometry (Becker et al., 1987), and exploratory data science (Wellmann and Müller, 2010). Dating back to its inception, simplicial depth has been utilized as a metric for detecting outliers in a multivariate data cloud (Liu, 1990). The determination and filtering of outliers is known to be a difficult problem, where many visual or ad-hoc methods are used: data-depth provides a well-defined measure of outlyingness. Data depth can also be used in hypothesis testing frameworks (Rousseeuw and Struyf, 2002), where we wish to test whether a certain subset of data (e.g. cell-type) is more extreme (shallow) than would be expected by chance. Further discussion regarding the applications of data depth to constructing confidence regions and performing other statistical tests can be found in Liu et al. (2006).

One common object of interest, particularly in data science settings, is the simplicial median (Gil et al., 1992). The simplicial median is defined as the point within a data set with the largest simplicial depth, which is equivalent to the traditional median for 1-dimensional data. The simplicial median satisfies several natural desiderata when extended to high-dimensional data as discussed in Liu (1990), which has led to its use in multivariate data analysis (Rousseeuw and Hubert, 2017; Schervish, 1987). The simplicial median falls within a data set, unlike a mean, and so has meaning even if the data is sparse in some domain, as is common in single cell RNA-seq data sets (Ntranos et al., 2016). Additionally, the simplicial median has some level of robustness, with a data dependent breakdown value of up to $1/(d+2)$, and hence is more robust to outliers and adversarial perturbations than objects like the mean (Rousseeuw and Hubert, 2017).

The rest of the paper is organized as follows: in the remainder of this section we discuss related works, providing background on data depth, multi-armed bandits, and the use of adaptivity as a computational tool. We formalize the problem setting in Section 2, summarizing our results in Table 1. We construct and provide theoretical guarantees for our adaptive algorithm for simplicial median computation in Section 3, which we generalize to other adaptivity amenable tasks in Sections 3.2 and 3.3, providing our meta algorithm for adaptive depth computation in Algorithm 3. Numerical experiments are provided in Section 4 to corroborate our theoretical results. In Section 5 we discuss two computational and theoretical extensions for our proposed algorithms. We conclude in Section 6. Proofs can be found in Appendix A, and additional experimental details are provided in Appendix B. Our code is publicly available on Github at `https://github.com/TavorB/adaSimplicialDepth` to enable the broader use of Simplicial Depth in data-scientific applications, as well as to reproduce the empirical results in this paper.

1.1 Related Work on Data Depth

The mathematical formalization of data depth dates back to Tukey's halfspace median (Tukey, 1975). Due to the lack of efficient algorithms for its computation, many other notions have been proposed, including Oja depth (Oja, 1983), Peel depth (Barnett, 1976), and simplicial depth (Liu, 1990), each with their own strengths and weaknesses (Gil et al., 1992). In this work we focus on simplicial depth, which was introduced by Liu (1990), receiving further treatment by Liu et al. (1999). Additional proposed improvements regarding the finite-sample characterization of simplicial depth were proposed by Burr et al. (2006), defining a modified simplicial depth as the average of the fraction of open and closed simplices the point is contained within. Naively computing the simplicial depth of a query point among n points in d-dimensional space requires $O(n^{d+1})$ time as it entails enumerating all possible simplices, but considerable effort has been devoted to developing more efficient geometric methods. State-of-the-art algorithms require $O(n \log n)$ time for $d = 2$ (Rousseeuw and Ruts, 1996), $O(n^2)$ time for $d = 3$ (Cheng and Ouyang, 2001), and $O(n^{d-1})$ for $d \geq 3$ (Pilz et al., 2020). A general lower bound of $\Omega(n^{d-1})$ has been conjectured for exactly computing the simplicial depth of a single point, and while this has been shown to be achievable up to logarithmic factors, this too quickly becomes infeasible for even moderate n. One approach to make this task computationally tractable is to relax the problem and instead only require a $(1 + \varepsilon)$ approximation of a points simplicial depth. In this case, it was recently shown that the simplicial depth of a given query point could be approximated in $\tilde{O}(n^{d/2+1})$ time for constant $\varepsilon > 0$ (Afshani et al., 2015). Additive approximations can be achieved by utilizing ε-nets and ε-approximations (Bagchi et al., 2007), however such approaches yield poor performance as $\varepsilon \to 0$.

As defined by Liu (1990), simplicial depth refers to the depth of a point x with respect to a distribution f, and sample simplicial depth refers to the depth of a point x with respect to a data set $\mathcal{D}$ (i.e., with respect to the empirical distribution of $\mathcal{D}$). This is based on the setting where $\mathcal{D} = \{x_1, \ldots, x_n\}$, and each x_i is independently drawn from a common distribution f. Then, the sample simplicial depth of a point x with respect to $\mathcal{D}$ is a sample estimate for the simplicial depth of x with respect to the distribution f. For readability we omit the qualifier "sample" when it is clear from context.

We now focus our attention on the computation of the simplicial median, the point in the data set with the deepest simplicial depth (uniqueness issues discussed later). This was tackled by Gil et al. (1992), where a scheme was proposed for $d = 2$ which utilizes an efficient simplicial depth algorithm similar to that of Rousseeuw and Ruts (1996) to compute the simplicial depth of each point exactly, requiring $O(n^2 \log n)$ time. In the same work the authors propose an algorithm for $d = 3$, later corrected by Cheng and Ouyang (2001), which computes the simplicial median in $O(n^3)$ time in a similar manner. Both of these schemes rely on an efficient algorithm for computing the simplicial depth of a single point, and loop over each point to compute its exact depth using this fast algorithm. Thus, finding the simplicial median degenerates to n independent simplicial depth computations. Indeed, it has been conjectured that these algorithms are optimal for the computation of the simplicial median if all the simplicial depths are required to be computed exactly (Gil et al., 1992). While in a minimax sense this may be the case, practical datasets are generated by nature, not by an adversary. Our proposed algorithm is able to provide data-dependent complexity guarantees, where for easier problem instances we only coarsely approximate the simplicial depths of many points, yielding a dramatically improved sample complexity (while retaining worst-case guarantees). We mention that other works have tackled more general depth approximation problems, where a scheme for determining a point with deepest simplicial depth over all of $\mathbb{R}^d$ up to a multiplicative $(1 + \varepsilon)$ was proposed by Aronov and Har-Peled (2008). Their work optimizes over all of $\mathbb{R}^d$ rather than simply within the data set however, leading to a more general problem and a computational complexity of $\Omega(n^{d+1})$ for their algorithm in this setting.

On a more practical note, no algorithms for $d > 2$ are implemented and empirically tested in the literature (approximation or otherwise). These algorithms rely on complex combinatorial subroutines, and many only yield asymptotic error probability guarantees (for a fixed ε) with unspecified or unwieldy constants. This mitigates their utility in real world applications, where the user wishes to specify both an allowable error tolerance ε (potentially 0), as well as a maximum error probability δ. This motivates our efficient multi-armed bandit-based algorithm, which is able to satisfy these desiderata in a theoretically sound, computationally efficient, and modular algorithm.

1.2 Adaptivity as a Computational Tool

Due to the ever increasing size of data sets, data-adaptive randomized algorithms have been recognized as a useful tool for constructing instance-optimal algorithms for data science primitives. From its early uses in Monte Carlo Tree Search (Kocsis and Szepesvári, 2006) to more recent utilization in hyper-parameter tuning (Li et al., 2017), adaptivity has been used to focus computational resources on relevant portions of computational tasks. Recently formalized under the general framework of Bandit-Based Monte Carlo Optimization (BMO) (Bagaria et al., 2021), this technique has been utilized to solve many problems, including finding the medoid of a data set (Bagaria et al., 2018; Baharav and Tse, 2019; Tiwari et al., 2020), k-nearest neighbor graph construction (Bagaria et al., 2021; LeJeune et al., 2019; Mason et al., 2019, 2021), Monte Carlo permutation-based multiple testing (Zhang et al., 2019), mode estimation (Singhal et al., 2021), and a rank-one estimation problem (Kamath et al., 2020). This general framework provides efficient algorithms for solving problems of the form $f(\theta_1, \ldots, \theta_n)$, where each θ_i is an expensive to compute but easy to estimate quantity, and f is a function that does not depend too strongly on too many of the θ_i. This algorithm design philosophy is discussed in more detail in the recent work of Bagaria et al. (2021), and analyzed in the case when f is real-valued and smooth by Baharav et al. (2022). The resulting adaptive algorithms provide a natural and efficient way to solve certain structured large scale computational problems, allowing for dramatic improvements in computational complexity. To realize these theoretical gains as wall-clock improvements, the incorporation of round-efficient multi-armed bandit algorithms is critical. This allows for batched computations, yielding up to 4-5 orders of magnitude speed ups in wall-clock time (Baharav and Tse, 2019).

In this paper we construct efficient algorithms for simplicial median computation and several related data depth tasks by building on and extending the BMO framework (Bagaria et al., 2021). First, in this work we show the utility of judicious use of domain-specific computation methods. While random sampling for θ_i estimation, as in the BMO framework, can offer statistical benefits, it is often inefficient for achieving high accuracy estimates. To this end, we show how to algorithmically incorporate these improved exact or high-precision computational methods, and the dramatic gains they can provide. Second, we highlight the importance and practical necessity of batched computation. In statistical tasks, adaptivity (which is inherently serial) is critical for statistical efficiency. In computational tasks however, parallelism is extremely important for computational efficiency, due to the structure of modern hardware and efficiency of BLAS. In this work, we showed that we can build off of bandit algorithms that achieve order optimal statistical performance (up to logarithmic factors), and use only a number of rounds scaling logarithmically in problem parameters, an exponential improvement. Finally, due to the combinatorial nature of simplicial depth (averaging over binary indicators), in this work we do not require *any* assumptions on the data for our algorithm to work. In the original BMO setting, observations could be arbitrarily large, e.g. distances between two points along a coordinate in the case of nearest neighbor computation. This required the algorithm to take as input a bound on the sub-Gaussianity of the distances. In the setting of simplicial depth, however, no assumptions are required to provide concentration inequalities for the sums of $\{0,1\}$-valued random variables that arise.

1.3 Related Work on Multi-Armed Bandits

Multi-armed bandits, the framework upon which we build our adaptive data depth algorithm, have a long history in the statistics community. The k-armed bandit problem was introduced by Robbins (1952) for $k = 2$ in his seminal paper on sequential design of experiments, in which he considered sequential sampling from two populations with unknown means to maximize the total expected rewards $\mathbb{E}[y_1 + \ldots + y_n]$, where y_i has mean μ_1 (or μ_2) if it is sampled from population 1 (or 2) and n is the total sample size. Letting $s_n = y_1 + \ldots + y_n$, he applied the law of large numbers to show that $\lim_{n\to\infty} n^{-1}\mathbb{E}[s_n] = \max(\mu_1, \mu_2)$ is attained by the following rule: sample from the population with the larger sample mean except at times belonging to a designated sparse set T_n of times, and sample from the population with the smaller size at the designated times. T_n is called "sparse" if $\#(T_n) \to \infty$ but $\#(T_n)/n \to 0$ as $n \to \infty$, where $\#(\cdot)$ denotes the cardinality of a set.

In 1957, Bellman introduced the dynamic programming approach to Robbins' 2-armed sequential sampling problem, generalizing it to k-arms and calling it a "k-armed bandit problem" (Bellman, 1957). The name derives from an imagined slot machine with k arms (levers) such that when arm j is pulled the player wins a reward generated by an unknown probability distribution Π_j. In this setting where we seek to maximize the expected sum of rewards there is a fundamental dilemma between "exploration" (to obtain information about $\Pi_1, \ldots, \Pi_k$ by pulling the individual arms) and "exploitation" (of the information so that inferior arms are pulled minimally). Dynamic programming offers a systematic solution of the dilemma in the Bayesian setting but suffers from the curse of dimensionality as k and n increase, as recognized by Bellman.

Extending these regret analyses beyond the standard stochastic multi-armed bandit setting we consider in this work, Lai and Yakowitz (1995) were able to use large deviation bounds to provide a regret lower bound in the nonparametric setting pioneered by Yakowitz and Lowe (1991), also providing a UCB (upper confidence bound) based adaptive allocation rule. Using a different method, Auer et al. (2002) developed an ε-greedy–based procedure similar to that used in reinforcement learning (Sutton and Barto, 2018), which they showed was also able to attain the logarithmic regret lower bound.

These ideas have been further developed in the contextual setting, where Kim et al. (2021) generalize the definition of regret to the nonparametric setting with stochastic covariates $\{\boldsymbol{x}_t\}_{t=1}^n$ which are independent and identically distributed. There, the authors construct an arm elimination based k-armed adaptive allocation procedure using ε-greedy randomization and a Welch-type test statistic, which generalizes parametric likelihood ratio statistics to the nonparametric setting. Kim et al. (2021) show that that under weak regularity conditions their procedure attains the asymptotic minimax rate of the risk functions for adaptive allocation rules. The authors additionally discuss their advances in the context of reinforcement learning for recommender systems and personalization technologies. This motivates a practically important extension of the contextual setting to the scenario where the number of arms scales with the time horizon and so $k = k_n$ (instead of fixing k and letting n tend to infinity), in which case k_n can exceed n, which has been definitively settled in the recent work of Lai et al. (2022) who consider a non-denumerable set of arms. This recent advance utilizes the decoupling inequalities of de la Peña and Lai (2000) to handle general dependence structure (unknown) among the covariates. Related applications of this breakthrough have been given by Sklar et al. (2021) and Lai et al. (2021). We direct the interested reader to these works, in particular Lai et al. (2022) and Kim et al. (2021), for a more thorough discussion of regret minimization techniques, a broader background regarding the multi-armed bandit literature, and a thorough characterization of the nonparametric contextual multi-armed bandit setting.

Since these foundational works, multi-armed bandits have proven to be an extremely broad and useful model for dealing with decision making under uncertainty, having been utilized in the design of adaptive clinical trials (Villar et al., 2015), online ad recommendation (Lu et al., 2010), multi-agent learning (Bistritz et al., 2021), and many more applications (Slivkins, 2019; Bubeck and Cesa-Bianchi, 2012). Most relevant to this work on data depth are the pure exploration variants of the multi-armed bandit problem, where the prototypical objective is to find the best arm (Jamieson and Nowak, 2014), which in this case translates to finding the point with the deepest simplicial depth. Common algorithmic paradigms for this setting involve either index-based policies similar to the original Upper Confidence Bound (UCB) algorithm (Lai and Robbins, 1985), or elimination-based algorithms where arms are successively pulled and discarded after they are determined to be suboptimal with high probability (Audibert et al., 2010). The hardness of the best arm identification problem has been precisely characterized through information-theoretic lower bounds (Kaufmann et al., 2016), asymptotically optimal algorithms (Garivier and Kaufmann, 2016), and algorithms that are within $\log\log$ factors of optimal and provide finite finite-sample guarantees (Jamieson et al., 2014).

There are many pure exploration problems beyond best arm identification where adaptivity is beneficial. The most natural extension is to the PAC scenario, where the objective is to identify an arm with mean within ε of the best arm (Kalyanakrishnan et al., 2012). Another extension is to the top-k setting, where the objective is to identify the k arms with largest means (Simchowitz et al., 2017). If a specific value k is not desired, one can instead find all arms with means within an additive or multiplicative ε of the best arm (Mason et al., 2020). Another approach would be to coarsely rank the arms into batches (Katariya et al., 2018; Karpov and Zhang, 2020). Alternatively, one may wish to identify all arms with means above a given input threshold, in what is called the thresholding bandit problem (Locatelli et al., 2016). A final problem could be determining one arm with mean above a threshold in the good arm identification problem (Katz-Samuels and Jamieson, 2020). While many of these objectives could be of interest in the context of data depth, in this work we focus on the task of identifying the simplicial median, corresponding to the best-arm identification problem, providing in Algorithm 3 a general framework.

2. Problem Formulation

For a data set $\mathcal{D} = \{x_i\}_{i=1}^n$ of n points in d-dimensional space, we defined simplicial depth in words as the fraction of closed d simplices with vertices in $\mathcal{D}$ which contain the target point. Denoting the convex hull (simplex) of points $x_1, \ldots, x_{d+1}$ as $S[x_1, \ldots, x_{d+1}]$, we are able to mathematically define the sample simplicial depth below.

Definition 1 *The sample simplicial depth of a point x with respect to a data set $x_1, \ldots, x_n$ is defined as*

$$D_n(x) \triangleq \binom{n}{d+1}^{-1} \sum_{1 \le i_1 < \ldots < i_{d+1} \le n} \mathbb{1}\left\{x \in S[x_{i_1}, \ldots, x_{i_{d+1}}]\right\}. \tag{1}$$

As discussed, one natural object of interest in data science applications is the simplicial median, the point in the data set with the largest simplicial depth. This is formalized as a point in the set

$$\operatorname*{argmax}_{x_i \in \mathcal{D}} D_n(x_i), \tag{2}$$

which we assume to be unique for the rest of this work for simplicity of presentation. Our proposed algorithms can accommodate standard solutions to lack of uniqueness such as defining the simplicial median as the average of all points contained in the argmax of (2) as is done by Rousseeuw and Hubert (2017).

For the computation of (1), determining whether a point x is contained within a given simplex $S[x_{i_1}, \ldots, x_{i_{d+1}}]$ is equivalent to verifying whether x can be expressed as a linear combination with nonnegative coefficients of $\{x_{i_k}\}_{k=1}^{d+1}$. These coefficients are the (normalized) barycentric coordinates of x with respect to $\{x_{i_k}\}$. Computing these barycentric coordinates entails solving a d-dimensional linear system, which can be efficiently performed in $O(d^3)$ time for a given simplex. The difficulty in computing (1) arises from the $\binom{n}{d+1}$ simplices that naively need to be enumerated to compute the exact depth of a point. We provide further discussion on this computation and on its batched efficiency in Section 5.2.

2.1 Results and Contributions

Our objective in this paper is to construct an algorithm that efficiently computes the simplicial median as defined in (2) with error probability at most δ. We provide in Table 1 a summary of the results in this paper, where $\tilde{O}$ hides logarithmic factors in n and the gaps $\{\Delta_i\}$, formally defined later.

Unlike prior works which utilize the BMO technique (Bagaria et al., 2021), an interesting feature of our algorithms for adaptive simplicial depth computation is that they are able to incorporate in a modular manner an optimized exact computation algorithm. Concretely, many of these previous tasks have θ_i (the difficult to compute but easy to approximate quantities) that are expressible as sums, where Monte Carlo samples are generated by sampling a random term in this sum and exact computation is performed by iterating over the entire sum. In this setting Monte Carlo samples are similarly generated by noting that

$$\mathbb{E}\left[\mathbb{1}\left\{x \in S[x_{i_1}, \ldots, x_{i_{d+1}}]\right\}\right] = D_n(x),$$

for $\{x_{i_j}\}_{j=1}^{d+1}$ drawn uniformly at random from subsets of $\mathcal{D}$ of size $d+1$. In this setting however, exact computation can be performed more efficiently than by simply enumerating

	Gaps	Task	Complexity
Theorem 1	arbitrary	simplicial median	See Theorem 1
Corollary 1.1	arbitrary	simplicial median	$\tilde{O}\left(\sum_i \min(\Delta_i^{-2}, n^{d-1})\right)$
Corollary 1.2	random, α power law	simplicial median	$\tilde{O}\left(n^{d-(d-1)\alpha/2}\right)$
Theorem 2	arbitrary	deepest-k	$\tilde{O}\left(\sum_i \min\left(\left(\Delta_i^{(k)}\right)^{-2}, n^{d-1}\right)\right)$
Theorem 3 Corollary 3.1	arbitrary	general	See Theorem 3

Table 1: Summary of the results provided in this paper for dynamic data depth computation for $d \geq 1$. Complexity column only displays coarsely bounded results for ease of readability, see formal statements for tighter results. All sample complexities scale linearly in $\log(1/\delta)$.

all $\binom{n}{d+1} = \Theta(n^{d+1})$ simplices (terms in the sum) to exactly compute the simplicial depth of a point; we know from the works of Rousseeuw and Ruts (1996) and Pilz et al. (2020) that algorithms for exact computation exist that require only $\tilde{O}(n^{d-1})$ time. This enables us to use a reduced threshold for exact computation, the number of Monte Carlo samples beyond which it is determined to be more efficient to exactly compute the depth of a given point than to obtain further Monte Carlo samples to approximate it to greater accuracy. This extends and highlights the modularity of the BMO framework (Bagaria et al., 2021); better approximation methods and exact computation schemes can be seamlessly integrated into a meta algorithm (Algorithm 3), yielding improved overall performance.

3. Proposed Algorithm

In this section we formulate and discuss Algorithm 1, which efficiently finds the simplicial median of an input data set $\mathcal{D}$ with failure probability at most δ. This algorithm utilizes the multi-armed bandit algorithm `Multi-Round` $\varepsilon-$`Arm` (Algorithm 3 of Hillel et al. (2013)). We proceed in rounds, where in round r the goal is to eliminate all $\varepsilon_r = 2^{-r}$ suboptimal points. This is achieved by constructing t_r total random simplices in round r such that we are able to estimate each remaining point's simplicial depth to within an additive $\varepsilon_r/2$ with high probability. A point's depth is iteratively approximated to greater and greater accuracy until it is either eliminated, or the cost of approximating its depth to accuracy $\varepsilon_r/2$ is greater than the cost of exactly computing its simplicial depth, `eCost`, as in Line 6. Note that in round r this cost `eCost` should be compared with the cost of performing t_r barycentric coordinate computations, as discussed in greater detail in Section 5.2. This allows for potentially dramatic computational gains on an instance by instance basis, as the simplicial depth of each point is only approximated to the necessary accuracy for the task at hand (determining if it is the deepest point or not).

3.1 Theoretical Guarantees

We are able to provide sample complexity bounds regarding the performance of Algorithm 1 by showing that every suboptimal point will be eliminated after an appropriate number of rounds (after its depth has been approximated to the necessary accuracy). For notational

Algorithm 1 Adaptive Simplicial Median

```
Input: data set $\mathcal{D} = x_1, \ldots, x_n \in \mathbb{R}^d$, error probability $\delta$, exact computation method eDepth with runtime eCost
Initialize: $S_1 \leftarrow [n]$, $r = 0$, $t_0 = 0$
repeat
    $r \leftarrow r + 1$
    Set $\varepsilon_r = 2^{-r}, t_r = \lceil 2\varepsilon_r^{-2} \log(4nr^2/\delta) \rceil$
    if $t_r >$ eCost then                              ▷ Use specialized exact method
        $\hat{\mu}_i^r \leftarrow$ eDepth$(x_i, \mathcal{D})$ for $i \in S_r$
        return argmax$_{i \in S_r} \hat{\mu}_i^r$
    end if
    Select $X_{t_{r-1}+1}, \ldots, X_{t_r}$ independently and u.a.r. as subsets of size $d+1$ of $\mathcal{D}$
    For $i \in S_r$, $j \in [t_{r-1}+1, t_r]$, compute $Y_{i,j} = \mathbb{1}\{x_i \in \Delta(X_j)\}$
    For $i \in S_r$ construct $\hat{\mu}_i^r = \frac{1}{t_r} \sum_{j=1}^{t_r} Y_{i,j}$
    Set $S_{r+1} \leftarrow S_r \setminus \{i \in S_r : \hat{\mu}_i^r < \max_{j \in S_r} \hat{\mu}_j^r - \varepsilon_r\}$
until $|S_{r+1}| = 1$
return Point in $S_{r+1}$
```

simplicity we assume that the points are sorted in order of simplicial depth and so x_1 is the simplicial median, but that the algorithm does not know this. We define $\mu_i = D_n(x_i)$ and $\Delta_i = \mu_1 - \mu_i$, with the understanding that if $\Delta_i = 0$ for some i, then $\Delta_i^{-2} = +\infty$. We additionally define $[n] \triangleq \{1, 2, \ldots, n\}$, $[m, n] \triangleq \{m, m+1, \ldots, n\}$, and denote selection uniformly at random as u.a.r. All logarithms are base e unless stated otherwise. With this definition in hand, we are able to state the following Theorem regarding the number of barycentric coordinate computations made by Algorithm 1 (computations of whether a point is inside a d-dimensional simplex).

Theorem 1 (Main Result) *Algorithm 1 succeeds with probability at least $1 - \delta$ in returning the simplicial median of an input data set $\mathcal{D}$, requiring computation at most*

$$n + \sum_{i=1}^{n} \min\left(\frac{32 \log\left(\frac{16n}{\delta} \log_2^2\left(\frac{2}{\Delta_i}\right)\right)}{\Delta_i^2}, 2 \times \mathtt{eCost} \right)$$

where eCost *is the cost of the input exact computation method. On this δ probability error event, the algorithm will still terminate within $2n \times$* eCost *time.*

We analyze the requisite sample complexity of our adaptive algorithm following standard multi-armed bandit techniques. We begin by showing that each depth estimator $\hat{\mu}_i^r$ is within $\varepsilon_r/2$ of the true simplicial depth of point x_i, μ_i, for all i, r simultaneously with probability at least $1 - \delta$. We then argue that the simplicial median cannot be eliminated on this good event, and that all suboptimal points will be efficiently eliminated. The proof of this theorem can be found in Appendix A.

We note that the factor of 2 in front of the eCost term can be made arbitrarily close to 1 by lowering the threshold for exact computation. This improves the leading constant of the worst case sample complexity of our algorithm but reduces its adaptivity, hampering its instance dependent performance. For example, if the exact simplicial depth of a point is computed if $t_r \geq c \times \texttt{eCost}$ for $c \in (0,1]$ in Line 6 in Algorithm 1, then the computation done for a point is at most $(1+c) \times \texttt{eCost}$. The gap dependent term will naturally suffer however, as points with larger gaps will now also be exactly computed. Up to logarithmic factors, all point with $\Delta_i^{-2} > c \times \texttt{eCost}$ will have their means exactly computed.

Using the modularity of our proposed algorithms and incorporating the optimized geometric exact computation methods developed by Rousseeuw and Ruts (1996) and Cheng and Ouyang (2001), we have the following corollary.

Corollary 1.1 *Algorithm 1 succeeds with probability at least* $1-\delta$ *in returning the simplicial median of the data set* $\mathcal{D}$*, on this event requiring time at most*

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

for $d=2$ *by utilizing the exact computation scheme in (Rousseeuw and Ruts, 1996), and*

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

for $d \geq 3$ *by utilizing the exact computation scheme in (Pilz et al., 2020).*

These results can be extended to the additive and multiplicative approximation settings, where a point $x_i \in \mathcal{D}$ is desired such that $D_n(x_i) \geq \max_{j\in[n]} D_n(x_j) - \varepsilon$ (respectively $D_n(x_i) \geq (1-\varepsilon)\max_{j\in[n]} D_n(x_j)$), by modifying Algorithm 1's termination condition.

Examining the requisite sample complexity of Algorithm 1, Theorem 1 provides a general guarantee for all data sets with data dependent computational complexity, where all constants are explicit. To compare this computational complexity with existing algorithms (which have a sample complexity independent of the gaps) we evaluate the guarantee of Theorem 1 when the gaps between the simplicial depths of each point and the simplicial median follow a power law distribution. We show that this is a reasonable assumption via experiments on synthetic data sets in Figure 2. Note that this assumption is on the gaps between the simplicial depths, $\Delta_i = \mu_1 - \mu_i$, not on the points x_i themselves.

Corollary 1.2 *Assume that a data set* $\mathcal{D} = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$ *is randomly generated such that* $\Delta_i \underset{i.i.d.}{\sim} \Delta$*, where* $F(\Delta) = \Delta^\alpha$ *for* $\Delta \in [0,1]$*, with constant* $\alpha \in [0,\infty)$*. Given this data set, Algorithm 1 will with probability at least* $1-\delta$ *identify the simplicial median of the data set, requiring* M *computations on this success event where, taking the expectation with respect to these* Δ_i*,*

$$\mathbb{E}\{M\} \leq \begin{cases} O\left(n\log(nd/\delta)\, \textit{eCost}^{1-\alpha/2}\right), & \text{for } \alpha \in [0,2), \\ O\left(n\log(nd/\delta)\log(\textit{eCost})\right), & \text{for } \alpha = 2, \\ O\left(n\log(nd/\delta)\right), & \text{for } \alpha > 2. \end{cases}$$

The proof of this corollary follows by evaluating the sample complexity in Theorem 1 with respect to the random gaps, and is relegated to Appendix A.2.

This result highlights the dependence of Algorithm 1's sample complexity on the distribution of the gaps. For $\alpha < 2$ the algorithm's runtime is near $\texttt{eCost}^{1-\alpha/2}$ per point in expectation; almost that of the brute force method that exactly computes each simplicial depth for small α. As α increases our per arm cost decreases, as many arms become easier to eliminate, up until the cost becomes independent of α when $\alpha > 2$, as then so few points require exact computation that the dependence on `eCost` vanishes.

3.2 Extension Beyond Simplicial Median Identification

In practice, one may be interested in finding more than just the simplicial median. One broad class of objectives of interest are coarse ranking problems, where given as input $\{m_i\}$ where $0 = m_0 < m_1 < \ldots < m_\ell = n$, we wish to coarsely cluster the points by their simplicial depth. This entails partitioning the n points into ℓ clusters, such that the first cluster contains the $(m_1 - m_0)$ points with deepest depth, the second cluster contains the following $(m_2 - m_1)$ in terms of depth, and so on. Note that this formulation subsumes simplicial median identification ($\ell = 2, m_1 = 1$), k-outlier detection ($\ell = 2, m_1 = n - k$), and sorting the points by simplicial depth ($\ell = n, m_i = i$ for $1 \leq i \leq n$). Round and sample efficient algorithms were recently proposed for the traditional stochastic multi-armed bandit formulation of this problem by Karpov and Zhang (2020). This has ties to earlier work in the simulation optimization literature, where works exploited the fact that the task of ranking points can be significantly cheaper than approximating all of their depths to high accuracy (Ho et al., 1992). At a high level, these coarse ranking algorithms maintain a set of active arms, which initially contains all n arms. They proceed in rounds of doubling accuracy, pulling each active arm uniformly. When an arm has been pulled sufficiently such that its confidence interval does not overlap with the confidence intervals of any arm in an alternative cluster (i.e. it has been uniquely identified as belonging to a certain cluster), it is removed from the active set. For the sake of concreteness, in the remainder of this section we focus on the objective of finding the k points with largest simplicial depth, but these techniques naturally extend to the general coarse ranking setting.

Theorem 2 (Top-k data depth) *Algorithm 2 will succeed with probability at least $1 - \delta$ in returning a set A with $|A| = k$ such that $\{i : \mu_i > \mu_k + \varepsilon\} \subseteq A \subseteq \{i : \mu_i \geq \mu_k - \varepsilon\}$, requiring on this event time at most*

$$O\left(\sum_{i=1}^{n} \min\left(\frac{\log\left(\frac{n}{\delta}\log\left(1/\Delta_i^{(k)}\right)\right)}{\left(\Delta_i^{(k)}\right)^2}, \textit{eCost}\right)\right)$$

where $\Delta_i^{(k)} = \max(\mu_i - \mu_{k+1}, \varepsilon)$ if $i \leq k$ and $\Delta_i^{(k)} = \max(\mu_k - \mu_i, \varepsilon)$ if $i > k$.

The proof of this theorem follows similarly to that of Theorem 1, and is deferred to Appendix A.3 for continuity.

Algorithm 2 `Adaptive top-`k` Data Depth`

```
1: Input: data set $\mathcal{D} = x_1, \ldots, x_n \in \mathbb{R}^d$, error probability $\delta$, target accuracy $\varepsilon$, integer $k$, exact computation method eDepth with runtime eCost
2: Initialize: $S_1 \leftarrow [n]$, $r = 0$, $A \leftarrow \emptyset$
3: repeat
4:     $r \leftarrow r + 1$
5:     Set $\varepsilon_r = 2^{-r}, t_r = \lceil 2\varepsilon_r^{-2} \log(4nr^2/\delta) \rceil$
6:     if $t_r \geq$ eCost then                                ▷ Use specialized exact method
7:         $\hat{\mu}_i^r \leftarrow$ eDepth$(x_i, \mathcal{D})$ for $i \in S_r$
8:         Break: go to line 16
9:     end if
10:    Select $X_{t_{r-1}+1}, \ldots, X_{t_r}$ independently and u.a.r. as subsets of size $d+1$ of $\mathcal{D}$
11:    For $i \in S_r$, $j \in [t_{r-1}+1, t_r]$, compute $Y_{i,j} = \mathbb{1}\{x_i \in \Delta(X_j)\}$
12:    For $i \in S_r$ construct $\hat{\mu}_i^r = \frac{1}{t_r} \sum_{j=1}^{t_r} Y_{i,j}$, let $\hat{\mu}_{\text{thresh}}^r = \hat{\mu}_{(k-|A|)}^r$
13:    Update $A \leftarrow A \cup \{i \in S_r : \hat{\mu}_i^r \geq \hat{\mu}_{\text{thresh}}^r + \varepsilon_r\}$
14:    Set $S_{r+1} \leftarrow S_r \setminus \{i \in S_r : |\hat{\mu}_i^r - \hat{\mu}_{\text{thresh}}^r| \geq \varepsilon_r\}$
15: until $\varepsilon_r \leq \varepsilon/2$ or $|A| = k$
16: if $|A| < k$ then
17:     Add $k - |A|$ top $\hat{\mu}_i^r$ for $i \in S_r$
18: end if
19: return $A$
```

As discussed, similar theoretical guarantees and efficient algorithms can be provided for a broad class of identification and estimation tasks, such as identifying the k points with smallest simplicial depth (outliers) or partitioning the points into sets (deepest 10% of points and shallowest 5% of points). Variable sized outputs can also be efficiently obtained (identifying all points with simplicial depth at least 80% of that of the simplicial median), as discussed in Section 1.3. These same techniques can also be used to estimate L-statistics, efficiently so if the desired linear combination is sparse. We discuss a meta-algorithm to accomplish these tasks in the following subsection.

3.3 Meta Algorithm

At a high level, the previous two proposed algorithms can be abstracted as in Algorithm 3, which encompasses many other common objectives. The goal of this meta algorithm is to adaptively estimate the simplicial depth of each point to only the *necessary* accuracy for the task at hand, allowing it to be more computationally efficient than a naive algorithm which exactly computes the simplicial depth of each point. This is accomplished by proceeding in rounds, where in each round a function `aDepth` is used to approximate the depth of the relevant points x_i to an additive tolerance of ε_r which decreases over the rounds r, with error probability at most $\delta/(2nr^2)$. For approximating simplicial depth, this can naturally be accomplished by randomly subsampling simplices and checking to see if the query point x_i is contained within them, as utilized in Algorithm 1, or by alternative techniques as those of Afshani et al. (2015). At the end of round r, a set of arms E_r is eliminated. This should be thought of as the set of arms for which the algorithm has already received enough information regarding their means, and no longer needs to refine their estimates. In the context of top-k identification, this constitutes the points which have been determined to have too shallow of a simplicial depth so as to with high probability not be in the top-k, and those points which have been determined to be sufficiently deep that they are with high probability in the top-k. For simplicial median computation this decision is one-sided. For coarse-ranking this elimination procedure is necessarily slightly more sophisticated as discussed by Karpov and Zhang (2020).

Algorithm 3 `Adaptive Data Depth Meta Algorithm`

```
Input: data set $\mathcal{D} = x_1, \dots, x_n \in \mathbb{R}^d$, error probability $\delta$, exact computation method
    eDepth with runtime eCost, approximation method aDepth with runtime aCost($\varepsilon, \delta$)
Initialize: $S_1 \leftarrow [n]$, $r = 0$
repeat
    $r \leftarrow r + 1$
    Set $\varepsilon_r = 2^{-r}$
    if aCost($\varepsilon_r/2, \delta/(2nr^2)$) $>$ eCost then            ▷ Use specialized exact method
        Exactly compute $\hat{\mu}_i^r \leftarrow$ eDepth($x_i, \mathcal{D}$) for $i \in S_r$
        Break
    end if
    For $i \in S_r$ compute $\hat{\mu}_i^r \leftarrow$ aDepth($x_i, \mathcal{D}, \varepsilon_r/2, \delta/(2nr^2)$)
    Construct $E_r \subseteq S_r$ as set of arms to eliminate, based on $\{\hat{\mu}_i^r\}$
    Set $S_{r+1} \leftarrow S_r \setminus E_r$
until terminationCondition($\{\hat{\mu}_i^r\}, \varepsilon, \varepsilon_r$)
return Solution as a function of $S_r, \{\hat{\mu}_i^r\}$
```

Extending the analysis of Algorithm 1 to Algorithm 3, we are able to provide the following Theorem regarding the performance of our meta algorithm when specified to simplicial median computation, using general methods for exact and approximate computation `eDepth` and `aDepth` with runtimes `eCost` and `aCost`(ε, δ) respectively.

Theorem 3 (Meta Algorithm) *Algorithm 3 succeeds with probability at least* $1 - \delta$ *in returning a point* $x_i \in \mathcal{D}$ *with simplicial depth within an additive* ε *of the simplicial median's depth, requiring on this event computation at most*

$$\sum_{i=1}^{n}\left[\left(\sum_{r=1}^{r_i-1}\texttt{aCost}\left(2^{-r-1},\delta/(2nr^2)\right)\right)+\min\left(\texttt{aCost}\left(\frac{\Delta_i}{8},\frac{\delta}{2nr_i^2}\right),\texttt{eCost}\right)\right],$$

where $r_i = \min\left(\{r : r \in \mathbb{N}, \texttt{aCost}(2^{-r-1},\delta/(2nr^2)) \geq \texttt{eCost}\} \cup \{\lceil\log(2/\varepsilon)\rceil, \lceil\log_2(2/\Delta_i)\rceil\}\right)$.

To simplify this expression we provide the following corollary, making the assumption that $2 \times$ `aCost`$(\varepsilon, \delta) \leq$ `aCost`$(\varepsilon/2, \delta)$ for all $\varepsilon \leq 1$, and that this function is increasing as δ decreases. Note that this holds for our sampling based approximation scheme, with $4 \times$ `aCost`$(\varepsilon, \delta) =$ `aCost`$(\varepsilon/2, \delta)$.

Corollary 3.1 *Algorithm 3 succeeds with probability at least* $1-\delta$ *in returning the simplicial median of the data set, and assuming that* $2 \times \texttt{aCost}(\varepsilon, \delta) \leq \texttt{aCost}(\varepsilon/2, \delta)$ *and* $\texttt{aCost}$ *is decaying in* δ*, requires on this event computation at most*

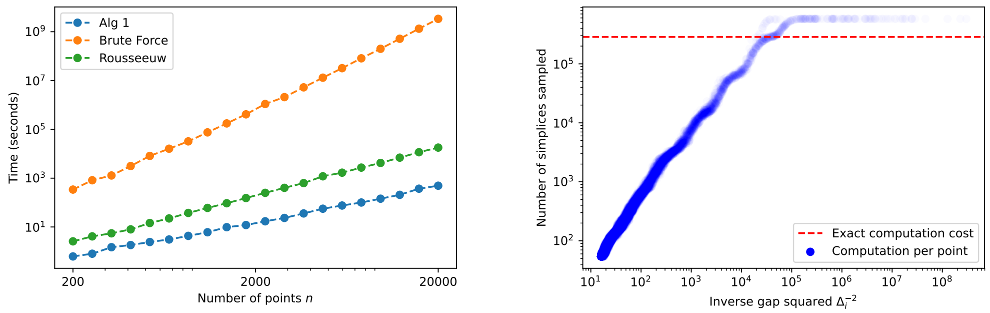
$$\sum_{i=1}^{n} \left[\min\left(2 \times \texttt{aCost}\left(\frac{\Delta_i}{8}, \frac{\delta}{2nr_i^2} \right), 3 \times \texttt{eCost} \right) \right],$$

where $r_i = \min\left(\{r : r \in \mathbb{N}, \texttt{aCost}(2^{-r-1}, \delta/(2nr^2)) \geq \texttt{eCost}\} \cup \{\lceil \log(2/\varepsilon) \rceil, \lceil \log_2(2/\Delta_i) \rceil\} \right)$.

This modularity allows for the incorporation of other approximation algorithms. For example, the high-dimensional approximation scheme proposed by Afshani et al. (2015) could be used (which works for relative approximations and so our meta-algorithm would need to be correspondingly modified). The runtime of this scheme is $\tilde{O}(n^{d/2+1})$, and so one potential scheme could be to use random sampling of simplices to coarsely approximate the simplicial depth of points in early rounds, the more sophisticated geometric approach for approximating the depth of a point to a higher accuracy by Afshani et al. (2015) in later rounds for small ε, and a specialized exact computation method for those points with near maximal simpicial depth.

4. Numerical Simulations

We simulate our adaptive simplicial median computation scheme on synthetic data sets and show the empirical performance improvement afforded by adaptivity. By adaptively approximating the simplicial depth of points in our data set to the necessary accuracy, we are able to obtain significant computational improvements. All experimental results can be reproduced from our publicly available code: `https://github.com/TavorB/adaSimplicialDepth`.



(a)

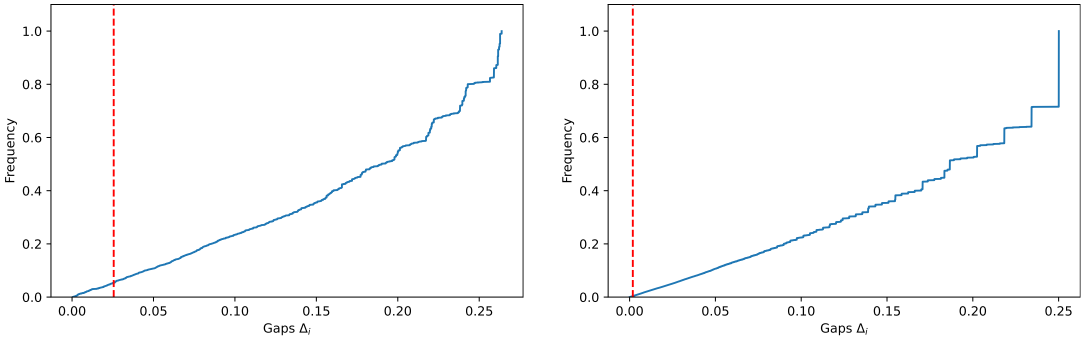
(b)

Figure 1: Simulations for simplicial median computation for $d = 2$, data drawn from isotropic gaussian. a) Wall clock time with 20 simulations per point; 10 random data sets, 2 trials per data set, further details in Appendix B. Adaptive method run with $\delta = 0.001$ returns the correct answer on every run. b) Number of computations made by our adaptive algorithm on a given point as a function of inverse gap squared (in terms of simplicial depth) for $n = 20,000$, averaged over 50 trials on the same randomly generated data set. Plotted in log-log scale to show points with small gaps. $R^2 = .98$ between inverse gap squared x and number of simplices sampled y for points with $y <$ `eCost`, validating that this relationship is indeed nearly linear.

As shown in Figure 1a, our adaptive algorithm dramatically outperforms both exact computation and the geometric but nonadaptive solution of Rousseeuw and Ruts, yielding superior scaling with n. Plotting the runtime of these different methods in log-log scale, we observe that the brute force algorithm has a wall clock time scaling theoretically as $O(n^4)$, but empirically as roughly $O(n^{3.4})$, due to the efficiency of batch operations in computation of whether n points are contained within a simplex, which practically scales sublinearly in n. Rousseeuw's algorithm has a runtime which theoretically scales as $O(n^2)$, and is practically validated as such with an empirical slope of 2.0. Our algorithm has an instance dependent runtime, where for isotropic Gaussians the runtime scales approximately as $O(n^{1.5})$. Full experimental details are detailed in Appendix B.

Analyzing the dramatic gain in Figure 1a, Figure 2 shows that only a very small fraction of points require exact computation of their simplicial depth. This highlights the gain afforded by adaptivity; many points require only a coarse approximation of their depth before they can be eliminated. In these simulations, no errors were recorded for our adaptive scheme. In all trials the threshold for deciding to compute a point's depth exactly was if $t_r \geq n \log n$ simplicies needed to be computed. Only approximate gaps based on the algorithm's output are plotted, leading to a jagged end behavior in Figure 2. We tested data sets ranging from $n = 200$ to $n = 20k$, and observe similar empirical gap distributions for all n tested, shown in Figure 3. The empirical CDFs can be well fit by a power law distribution, and the corresponding performance gain estimated by Corollary 1.2.

In Figure 1b we show that our algorithm adapts as desired on a point by point basis. We plot the number of computations required for point x_i (which we denote as T_i) as a function of Δ_i^{-2}, which exposes a linear relationship between T_i and Δ_i^{-2} up until T_i exceeds the threshold for exact computation, at which point the number of pulls required is constant.



(a) (b)

Figure 2: Empirical CDF of gaps when data is drawn from an isotropic Gaussian in $d = 2$ for a) $n = 200$, b) $n = 20k$. Red line indicates exact computation threshold. Averaged over 10 random instances, with 2 trials per instance, gaps approximately computed using our adaptive method. Fitting the gaps to a power law distribution yields $\alpha = 1.15$ for (a) and $\alpha = 1.16$ for (b). Corollary 1.2 gives that the complexity of our adaptive algorithm in this setting is $\tilde{O}\left(n^{2-\alpha/2}\right) = \tilde{O}\left(n^{1.42}\right)$, close to the empirically observed $n^{1.5}$, empirically validating the dramatic theoretical improvement of our adaptive method.

5. Discussion

With our adaptive methods in place, we discuss two additional algorithmic design points.

5.1 Adapting to Variances is Unnecessary

Viewing this problem from the multi-armed bandit perspective, we note that since our arm rewards are Bernoulli, our arm pulls are bounded leading to an immediate bound on the sub-Gaussian parameter of the arm distributions. As we will show, this coarse bound is sufficient; adapting to the variance of the arms can yield only a constant factor improvement for simplicial median identification when considering the refined definition of simplicial depth as the average number of open and closed simplices that contain the query point (Burr et al., 2006).

For a data set of n points in $\mathbb{R}^2$, Boros and Füredi (1984) showed that a point can be contained in no more than a $2^{-d} + o(1)$ fraction of simplicies. Conversely, it was shown by Bárány (1982) and discussed in the context of simplicial depth by Gil et al. (1992) that there must exist a point contained within at least a $\frac{1}{(d+1)^{d+1}}$ fraction of simplices. Taking these together, the depth of the simplicial median, μ_1, is bounded away from both 0 and 1 by constants, which are functions of d but independent of n as $n \to \infty$.

In the multi-armed bandit literature, several recent works have focused on pure exploration settings with unknown variances (Lu et al., 2021), which yield results applicable to this setting of arms with unknown means and variances but bounded rewards. In this setting we would like confidence intervals whose width depends on the variance of the arm in question, for example those offered by empirical variants of Bernstein's inequality as developed by Maurer and Pontil (2009). As was shown by Lu et al. (2021), the optimal sample complexity in the case where arms have unknown means μ_i and variances σ_i^2 is essentially lower bounded by

$$\tilde{\Omega}\left(\sum_{i=1}^{n}\left[\frac{\sigma_i^2}{\Delta_i^2}+\frac{1}{\Delta_i}\right]\log\left(\frac{1}{\delta}\right)\right). \tag{3}$$

Since $\sigma_i^2 = \mu_i(1-\mu_i)$ for these Bernoulli rewards, we can simplify Equation (3) in our setting to bound the cost of eliminating the i-th arm as

$$\frac{\sigma_i^2}{\Delta_i^2}+\frac{1}{\Delta_i} = \frac{\mu_1-\mu_i^2}{\Delta_i^2} \geq \frac{\mu_1-\mu_1^2}{\Delta_i^2} = \Theta(\Delta_i^{-2}), \tag{4}$$

where the final equality follows from the fact that μ_1 is upper and lower bounded by constants with respect to n. Equation (4) shows that variance adaptation can only yield a d dependent constant factor improvement in this setting, and thus that adapting to the variance of the arms is unnecessary to achieve order optimal performance. The intuition behind this is that if the gap Δ_i is small then the arm mean μ_i must be large, $\Theta(1)$, and so its variance is constant. On the other hand if the gap is large and $\Delta_i = \Theta(1)$,then Δ_i^{-1} and Δ_i^{-2} are the same up to constants, and so this arm does not require many samples to eliminate. For alternative objectives which necessitate differentiating between points with shallow simplicial depth, variance adaptation may yield more substantial improvements. In these instances, techniques similar to those used by Lu et al. (2021) can be readily employed in our algorithms for alternative objectives, as discussed in prior sections.

5.2 Barycentric Coordinate Computation

One aspect of simplicial depth computation that we did not focus on in this work is how arm pulls are actually performed. That is, for a given point $x_0 \in \mathbb{R}^d$, how do we check whether it is in the simplex created by $d+1$ other points? Verifying if x_0 is contained within the convex hull (simplex) of $d+1$ points $x_1, \ldots, x_{d+1}$ is equivalent to computing the barycentric coordinates of x_0 with respect to $\{x_i\}_{i=1}^{d+1}$, and verifying that all are positive. Mathematically, denoting the normalized barycentric coordinates of x_0 as $\lambda \in \mathbb{R}^{d+1}$ where $\sum_i \lambda_i = 1$, we wish to express $x_0 = \sum_{i=1}^{d+1} \lambda_i x_i$. To solve for such a λ, we define our centered data matrix $X = [x_1 - x_{d+1} \; x_2 - x_{d+1} \; \ldots \; x_d - x_{d+1}]$, and by taking $\lambda_{[d]}$ as the first d coordinates of λ we have $X\lambda_{[d]} = x_0 - x_{d+1}$ and $\lambda_{d+1} = 1 - \sum_{i=1}^d \lambda_i$. x_0 is contained within the closed simplex with vertices $\{x_i\}_{i=1}^{d+1}$ if and only if $\lambda_i \geq 0$ for all i. Note that while solving this for one query point x_0 requires 1 linear system solve, practically requiring $O(d^3)$ time, solving this for several different query points simultaneously is significantly more efficient. If we precompute X^{-1} (or an LU decomposition of X), then computing the barycentric coordinates for an additional point x_0' with respect to the same set of points $\{x_i\}_{i=1}^{d+1}$ will require only a matrix vector multiplication (or backsolving), requiring $O(d^2)$ time. This means that while obtaining k samples for the simplicial depth of a single point x_0 requires $O(kd^3)$ time, obtaining 1 sample each for k different points requires only $O(d^3 + kd^2)$ time, a factor of d improvement. More importantly, solving many systems of linear equations for the same data matrix X is efficient from a computational perspective, as this allows for better utilization of BLAS (Basic Linear Algebra Subprograms). To this end, our algorithm is optimized for batch efficiency, operating in a number of rounds scaling logarithmically with $1/\varepsilon$ or $1/\texttt{eCost}$, which is the minimal possible to achieve order optimal sample complexity up to logarithmic factors, as discussed by Karpov and Zhang (2020). Within each round, our uniform sampling of arms allows us to obtain this theoretical factor of d improvement as well as the practical benefits of more efficient batched operations.

6. Concluding Remarks

In this work we proposed a novel algorithmic framework for the adaptive computation of data depth. This method enables us to compute the simplicial median of a data set efficiently by approximating the depth of each point to the necessary accuracy, with dramatic time savings allowing for these computations to be run on larger and higher dimensional data sets than was previously possible. We provided instance dependent theoretical guarantees for our adaptive method, and showed its excellent empirical performance.

There are several important lines of future work regarding adaptive simplicial depth computation. One is to theoretically analyze the effects of arm correlation on the performance of Algorithm 1. In the work of Baharav and Tse (2019) it was shown that correlated arm pulls could provably improve sample complexity in the fixed budget setting. In this fixed confidence regime however, a more sophisticated scheme is required to estimate and exploit these unknown dependencies. Another direction is to note that there exists spatial information that we are neglecting; if two points x_1, x_2 are close to each other then $D_n(x_1)$ should not be too far from $D_n(x_2)$, depending on the positions of the other points. This can potentially improve the algorithmic dependence on n, necessitating a more sophisticated bandit algorithm based on bandits in metric spaces (Mason et al., 2019).

In addition to simplicial depth, our proposed technique of using adaptivity to efficiently approximate the points' depths to the necessary accuracy can be extended to several other common measures of depth including majority depth, Oja depth, and likelihood depth (Liu et al., 1999). Majority depth is defined with respect to half-spaces, where the sample majority depth is defined as $M_n(x) \triangleq \mathbb{E}\left[\mathbb{1}\{x \text{ is in a major side determined by } (X_1, \ldots, X_d)\}\right]$, where major side indicates the half-space bounded by the hyperplane containing $(X_1, \ldots, X_d)$ which has probability at least one half, where $(X_1, \ldots, X_d)$ are drawn uniformly at random from all subsets of $\{x_i\}$ of size d. This is naturally amenable to our technique of adaptive sampling, as now instead of computing whether x is within a simplex of $d+1$ random points, we rather check whether it is on a given side of a hyperplane defined by d points, and adaptively approximate the more promising points. The sample Oja depth is defined as $OD_n(x) \triangleq \left(1 + \mathbb{E}\left[\text{volume}(S[x, X_1, \ldots, X_d])\right]\right)^{-1}$, for $X_1, \ldots, X_d$ drawn uniformly at random from all subsets of $\{x_i\}$ of size d, where S is the closed simplex formed by the $d+1$ points. Adaptive sampling can be incorporated using our techniques, as not all $\binom{n}{d}$ subsets need to be enumerated for each point in order to determine whether it is the deepest or not. Note that we are no longer obtaining unbiased estimates of the Oja depth itself, and now essentially need to transfer the confidence intervals from y to $(1+y)^{-1}$ where $y = \mathbb{E}\left[\text{volume}(S[x, X_1, \ldots, X_d])\right]$ is the quantity we approximate. A final common measure of depth that we highlight is likelihood depth, which can also be adaptively approximated in certain cases. The distributional likelihood depth for a point x drawn from a distribution with density f is $L(x) = f(x)$, where the sample version is based on an empirical estimate of the density at x. For kernel-based density estimates adaptivity can be utilized, as then $L_n(x) = \frac{1}{n}\sum_{i=1}^{n} k(x, x_i)$, at which point we observe that not all n kernel computations need to be performed for each point. These alternative notions of depth show the natural transferability of our algorithmic principle; if *relative* ordering is the object of interest, adaptivity can yield dramatic gains.

Acknowledgments

Baharav was supported in part by the NSF GRFP and the Alcatel-Lucent Stanford Graduate Fellowship. Lai was supported in part by the NSF under DMS-1811818.

Appendix A. Proofs

In this Appendix, we provide the full proofs missing from the main text.

A.1 Proof of Theorem 1

We analyze Algorithm 1 following the steps of Hillel et al. (2013), with the necessary modifications for the BMO method similar to those made by Bagaria et al. (2021). We begin by showing that each depth estimator $\hat{\mu}_i^r$ is within $\varepsilon_r/2$ of the true simplicial depth μ_i of point x_i for all i, r with probability at least $1-\delta$. We do this by assuming that samples are drawn for all arms in each round, even if they are not observed, and bound the probability that any of these empirical estimates deviate from their mean. While some arms i may be eliminated prior to round r, we still consider what their estimator $\hat{\mu}_i^r$ would have been were it computed in round r.

Lemma 1 *With probability at least $1-\delta$, $|\hat{\mu}_i^r - \mu_i| < \varepsilon_r/2$ for all $i \in [n]$ and all $r \in \mathbb{N}$ simultaneously, where*

$$\hat{\mu}_i^r \triangleq \frac{1}{t_r} \sum_{j=1}^{t_r} \mathbb{1}\{x_i \in S[x_{i_1}^{(j)}, \dots, x_{i_{d+1}}^{(j)}]\}$$

and $\{x_{i_1}^{(j)}, \dots, x_{i_{d+1}}^{(j)}\}$ is drawn uniformly at random without replacement from $\mathcal{D}$, independently for each $j \in [t_r]$.

Proof [Proof of Lemma 1] Applying Hoeffding's inequality for $[0,1]$ bounded random variables (Wainwright, 2019) yields

$$\begin{aligned}\mathbb{P}\left(\bigcup_{i,r}\{|\hat{\mu}_i^r - \mu_i| \geq \varepsilon_r/2\}\right) &\leq \sum_{i,r} \mathbb{P}\left(|\hat{\mu}_i^r - \mu_i| \geq \varepsilon_r/2\right) \\ &\leq \sum_{i,r} 2\exp\left(-2t_r(\varepsilon_r/2)^2\right) \\ &\leq \sum_{r=1}^{\infty} \frac{\delta}{2r^2} \\ &\leq \delta \qquad \blacksquare\end{aligned}$$

With this lemma in place, we are now able to prove Theorem 1.

Proof [Proof of Theorem 1] Conditioning on the event in Lemma 1 where our confidence intervals hold, the simplicial median (best arm) is not eliminated during the course of the algorithm. Further, any suboptimal arm i must be eliminated by the end of round $r_i = \lceil \log_2(2/\Delta_i) \rceil$, since then $\Delta_i \geq 2\varepsilon_r$, and so $\max_{j \in S_r} \hat{\mu}_j^r - \varepsilon_r > \mu_1 - 3\varepsilon_r/2 \geq \mu_i + \varepsilon_r/2 > \hat{\mu}_i^r$. Thus, the algorithm must terminate after at most $\lceil \log_2(2/\Delta_2) \rceil$ rounds, at which point only the simplicial median will remain. If multiple points attain the maximum simplicial depth, then the algorithm will still terminate once $t_{r_i} \geq \texttt{eCost}$, requiring at most $\lceil \log_2(\sqrt{\texttt{eCost}}/2) \rceil$ rounds due to the doubling accuracy.

Analyzing the algorithm's requisite sample complexity, since arm i must be eliminated prior to round $r_i = \lceil \log_2(2/\Delta_i) \rceil$, it can be pulled no more than

$$\begin{aligned} t_{r_i} &= \lceil 2\varepsilon_{r_i}^{-2} \log(4nr_i^2/\delta) \rceil \\ &\leq 32\Delta_i^{-2} \log(16n \log_2^2(2/\Delta_i)/\delta) + 1 \end{aligned} \tag{5}$$

times. Here, we used inside the logarithm that $r_i \geq 1$ and so $r_i \leq 2\log_2(2/\Delta_i)$. Additionally, due to our ability to exactly compute the simplicial depth of a point, our algorithm will not pull any single arm too many times. Concretely, since the mean of arm i is exactly computed if $t_{r_i} \geq \texttt{eCost}$, at most `eCost` computation is done before x_i's depth is exactly computed, i.e. at most $2 \times \texttt{eCost}$ total work.

Since the algorithm must terminate if there is only one remaining arm, the best arm will be pulled the same number of times as the second most pulled arm. Thus, defining $r_1 \triangleq r_2$ and $\Delta_1 \triangleq \Delta_2$, on the event where the confidence intervals hold the total sample complexity will be at most

$$\sum_{i=1}^{n} t_{r_i} \leq n + \sum_{i=1}^{n} \min\left(\frac{32 \log\left(\frac{16n}{\delta} \log_2^2\left(\frac{2}{\Delta_i}\right)\right)}{\Delta_i^2}, 2 \times \texttt{eCost} \right). \tag{6}$$

■

A.2 Proof of Corollary 1.2

In this subsection we show that when the gaps in the data follow a power law distribution, the sample complexity of Algorithm 1 is significantly better than the naive $O(n^d)$ required for simplicial median identification. This average case analysis where the gaps follow a power law distribution follows similarly to that of Corollary 1 in (Bagaria et al., 2021).

Proof [Proof of Corollary 1.2] We have by Theorem 1 that with probability at least $1 - \delta$ Algorithm 1 will successfully return the simplicial median. Since the event where our confidence intervals hold is independent of the random gaps that are drawn, we can integrate out the expected sample complexity with respect to the random gaps satisfying $F(\Delta) = \Delta^\alpha$. Thus, on this success event our number of samples M satisfies

$$\begin{aligned} \mathbb{E}\{M\} &= O\left(\mathbb{E}\left\{ \sum_{i=1}^{n} \min\left(\frac{\log\left(\frac{n}{\delta} \log(\frac{1}{\Delta_i})\right)}{\Delta_i^2}, \texttt{eCost} \right) \right\} \right) \\ &= O\left(n \int_{\Delta=0}^{1} \min\left(\frac{\log\left(\frac{n}{\delta} \log(\frac{1}{\Delta})\right)}{\Delta^2}, \texttt{eCost} \right) f(d\Delta) \right) \\ &= O\left(n \left(\texttt{eCost}^{1-\alpha/2} + \log\left(\frac{nd\log n}{\delta}\right) \int_{\Delta=\texttt{eCost}^{-1/2}}^{1} \alpha\Delta^{\alpha-3} d\Delta \right) \right) \\ &= \begin{cases} O\left(n \log\left(nd/\delta\right) \texttt{eCost}^{1-\alpha/2}\right), & \text{for } \alpha \in [0,2), \\ O\left(n \log\left(nd/\delta\right) \log(\texttt{eCost})\right), & \text{for } \alpha = 2, \\ O\left(n \log\left(nd/\delta\right)\right), & \text{for } \alpha > 2. \end{cases} \end{aligned} \tag{7}$$

noting that $\log(1/\Delta) \leq d \log n$ for $\Delta \geq \texttt{eCost}^{-1/2}$. ■

A.3 Proof of Theorem 2

We analyze Algorithm 2 similarly to Algorithm 1, showing that the good event ξ where our mean estimators stay within their confidence intervals occurs with high probability. Then, on this good event, our algorithm will correctly identify the k-deepest elements, up to an allowable additive ε. To this end, we define the difficulty measure $\Delta_i^{(k)}$, which is defined as $\Delta_i^{(k)} = \max(\mu_i - \mu_{k+1}, \varepsilon)$ if $i \leq k$ and $\Delta_i^{(k)} = \max(\mu_k - \mu_i, \varepsilon)$ if $i > k$, as estimating the mean of point i to accuracy $\varepsilon/2$ is also sufficient.

Proof [Proof of Theorem 2] As in the proof of Theorem 1 we assume that all arms are sampled in all rounds until termination, even if they are not observed, and bound the probability that any of these estimators $\{\hat{\mu}_i^r\}_{i,r}$ stray outside of their confidence intervals via Lemma 1. Denote this good event in Lemma 1 as ξ.

Prior to the termination condition in line 16, on the event ξ it must be that 1) no point i with $\mu_i < \mu_k$ will be added to the acceptance set A, 2) no point i with $\mu_i > \mu_k$ will be eliminated without being added to A, 3) all points i with $\Delta_i \geq 2\varepsilon_r$ will be removed from the active set by the end of round r. These can be seen by induction on r, on the good event ξ. Thus, when the algorithm exits the sampling loop, all arms with $\Delta_i \geq \varepsilon$ will have been removed from the active set. Examining these points by cases, observe that points with $\mu_i > \mu_k + \varepsilon$ will necessarily be contained within the output set A. Conversely, those points with $\mu_i < \mu_k - \varepsilon$ will necessarily not be contained within the output set A. Thus, on this event ξ the set A satisfies

$$\{i : \mu_i > \mu_k + \varepsilon\} \subseteq A \subseteq \{i : \mu_i \geq \mu_k - \varepsilon\}. \tag{8}$$

Taking $\varepsilon = 0$ recovers the exact top-k, assuming $\mu_k > \mu_{k+1}$.

Bounding the number of pulls required, a point i must be eliminated by round $r_i = \lceil \log_2(2/\Delta_i^{(k)}) \rceil$. Either it will have been eliminated by round r_i where $\Delta_i^{(k)} \geq 2\varepsilon_{r_i}$, or $\varepsilon_r \leq \varepsilon/2$ in which case the algorithm will terminate. Following the argument in Theorem 1, this implies that the number of pulls required to eliminate arm i satisfies

$$\begin{aligned} t_{r_i} &= \lceil 2\varepsilon_{r_i}^{-2} \log(4nr_i^2/\delta) \rceil \\ &\leq 32 \left(\Delta_i^{(k)}\right)^{-2} \log(16n \log_2^2(2/\Delta_i^{(k)})/\delta) + 1. \end{aligned} \tag{9}$$

Utilizing the fact that our algorithm also terminates if the approximation cost is deemed to be greater than the exact computation cost, i.e. at most `eCost` time is spent we have a total sample complexity on ξ of at most

$$\sum_{i=1}^{n} \min \left(\frac{32 \log \left(\frac{16n}{\delta} \log_2^2 \left(2/\Delta_i^{(k)} \right) \right)}{\left(\Delta_i^{(k)} \right)^2}, 2 \times \texttt{eCost} \right), \tag{10}$$

giving us the desired result. ■

A.4 Proof of Theorem 3

In this section we provide the proof for our meta algorithm when applied to approximate simplicial depth computation.

By the choice of failure probabilities in line 10, all our `aDepth` calls will be correct with probability at least $1-\delta$, as

$$\begin{aligned}\mathbb{P}(\text{failure}) &= \mathbb{P}\left(\bigcup_{r\in\mathbb{N}}\bigcup_{i\in S_r}\{\texttt{aDepth}\text{ confidence intervals hold}\}\right)\\ &\le \sum_{r\in\mathbb{N}}\sum_{i\in S_r}\mathbb{P}\left(\texttt{aDepth}\text{ confidence intervals hold}\right)\\ &\le \sum_{r\in\mathbb{N}}\frac{\delta}{2r^2}\\ &\le \delta.\end{aligned}$$

Conditioning on the good event ξ where our `aDepth` confidence intervals hold, in the case of approximate simplicial median identification arm i must be eliminated by the end of round r_i, where

$$r_i = \min\left(\{r : r\in\mathbb{N}, \texttt{aCost}(2^{-r-1}, \delta/(2nr^2)) \ge \texttt{eCost}\} \cup \{\lceil\log(2/\varepsilon)\rceil, \lceil\log_2(2/\Delta_i)\rceil\}\right). \tag{11}$$

We show this by contradiction; first, assuming that arm i is active in round r where $r > r_i$. This implies that either $r > \lceil\log_2(2/\Delta_i)\rceil$, which we know cannot happen on the good event ξ where our confidence intervals hold, as was shown in the proof of Theorem 1. If $r > \lceil\log_2(2/\varepsilon)\rceil$, then $\varepsilon_r < \varepsilon/2$, which is included as a termination condition for our meta algorithm in this setting, as if all points have depths estimated to accuracy $\varepsilon/2$, then the point with the largest estimated simplicial depth must be within ε of the maximum. Finally, if $r > \operatorname{argmin}_{r\in\mathbb{N}} \texttt{aCost}(2^{-r-1}, \delta/(2nr^2)) \ge \texttt{eCost}$, then the depth of point i would be exactly computed.

Thus, this algorithm (when applied to approximate simplicial median computation) will spend at most r_i rounds approximating the depth of point i, and in total spend at most

$$\left(\sum_{r=1}^{r_i-1}\texttt{aCost}\left(2^{-r-1}, \delta/(2nr^2)\right)\right) + \min\left(\texttt{aCost}\left(\frac{\Delta_i}{8}, \frac{\delta}{2nr_i^2}\right), \texttt{eCost}\right) \tag{12}$$

computation on point i. Summing over the points in the data set yields the desired result.

Appendix B. Experimental Details

Here we provide details for reproducing the simulation results shown in the paper. All our code is publicly available on GitHub for reproducibility: `https://github.com/TavorB/adaSimplicialDepth`.

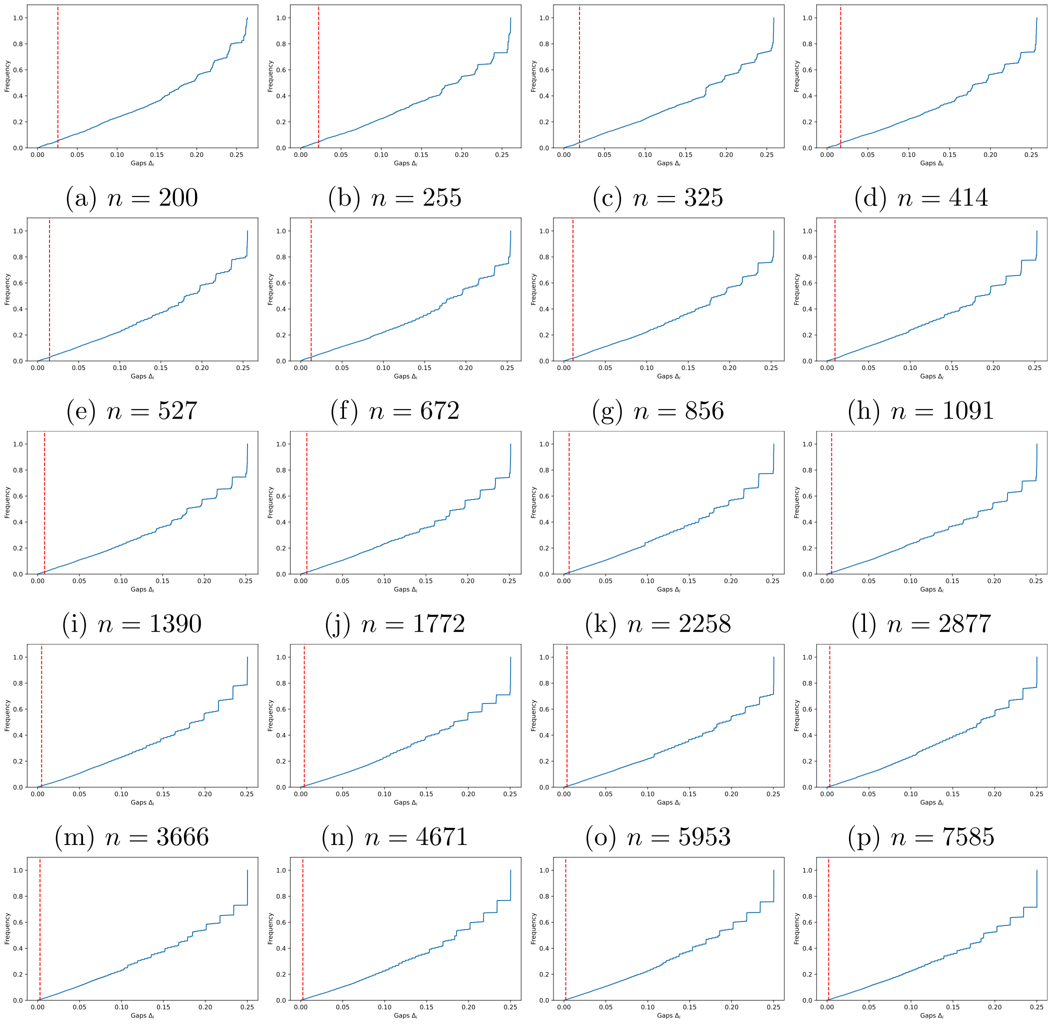
The data for each experiment was generated as n independent samples from an isotropic gaussian. We use the original definition of simplicial depth, as the probability that a random *closed* simplex contains the query point. All experiments were run on one core of an AMD

Opteron Processor 6378 with 500GB memory (no parallelism within a trial). All adaptive experiments are run with 20 trials per point. As simple Hoeffding-based confidence intervals are known to be overly conservative for multi-armed bandits in practice, to obtain tighter confidence intervals we set $t_r = \lceil .2 \times 2^{-r} \log(4nr^2/\delta) \rceil$ in all our simulations. The threshold for exact computation was determined as whether $t_r \geq n \log n$. Trials are grouped in pairs, where the same data set is used for both trials, but a different random seed is used to facilitate a different random selection of simplicies. Since computing ground truth results is computationally infeasible for all but the smallest of data sets, we declare the run a success and the correct result returned if the two independent runs on the same data set return the same result. For generating timing results for the brute force method, if there were more than 10,000 simplicies that needed to be generated we instead randomly sampled 10,000 simplices and then scaled the total estimated runtime by a factor of $\binom{n}{d+1}/10000$. For generating timing results for Rousseeuw's method, we ran their algorithm for finding the exact depths of 50 points within $\mathcal{D}$, and scaled the estimated runtime by a factor of $n/50$.

For Figure 1b, 50 simulations were run on the same data set with different random seeds, and the number of pulls per point averaged.

B.1 Additional Experiments

In Figure 3 we provide extended simulation results as in Figure 2. Observe that the maximum gap stays roughly constant at $D_n(x_i) \approx .25$. This is because for rotationally symmetric distributions, like the isotropic Gaussian we utilize, the probability that the origin is contained within the triangle constructed from 3 randomly drawn samples from this distribution is 1/4. Since many samples are drawn from this 2-dimensional distribution, the simplicial median is close to the origin, and has a simplicial depth of approximately 1/4, compared with points on the outside of the distribution which have a simplicial depth close to 0.



(a) $n = 200$ (b) $n = 255$ (c) $n = 325$ (d) $n = 414$

(e) $n = 527$ (f) $n = 672$ (g) $n = 856$ (h) $n = 1091$

(i) $n = 1390$ (j) $n = 1772$ (k) $n = 2258$ (l) $n = 2877$

(m) $n = 3666$ (n) $n = 4671$ (o) $n = 5953$ (p) $n = 7585$

(q) $n = 9666$ (r) $n = 12317$ (s) $n = 15695$ (t) $n = 20000$

Figure 3: Additional numerical experiments showing empirical CDF of gaps for different size datasets, as in Figure 2.

References

Peyman Afshani, Donald R Sheehy, and Yannik Stein. Approximating the simplicial depth. *European Workshop on Computational Geometry*, 32, 2015.

Boris Aronov and Sariel Har-Peled. On approximating the depth and related problems. *SIAM Journal on Computing*, 38(3):899–921, 2008.

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

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

Vivek Bagaria, Govinda Kamath, Vasilis Ntranos, Martin Zhang, and David Tse. Medoids in almost-linear time via multi-armed bandits. *International Conference on Artificial Intelligence and Statistics*, pages 500–509, 2018.

Vivek Bagaria, Tavor Z Baharav, Govinda M Kamath, and N Tse David. Bandit-based monte carlo optimization for nearest neighbors. *IEEE Journal on Selected Areas in Information Theory*, 2021.

Amitabha Bagchi, Amitabh Chaudhary, David Eppstein, and Michael T Goodrich. Deterministic sampling and range counting in geometric data streams. *ACM Transactions on Algorithms (TALG)*, 3(2):16–es, 2007.

Tavor Baharav and David Tse. Ultra fast medoid identification via correlated sequential halving. *Advances in Neural Information Processing Systems*, 32, 2019.

Tavor Z Baharav, Gary Cheng, Mert Pilanci, and David Tse. Approximate function evaluation via multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*. Proceedings of Machine Learning Research, 2022.

Imre Bárány. A generalization of carathéodory's theorem. *Discrete Mathematics*, 40(2-3): 141–152, 1982.

Vic Barnett. The ordering of multivariate data. *J. Roy. Statist. Soc. Ser. A*, 139(3):318–344, 1976.

Richard A Becker, William S Cleveland, and Allan R Wilks. Dynamic graphics for data analysis (with discussions and rejoinder by becker, cleveland and wilks). *Statistical Science*, 2(4):355–383, 1987.

Richard Bellman. *Dynamic programming: An Introduction.* Princeton University Press, 1957.

Ilai Bistritz, Tavor Z Baharav, Amir Leshem, and Nicholas Bambos. One for all and all for one: Distributed learning of fair allocations with multi-player bandits. *IEEE Journal on Selected Areas in Information Theory*, 2(2):584–598, 2021.

Endre Boros and Zoltán Füredi. The number of triangles covering the center of an n-set. *Geometriae Dedicata*, 17(1):69–77, 1984.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. ISSN 1935-8237.

Michael A Burr, Eynat Rafalin, and Diane L Souvaine. Simplicial depth: An improved definition, analysis, and efficiency for the finite sample case. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 72:195, 2006.

Andrew Y Cheng and Ming Ouyang. On algorithms for simplicial depth. In *Canadian Conference on Computational Geometry*, pages 53–56, 2001.

V. de la Peña and Tze Leung Lai. Theory and applications of decoupling. In *Probability and Statistical Models with Applications*, pages 117–145. Chapman & Hall/CRC, Boca Raton FL, 2000.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027. PMLR, 2016.

Joseph Gil, William Steiger, and Avi Wigderson. Geometric medians. *Discrete Mathematics*, 108(1-3):37–51, 1992.

Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems*, 26, 2013.

Yu-Chi Ho, R_S Sreenivas, and P Vakili. Ordinal optimization of deds. *Discrete Event Dynamic Systems*, 2(1):61–88, 1992.

Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–6. IEEE, 2014.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning*, volume 12, pages 655–662, 2012.

Govinda Kamath, Tavor Baharav, and Ilan Shomorony. Adaptive learning of rank-one models for efficient pairwise sequence alignment. *Advances in Neural Information Processing Systems*, 33:7513–7525, 2020.

Nikolai Karpov and Qin Zhang. Batched coarse ranking in multi-armed bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

Sumeet Katariya, Lalit Jain, Nandana Sengupta, James Evans, and Robert Nowak. Adaptive sampling for coarse ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 1839–1848. Proceedings of Machine Learning Research, 2018.

Julian Katz-Samuels and Kevin Jamieson. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*, pages 1781–1791. Proceedings of Machine Learning Research, 2020.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

Dong Woo Kim, Tze Leung Lai, and Huanzhong Xu. Multi-armed bandits with covariates: Theory and applications. *Statistica Sinica*, 31:2275–2287, 2021.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Tze-Leung Lai and Sidney Yakowitz. Machine learning and nonparametric bandit theory. *IEEE Transactions on Automatic Control*, 40(7):1199–1209, 1995.

Tze Leung Lai, Michael Sklar, and Nikolas Thomas Weissmueller. Novel clinical trial designs and statistical methods in the era of precision medicine. *Statistics in Biopharmaceutical Research*, 13(2):133–146, 2021.

Tze Leung Lai, Michael Benjamin Sklar, and Huanzhong Xu. Bandit and covariate processes, with finite or non-denumerable set of arms. *Stochastic Processes and their Applications*, 2022.

Daniel LeJeune, Reinhard Heckel, and Richard Baraniuk. Adaptive estimation for approximate k-nearest-neighbor computations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3099–3107. Proceedings of Machine Learning Research, 2019.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

Regina Y Liu. On a notion of data depth based on random simplices. *Ann. Statist.*, 18(1): 405–414, 1990.

Regina Y Liu, Jesse M Parelius, Kesar Singh, et al. Multivariate analysis by data depth: descriptive statistics, graphics and inference (with discussions and rejoinder by liu and singh). *Ann. Statist.*, 27(3):783–858, 1999.

Regina Y Liu, Robert Joseph Serfling, and Diane L Souvaine. *Data depth: robust multivariate analysis, computational geometry, and applications*, volume 72. American Mathematical Soc., 2006.

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *International Conference on Machine Learning*, pages 1690–1698. Proceedings of Machine Learning Research, 2016.

Pinyan Lu, Chao Tao, and Xiaojin Zhang. Variance-dependent best arm identification. In *Uncertainty in Artificial Intelligence*, pages 1120–1129. PMLR, 2021.

Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.

Ricardo A Maronna, R Douglas Martin, Victor J Yohai, and Matías Salibián-Barrera. *Robust Statistics: Theory and Methods (with R)*. John Wiley & Sons, 2019.

Blake Mason, Ardhendu Tripathy, and Robert Nowak. Learning nearest neighbor graphs from noisy distance samples. *Advances in Neural Information Processing Systems*, 32, 2019.

Blake Mason, Lalit Jain, Ardhendu Tripathy, and Robert Nowak. Finding all ε-good arms in stochastic bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

Blake Mason, Ardhendu Tripathy, and Robert Nowak. Nearest neighbor search under uncertainty. In *Uncertainty in Artificial Intelligence*, pages 1777–1786. Proceedings of Machine Learning Research, 2021.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *Conference on Learning Theory*, 22, 2009.

Vasilis Ntranos, Govinda M Kamath, Jesse M Zhang, Lior Pachter, and N Tse David. Fast and accurate single-cell rna-seq analysis by clustering of transcript-compatibility counts. *Genome Biology*, 17(1):1–14, 2016.

Hannu Oja. Descriptive statistics for multivariate distributions. *Statistics & Probability Letters*, 1(6):327–332, 1983.

Alexander Pilz, Emo Welzl, and Manuel Wettstein. From crossing-free graphs on wheel sets to embracing simplices and polytopes with few vertices. *Discrete & Computational Geometry*, 64(3):1067–1097, 2020.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Peter J Rousseeuw and Mia Hubert. Computation of robust statistics: Depth, median, and related measures. In *Handbook of discrete and computational geometry*, pages 1541–1554. Chapman and Hall/CRC, 2017.

Peter J Rousseeuw and Ida Ruts. Algorithm as 307: Bivariate location depth. *J. Roy. Statist. Soc. Ser. C*, 45(4):516–526, 1996.

Peter J Rousseeuw and Anja Struyf. A depth test for symmetry. In *Goodness-of-fit tests and model validity*, pages 401–412. Springer, 2002.

Mark J Schervish. A review of multivariate analysis (with discussions and author's rejoinder). *Statistical Science*, 2(4):396–433, 1987.

Max Simchowitz, Kevin Jamieson, and Benjamin Recht. The simulator: Understanding adaptive sampling in the moderate-confidence regime. In *Conference on Learning Theory*, pages 1794–1834. Proceedings of Machine Learning Research, 2017.

Anirudh Singhal, Subham Pirojiwala, and Nikhil Karamchandani. Query complexity of k-nn based mode estimation. In *2020 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2021.

Michael Sklar, Mei-Chiung Shih, and Philip Lavori. Bandit theory: applications to learning healthcare systems and clinical trials. *Statistica Sinica*, 31:2289–2307, 2021.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019. ISSN 1935-8237. doi: 10.1561/2200000068.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Mo Tiwari, Martin J Zhang, James Mayclin, Sebastian Thrun, Chris Piech, and Ilan Shomorony. Banditpam: Almost linear time k-medoids clustering via multi-armed bandits. *Advances in Neural Information Processing Systems*, 33:10211–10222, 2020.

John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians*, volume 2, pages 523–531, 1975.

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a Review Journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Robin Wellmann and Christine H Müller. Tests for multiple regression based on simplicial depth. *Journal of Multivariate Analysis*, 101(4):824–838, 2010.

Sid Yakowitz and Wing Lowe. Nonparametric bandit methods. *Annals of operations Research*, 28(1):297–312, 1991.

Martin Zhang, James Zou, and David Tse. Adaptive monte carlo multiple testing via multi-armed bandits. In *International Conference on Machine Learning*, pages 7512–7522. Proceedings of Machine Learning Research, 2019.