Improved Coresets and Sublinear Algorithms for Power Means in Euclidean Spaces

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we consider the problem of finding high dimensional power means: given a set A of n points in $\mathbb{R}^d$, find the point m that minimizes the sum of Euclidean distance, raised to the power z, over all input points. Special cases of problem include the well-known Fermat-Weber problem – or geometric median problem – where $z = 1$, the mean or centroid where $z = 2$, and the Minimum Enclosing Ball problem, where $z = \infty$.

We consider these problem in the big data regime. Here, we are interested in sampling as few points as possible such that we can accurately estimate m. More specifically, we consider sublinear algorithms as well as coresets for these problems. Sublinear algorithms have a random query access to the A and the goal is to minimize the number of queries. Here, we show that $\tilde{O}(\varepsilon^{-z-3})$ samples are sufficient to achieve a $(1+\varepsilon)$ approximation, generalizing the results from Cohen, Lee, Miller, Pachocki, and Sidford [STOC '16] and Inaba, Katoh, and Imai [SoCG '94] to arbitrary z. Moreover, we show that this bound is nearly optimal, as any algorithm requires at least $\Omega(\varepsilon^{-z+1})$ queries to achieve said approximation.

The second contribution are coresets for these problems, where we aim to find find a small, weighted subset of the points which approximate cost of every candidate point $c \in \mathbb{R}^d$ up to a $(1 \pm \varepsilon)$ factor. Here, we show that $\tilde{O}(\varepsilon^{-2})$ points are sufficient, improving on the $\tilde{O}(d\varepsilon^{-2})$ bound by Feldman and Langberg [STOC '11] and the $\tilde{O}(\varepsilon^{-4})$ bound by Braverman, Jiang, Krauthgamer, and Wu [SODA 21].

1 Introduction

Large data sets have shifted the focus of algorithm design. In the past, an algorithm might have been deemed feasible if its running time was polynomial in the input size and so a textbook *fast* algorithm can have time complexity for example quadratic. For truly gargantuan data sets, even linear time or nearly linear time algorithms could be considered too slow or requiring too much memory. This led to the emergence of the field of sublinear algorithms: How well can we solve a problem without reading the entire input?

Except for trivial problems, deterministic time sublinear algorithms do not exist. Our primary tool in designing sublinear algorithms is thus the following basic approach:

- Take a uniform sample of the input.
- Run an algorithm on the sample.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Hence, the performance of a sublinear algorithm is often measured in terms of its *query complexity*, i.e. the number of samples required such that we can extract a high quality solution in the second step above that generalizes to the entire input. Sublinear algorithms have close ties to questions in learning theory and estimation theory, where we are similarly interested in a quality to sample size tradeoff.

A perhaps very fundamental problem of primary importance in machine learning and data analysis is to efficiently estimate the parameters of a distribution. For example, given a distribution $\mathcal{D}$, how many samples do we need to estimate the mean? Even such a simple and basic question has surprisingly involved answers and are still subject to ongoing research [24].

In this paper, we investigate the possibility of estimating power means in high dimensional Euclidean spaces. Specifically, given an arbitrary set of points A, we wish to determine the number of uniform queries S such that we can extract a power mean m with

$$\text{cost}(m) := \sum_{p \in A} \|p - m\|^z \leq (1+\varepsilon) \cdot \min_{\mu} \sum_{p \in A} \|p - \mu\|^z,$$

where $\|p\|$ denotes the Euclidean norm of a vector p.

The power mean problem captures a number of important problems in computational geometry and multivariate statistics. For example, for $z = 1$, this corresponds to the Fermat-Weber problem also known as the geometric median. For $z = 2$, the problem is to determine the mean or centroid of the data set. Letting $z \to \infty$, we have the *Minimum Enclosing Ball* (MEB), where one needs to find the Euclidean sphere of smallest radius containing all input points.

These problems are convex and thus can be approximated in the near-linear time efficiently via convex optimization techniques. However, aside from the mean ($z = 2$), doing so in a sublinear setting is challenging and to the best of our knowledge, only the mean and the geometric median ($z = 1$) are currently known to admit nearly linear time algorithms.

Our main result is:

Theorem 1. *There exists an algorithm that, with query complexity $O(\varepsilon^{-z-3} \cdot \textit{polylog}(\varepsilon^{-1}) \log^2 1/\delta)$, computes a $(1+\varepsilon)$ approximate solution to the high dimensional power mean problem with probability at least $1 - \delta$.*

A key component in designing this algorithm is a novel analysis for *coresets* for these problems. Coresets are succinct summaries that approximate cost for any center solution c.

Theorem 2. *For any set of points in d-dimensional Euclidean space, there exists an ε-coreset for the high dimensional power mean problem of size $\tilde{O}(\varepsilon^{-2} \cdot 2^{O(z)})$.*

With the exception of the mean, previous coresets for these problems achieved bounds $O(\varepsilon^{-2} \cdot 2^{O(z)} \cdot \min(d, \varepsilon^{-2}))$ ([6, 15]), or had weaker guarantees such as merely approximating an optimal solution or requiring removal of outliers from the data set.

Comparing the bounds in Theorem 1 and Theorem 2, one may question whether the exponential dependency in z is necessary for computing an approximation. Indeed, previous sublinear algorithms for both the geometric median and the mean had a query complexity of $O(\varepsilon^{-2})$, and thus matched our coresets bounds. Unfortunately, we show that the exponential dependency in the power is indeed necessary even in a single dimension:

Theorem 3. *For any $\varepsilon > 0$ and z, any algorithm that computes with probability more than $4/5$ a $(1+\varepsilon)$-approximation for a one-dimensional power mean has query complexity $\Omega(\varepsilon^{-z+1})$.*

Hence, up to constants in the exponent, our sublinear algorithm is tight. Moreover, the algorithm is very simple to implement and performs well empircally.

1.1 Techniques

While stochastic gradient descent has been used for a variety of center-based problems [8, 9], it is difficult to apply it for higher powers. Indeed, [9] remark in their paper that even for the mean[1] ($z = 2$) their analysis does not work as the objective function is neither Lipschitz, nor strictly convex.

Hence, one needs to use new tools. A natural starting point is to use techniques from coresets, as they allow us to preserve most of the relevant information, using a substantially smaller number of points. Unfortunately, coresets have a drawback: the sampling distributions used to construct coresets is non-uniform and therefore difficult to use in a sublinear setting. The first step is therefore to design coresets from uniform sampling. To do this, we use and improve upon a technique originally introduced by Chen [7]. Chen showed that, given a sufficiently good initial solution q, one can partition the points into rings exponentially increasing radii such that the points cost the same, up to constant factors. Thereafter, taking a uniform sample of size $\tilde{O}(d \cdot \varepsilon^{-2})$ from each ring produces a coreset. Since there are at most $O(\log n)$ rings in the worst case, this yields a coreset of size $\tilde{O}(d \cdot \varepsilon^{-2} \cdot \log n)$.

To realize these ideas in a sublinear setting, we are now faced with a number of challenges. First, rings that are particularly far from q may contain very few points. This makes is difficult for a sublinear algorithm to access them. Second, partitioning the points into rings depends on the cost of q. It is very simple to construct examples where estimating the cost of an optimal power center requires $\Omega(n)$ many queries. Finally, this analysis loses both factors $\log n$ and d, which we aim to avoid.

We improve and extend this framework in two ways. First, we show that it is sufficient to only consider $O(\log \varepsilon^{-1})$ many rings, which, in of itself, already removes the dependency on $\log n$. Moreover, we show that it is possible to simply ignore any ring containing too few points, i.e. any ring with less than $\varepsilon^{z+O(1)} \cdot n$ points may be discarded. The intuition is that, while rings with few point may contribute significantly to the cost, these points do not influence the position of the optimal center by much. Thus, using a number of carefully chosen pruning steps, we show how to reduce both the problem of obtaining a sublinear algorithm as well as obtaining coresets to sampling from a select few rings containing many points.

The second improvement directly considers an improved analysis of sampling from rings. The state of the art coreset analysis by Feldman and Langberg [15] reduces the problem of computing a coreset to sampling in a (weighted) range space of bounded VC dimension. Unfortunately, this results in a dependency on d, which may be improved, using dimension reduction techniques, to an additional dependency on ε^{-2} [6].

We improve over these analyses via the chaining technique [29]. Essentially, the idea is to use a sequence of solutions c_h that approximate a candidate solution c with increasing accuracy as $h \to \infty$. Approximating the cost of a candidate center c can then be written as a telescoping sum of solutions induced by nets, i.e. $\text{cost}(c) = \sum_{h=0}^{\infty} \text{cost}(c_{h+1}) - \text{cost}(c_h)$, where c_h is a solution that has *distance* to c of the order 2^{-h} and $\text{cost}(c_0) = 0$. The notion of distance between candidate solutions is perhaps most easily understood by imagining a solution to be a cost vector v^c where the ith entry corresponds to the cost of the ith point of A when using c as a candidate center. Informally, we consider two solutions c and c' to have distance at most 2^{-h} if $|\|p_i - c\|^z - \|p_i - c'\|^z| \leq 2^{-h} \cdot (\|p_i - c\|^z + \|p_i - c'\|^z)$ for all points $p_i \in A$.

Instead of applying the union bound to the final element of the chain which corresponds to a naive union bound, the union bound is applied to each successive element of the chain. The total error is then the summed errors for every element of the chain, i.e. we aim to show for our designated coreset Ω

$$|\text{cost}(c_{h+1}) - \text{cost}(c_h) - (\text{cost}(\Omega, c_{h+1}) - \text{cost}(\Omega, c_h))| \leq 2^{-h} \cdot \varepsilon \cdot \text{cost}(c), \tag{1}$$

The improvement comes from the fact that we can determine a significantly smaller set of solutions c_h, depending on the approximation bound 2^{-h}. Summing the loss over all elements of the chain

[1] For the special case of the mean, Inaba et al. [20] observed that $O(\varepsilon^{-2})$ samples are nevertheless sufficient.

leads to an overall loss of at most $\varepsilon \cdot \text{cost}(c) \sum_{h=0}^{\infty} 2^{-h} \in O(1) \cdot \varepsilon \cdot \text{cost}(c)$. Thus, applying a union bound over all elements of the chain significantly improves the more conventional union bounds.

1.2 Related Work

Sublinear Approximation for Clustering: A number of sublinear algorithm are known for clustering problems. For k-Median, under the constraint that the input space has a small diameter, a constant factor approximation is known [10]. Ben-David [4] proposed a different set of conditions under which a sublinear algorithm for k-median and k-means exists. Other approximations, with different constraint are also known: for instance, Meyerson O'Callaghan and Plotkin [27] give an algorithm achieving a $O(1)$-approximation in time poly(k/ε) for discrete metrics, when each cluster has size $\Omega(n\varepsilon/k)$. For the 1-median problem, this assumption is always satisfied, and their algorithm gives a constant factor approximation in constant running time. The algorithm by Cohen, Lee, Miller, Pachocki, and Sidford [9] produces a $(1+\varepsilon)$-approximation in time $\min(nd\log^3(n/\varepsilon), d/\varepsilon^2)$ for Euclidean spaces of dimension d. Ding [11] and Clarkson, Hazan and Woodruff [8] showed how to obtain sublinear algorithms for the minimum enclosing ball problem assuming that either the algorithm is allowed to drop a fraction of the points, or with an additive error. For the unconstrained version of the problem, no sublinear time algorithm is possible. For the k-means problem, Bachem, Lucic, Hassani, and Krause [1] showed how to approximate the k-means++ algorithm in sublinear time. To our knowledge, no algorithm is known for higher distance powers z.

Coreset Constructions: A coreset is a weighted subset of the input points, such that the cost of any clustering is the same on the coreset than on the input points, up to a $(1 \pm \varepsilon)$ factor. The arguably most widely studied problem is coresets for the k clustering problems with powers of distances. Following a long line of work [3, 6, 7, 15, 16, 18, 17, 19, 22, 31], we now know that coreset of size $\tilde{O}(k\varepsilon^{-4} \cdot 2^{O(z)} \cdot \min(k, \varepsilon^{-2(z-1)}))$ exist. In a low-dimensional regime, this may be improved to $\tilde{O}(k^2 d\varepsilon^{-2} 2^{O(z)})$ and $\tilde{O}(kd\varepsilon^{-2z})$ [15]. For the special case of clustering with a single center, this yields the state of the art $\tilde{O}(\varepsilon^{-4})$ bound due to Braverman, Jian, Krauthgamer, and Wu [6]. For the Minimum Enclosing Ball problem and its generalizations, these algorithms yield no space saving, although sketches of weaker gaurantees of size $O(\varepsilon^{-1})$ exist [2]. Given a coreset, it is easy to compute a $(1+\varepsilon)$-approximation in time independent of n: one just needs to find a $(1+\varepsilon)$-approximation on the coreset points. The interested reader is referred to [14] for a recent survey.

1.3 Organization

We prove our key sampling result in Section 2. We follow up on that result by applying it to the sublinear setting (Section 3) and to coresets (Section 4). We conclude with a short experimental evaluation. Due to space constraints, all proofs are included in the supplementary material.

1.4 Preliminaries

We denote the Euclidean distance of a vector x by $\|x\| := \sqrt{\sum_{i=1}^{d} x_i^2}$. For a set of points A, we say that $\|A\|_0$ is the distinct number of points. Note that if A contains multiplicities $|A| \neq \|A\|_0$. We write $\tilde{O}(x)$ to denote $O(x \cdot \log^a x)$, where a is any constant. For any set A and candidate solution c, we defined $\text{cost}(A, c) = \sum_{p \in A} \|p - c\|^z$. If the set is clear from context, we simply write $\text{cost}(c)$. We frequently use the following generalized triangle inequality.

Lemma 1 (Triangle Inequality for Powers, [26])**.** *Let a, b, c be an arbitrary set of points in a metric space with distance function d and let z be a positive integer. Then for any $\varepsilon > 0$*

$$d(a,b)^z \leq (1+\varepsilon)^{z-1} d(a,c)^z + \left(\frac{1+\varepsilon}{\varepsilon}\right)^{z-1} d(b,c)^z$$

$$|d(a,b)^z - d(a,c)^z| \leq \varepsilon \cdot d(a,c)^z + \left(\frac{z+\varepsilon}{\varepsilon}\right)^{z-1} d(b,c)^z.$$

We also use the fact that uniform sampling is efficient to approximate the density: we review details on sampling in bounded VC dimension in the supplementary material.

Lemma 2 ([23]). *Given a range space $(X, \mathcal{R})$ with VC-dimension d, an constants $\varepsilon, \delta, \eta$, and a uniform sample $S \subset X$ of size at least $O(\frac{1}{\eta \cdot \varepsilon^{-2}}(d \log 1/\eta + \log 1/\delta))$, we have for all ranges $R \in \mathcal{R}$ with $|X \cap R| \geq \eta \cdot |X|$*

$$\left| \frac{|X \cap R|}{|X|} - \frac{|S \cap R|}{|S|} \right| \leq \varepsilon \cdot \frac{|X \cap R|}{|X|}$$

and for all ranges $R \in \mathcal{R}$ with $|X \cap R| \leq \eta \cdot |X|$

$$\left| \frac{|X \cap R|}{|X|} - \frac{|S \cap R|}{|S|} \right| \leq \varepsilon \cdot \eta$$

with probability at least $1 - \delta$.

The only range space we will consider is the one induced by Euclidean spheres centered around a single fixed point. The VC dimension induced of this range space is 2, which seems to be a well known fact, although we could not find a reference. For completeness, we added a short proof in the supplementary material.

First, we recall the commonly used coreset definition for clustering problems in Euclidean spaces.

Definition 1. *Let A be a set of points in $\mathbb{R}^d$. Then a set Ω is a strong (ε, z)-coreset if there exists a weight function $w : \Omega \to \mathbb{R}^+$ and a constant κ such that for any point c*

$$\left| \sum_{p \in A} \|p - c\|^z - (\sum_{p \in S} w_p \|p - c\|^z + \kappa) \right| \leq \varepsilon \cdot \sum_{p \in A'} \|p - c\|^z.$$

We say that Ω is a weak (ε, z)-coreset if for some $\alpha \in [0, 1]$ any point satisfying $\sum_{p \in \Omega} w_p \cdot \|p - c'\|^z \leq (1 + \alpha \cdot \varepsilon) \underset{c \in \mathbb{R}^d}{\text{argmin}} \sum_{p \in \Omega} w_p \cdot \|p - c\|^z$ also satisfies $\sum_{p \in A} \|p - c'\|^z \leq (1 + \varepsilon) \underset{c \in \mathbb{R}^d}{\text{argmin}} \sum_{p \in A} \|p - c\|^z$.

The difference between the two notions is that strong coresets give a guarantee for all candidate centers, whereas the weak coreset guarantee only applies for the optimum. In an offline setting, there is no difference in the size for our construction for either guarantee. In the sublinear setting, we will be satisfied with a weak coreset as it can be obtained with a nearly optimal query complexity.

2 Uniform Sampling Routine

In this section, we outline the proof of our basic sampling subroutine. We assume that we are given a point q, and a set of points R such that for all $p, p' \in R$

$$\|p - q\|^z \leq 2 \cdot \|p' - q\|^z.$$

In the following sections, we refer to R as a ring. Under this assumption, the following claim holds.

Theorem 4. *Let q and R be defined as above and let Ω be a uniform random sample consisting of $s \in \tilde{O}(\varepsilon^{-2} \cdot \log 1/\delta)$ points. Then with probability at least $1 - \delta$, we have for all candidate centers c*

$$\left| \textit{cost}(R, c) - \sum_{p \in \Omega} \frac{|R|}{s} \cdot \|p - c\|^z \right| \quad \leq \quad \varepsilon \cdot (\textit{cost}(R, q) + \textit{cost}(R, c)).$$

We prove this theorem, first by proving the following slightly simpler result and then showing how to apply the result recursively to remove dependency in $\|R\|_0$:

Lemma 3. *Let q and A be defined as above and let Ω be a uniform random sample consisting of $s \in \tilde{O}(\varepsilon^{-2} \cdot \log \|R\|_0 \cdot \log 1/\delta)$ points. Then with probability at least $1 - \delta$, we have for all candidate centers c that $\left| \textit{cost}(R, c) - \sum_{p \in \Omega} \frac{|R|}{s} \cdot \|p - c\|^z \right| \leq \varepsilon \cdot (\textit{cost}(R, q) + \textit{cost}(R, c))$.*

To set up a chaining analysis, we require two ingredients: (1) a notion of nets and (2) a Gaussian process. We focus on the latter first. Let $p_j \in \Omega$ with $j \in \{1, \ldots, s\}$ be the jth point of the sample. Observe for any $c \in \mathbb{R}^d$

$$\mathbb{E}_\Omega \left[\frac{n}{s} \|p_j - c\|^z \right] = \sum_{p \in R} \|p - c\|^z = \text{cost}(R, c).$$

Furthermore, let $g_1, \dots g_s$ be standard Gaussian random variables, i.e. Gaussians with mean 0 and variance 1. Then we have (see Appendix B.3 of [30])

$$\mathbb{E}_\Omega \sup_{c\in\mathbb{R}^d} \left| \frac{\sum_{j=1}^s \frac{|R|}{s} \|p_j - c|^z - \text{cost}(R,c)}{\text{cost}(R,q) + \text{cost}(R,c)} \right| \quad \leq \quad \sqrt{2\pi}\mathbb{E}_\Omega \mathbb{E}_g \sup_{c\in\mathbb{R}^d} \left| \frac{\sum_{j=1}^s \frac{|R|}{s} \|p_j - c|^z \cdot g_j}{\text{cost}(R,q) + \text{cost}(R,c)} \right| \quad (2)$$

Note that the first term in Equation 2 is the expected maximum deviation from the (normalized) expected cost of the sample. In other words, if $\sup_{c\in\mathbb{R}^d} \left| \frac{\sum_{j=1}^s \frac{|R|}{s} \|p_j - c\|^z - \text{cost}(R,c)}{\text{cost}(R,q)+\text{cost}(R,c)} \right| \leq \varepsilon$, we have the desired guarantee. Introducing Gaussians is standard in this type of analysis as concentration bounds typically used for weighted Bernoulli random variables such as Hoeffding or Bernstein are too weak. Our goal is therefore to show that $\mathbb{E}_\Omega \mathbb{E}_g \sup_{c\in\mathbb{R}^d} \left| \frac{\sum_{j=1}^s \frac{|R|}{s} \|p_j - c|^z \cdot g_j}{\text{cost}(R,q)+\text{cost}(R,c)} \right| \leq \frac{\varepsilon}{\sqrt{2\pi}}$.

We now define the nets we will use.

Definition 2. *Let R be a set of points and let q be a candidate solution. For any point c, let v^c be the $|R|$-dimensional vector, henceforth called a* cost vector*, such that $v_{p_i} = \|p_i - c\|^z$ for some arbitrary but fixed ordering of the points in R. For $\beta > 0$, we say that a set of vectors $\mathbb{N}_\beta$ is a β-net of R, if there exists a vector $v' \in \mathbb{N}_\beta$ such that for every point $p \in R$ with $\|p - c\| \leq \frac{8z}{\varepsilon} \cdot \|p - q\|$, we have*

$$\left| \|p - c\|^z - v'_p \right| \leq \beta \cdot \left(\|p - c\|^z + \|p - q\|^z \right).$$

We need to show the following three properties:

1. By bounding $\mathbb{E}_\Omega \sup_{c\in\mathbb{R}^d} \left| \frac{\sum_{j=1}^s \frac{|R|}{s} v_{p_j} - \sum_{p\in R_i} v_p}{\text{cost}(R,q)+\text{cost}(R,c)} \right|$ for $v \in \mathbb{N}_\varepsilon$we can also bound $\mathbb{E}_\Omega \sup_{c\in\mathbb{R}^d} \left| \frac{\sum_{j=1}^s \frac{|R|}{s} \|p_j - c|^z - \text{cost}(R,c)}{\text{cost}(R,q)+\text{cost}(R,c)} \right|$

2. There exist β-nets of size $\exp\left(z^2 \log \|R\|_0 \cdot \beta^{-2} \cdot \log \frac{1}{\varepsilon \cdot \beta}\right)$.

3. Let v be a cost vector and let v', v'' be cost vectors from β and $\beta/2$ nets $\mathbb{N}_\beta$ and $\mathbb{N}_{\beta/2}$, respectively. By bounding the term $\mathbb{E}_\Omega \mathbb{E}_g \sup_{c\in\mathbb{R}^d} \left| \sum_{j=1}^s \frac{|R|}{s} |v'_{p_j} - v''_{p_j}| \cdot g_j \right|$, we can also bound $\mathbb{E}_\Omega \mathbb{E}_g \sup_{v\in\mathbb{N}_\varepsilon} \left| \sum_{j=1}^s \frac{|R|}{s} v_{p_j} \cdot g_j \right|$

We start with item 1 via the following lemma.

Lemma 4. *Let $\mathbb{N}_{\varepsilon/10}$ be an $\varepsilon/10$-net of R. Then if $\sup_{v\in\mathbb{N}_{\varepsilon/10}} \frac{\left|\sum_{j=1}^s \frac{|R|}{s} v_{p_j} - \|v\|_1\right|}{\text{cost}(R,q)+\text{cost}(R,c)} \leq \frac{\varepsilon}{10}$ we have $\sup_{c\in\mathbb{R}^d} \frac{\left|\sum_{j=1}^s \frac{|R|}{s} \|p_j - c\|^z - \text{cost}(R_i,c)\right|}{\text{cost}(R,q)+\text{cost}(R,c)} \leq \varepsilon$ for all $c \in \mathbb{R}^d$.*

To prove item 2, we use terminal embeddings [13, 25, 28], defined as follows.

Definition 3 (Terminal Embeddings)**.** *Let A be a set of points in $\mathbb{R}^d$. A mapping $f : \mathbb{R}^d \to R^k$ is a terminal embedding if for all $p \in A$ and all $c \in \mathbb{R}^d$*

$$(1 - \varepsilon) \cdot \|p - c\|_2 \leq \|f(p) - f(c)\|_2 \leq (1 + \varepsilon) \cdot \|p - c\|_2.$$

The guarantee of a terminal embedding is very similar to the guarantee of the famous Johnson Lindenstrauss lemma, but stronger in one crucial detail. A terminal embedding preserves not only the distances between the points of A but also the distance between an arbitrary point in $\mathbb{R}^d$ and any point of A. Despite this stronger guarantee, the target dimension of terminal embedding is in fact no worse than that of the Johnson Lindenstrauss lemma. Specifically:

Theorem 5 ([28])**.** *For any n point-set $A \subset \mathbb{R}^d$, there exists a terminal embedding $f : \mathbb{R}^d \to \mathbb{R}^k$ with $k \in \gamma \cdot \varepsilon^{-2} \log n$ for some absolute constant γ.*

We now use the terminal embeddings to show that small nets exist.

Lemma 5. *Let R and q be defined as above. Then for every $\beta > 0$, there exists a β-net of R of size at most $\exp(\gamma \cdot z^3 \beta^2 \log \|R\|_0 \cdot \log \varepsilon^{-1})$, where γ is an absolute constant.*

We now move onto item 3.

Lemma 6. *Let R and q be defined as above and let Ω be a uniform sample consisting of s points. Then for any point c and $s \geq \eta \cdot z^3 2^{8z} \cdot \varepsilon^{-2} \cdot \log \|R\|_0 \cdot \log^3 \varepsilon^{-1}$ for some absolute constant η, we have $\mathbb{E}_\Omega \mathbb{E}_g \sup_{c \in \mathbb{R}^d} \left| \frac{\sum_{j=1}^{s} \frac{|R|}{s} \|p_j - c\|^z \cdot g_j}{cost(R,q) + cost(R,c)} \right| \leq \varepsilon$. Moreover, if $s \geq \eta \cdot z^3 2^{8z} \cdot \varepsilon^{-2} \cdot \log \|R\|_0 \cdot \log^4 \varepsilon^{-1} \log 1/\delta$ for some absolute constant η, then we have $\sup_{c \in \mathbb{R}^d} \left| \frac{\sum_{j=1}^{s} \frac{|R|}{s} \|p_j - c\|^z \cdot g_j}{cost(R,q) + cost(R,c)} \right| \leq \varepsilon$, with probability at least $1 - \delta$.*

The proof of Lemma 3 is now a direct consquence of Lemma 6 and Equation 2. To finally prove Theorem 4, we apply Lemma 6 recursively. The result is essentially a special case of Theorem 3.1 from [6].

Lemma 7. *Let R and q be defined as above. Suppose a uniform sample of size $s \in \tilde{O}(\Gamma \cdot \log \|R\|_0 \cdot \log 1/\delta)$ satisfies with probability at least $1 - \delta$ for all candidate centers c*

$$|cost(R, c) - \sum_{p \in \Omega} \frac{|R|}{s} \cdot \|p - c\|^z| \quad \leq \quad \varepsilon \cdot (cost(R, q) + cost(R, c)).$$

Then a uniform sample of size $\tilde{O}(\Gamma \cdot \log 1/\delta)$ achieves the same guarantee.

3 Sublinear Algorithm and Analysis

We first describe an algorithm that succeeds with constant probability. This probability can be amplified (non-trivially) using independent repetition. In the following we will use parameters β and η that depends on ε which we will specify later. We let A be the set of input points.

Algorithm 1 Sublinear Algorithm for Power Means

1. Sample a random point q.
2. Sample a set S of $O(\varepsilon^{-z-3} \cdot \text{poly}(\varepsilon^{-1}))$ points uniformly at random.
3. Compute the maximum distance d such that there exist $2/3 \cdot \varepsilon \cdot \eta \cdot |S|$ points with distance at least d from q. Discard all points at distance greater than d.
4. Define rings R_i such that $R_i \cap S$ contains all the points at distance $(d \cdot 2^{-i}, d \cdot 2^{-i+1}]$ from q, with $i = \{1, \ldots, \beta\}$.
5. If $|R_i \cap S| < \varepsilon\eta \cdot |S|$, remove all points in $R_i \cap S$ from S.
6. Solve the problem on S.

Our goal is to show that S satisfies a weak coreset guarantee. Specifically, we will show that S has the property that for all points $A' \subseteq A$ at distance at most d (defined in Algorithm 1) from q there exists weight function $w : S \to \mathbb{R}^+$ such that S is a strong coreset for A'. We then show that a strong coreset for A' is also weak coreset for A, i.e. we preserve the cost of all points in A', and we preserve the optimum for A.

In order to call Theorem 4 on only few rings, we now show that the loss incurred by only considering A' is indeed negligible.

Pruning Lemmas We first show that we can safely discard points that are sufficiently far away, as parameterized by γ. We recall the meaning of parameters: α is the approximation-factor of the initial solution, β is such that points at distance closer than $2^\beta d$ can be merged to the center (see Lemma 9), η is such that rings with less than an $\varepsilon\eta$-fraction of the points can be discarded (see Lemma 10)

Lemma 8. *Suppose we are given an α-approximate center q. Let $B(q, r)$ be the ball centered at q with radius $r = 4 \cdot \left(\frac{2\alpha \cdot OPT}{n}\right)^{1/z}$. Then the following two statements hold.*

1. *Any α-approximate center is in $B(q, r)$.*

2. *For any two points $c, c' \in B(q,r)$ and for any point p with $\|p-q\|^z > \gamma \cdot r^z$ with $\gamma > \varepsilon^{-z} \cdot (12z)^z$, we have $\|p-c\|^z \leq (1+\varepsilon) \cdot \|p-c'\|^z$.*

Unfortunately, we are not given a knowledge of OPT a priori. We therefore have to describe how to implement this lemma in a sublinear fashion. For this, we observe the following. First, the point q is an α approximation with probability at least $1/\alpha$. Second, the number of points that cost more than $\gamma \cdot \alpha \cdot \frac{OPT}{n}$ is at most $1/\gamma \cdot n$. Combining this observation with Lemma 2 ensures that we are not considering any points that are too far away. Since it is difficult to identify $\frac{OPT}{n}$ in a sublinear fashion, we will rely on additional pruning arguments for points that are close to q.

We proceed here in two steps. First, we consider the points that are very close to q. Second, we will show that the rings which contain too few points to be efficiently sampled have an overall negligible contribution to the cost.

Lemma 9. *Suppose that q is an α-approximate solution. Let $A_{near} \subset A$ be a set of points with cost at most $(\varepsilon/(\alpha 5z))^z \cdot \frac{OPT}{n}$. Let $\hat{A} = (1 \pm \varepsilon)|A_{near}|$ Then for any candidate solution c we have*

$$\left| \hat{A} \cdot \|q-c\|^z - \sum_{p \in A_{near}} \|p-c\|^z \right| \leq \varepsilon/\alpha \cdot \left(\sum_{p \in A_{near}} \|p-c\|^z + \boldsymbol{OPT} \right).$$

For rings with few points, we have the following.

Lemma 10. *Suppose that q that is an α-approximate solution. Let R_{cheap} the union of rings with $|R_i \cap A| < \varepsilon \cdot \eta \cdot n$ and with radius at most $4(\gamma \cdot \frac{\alpha \cdot OPT}{n})^{1/z}$, where γ is given by Lemma 8. Then, for any candidate solution c*

$$\sum_{p \in R_{cheap} \cap A} \|p-c\|^z \leq \varepsilon \cdot 4^{z+1} \cdot \beta \cdot \eta \cdot \gamma \cdot \alpha \cdot \sum_{p \in A} \|p-c\|^z.$$

While we will defer the exact parameterization to later, observe that for $\beta \in O(\log 1/\varepsilon)$, $\alpha \in 2^{O(z)}$ and $\eta \in O(\varepsilon^z)$ this entire sum can be bounded by $O(\varepsilon) \cdot OPT$.

The next lemma states that we can use these pruning results in a sublinear fashion.

Lemma 11. *Let q be a point that is an α-approximation and let S be a uniform sample consisting of $O\left(\alpha \cdot \eta^{-1} \cdot \varepsilon^{-3} polylog(\varepsilon^{-1} \cdot \delta^{-1})\right)$ points. Then with probability at least $1-\delta$ for all rings R_i,*

$$|R_i \cap A| - \varepsilon \cdot \max\left(n \cdot \eta, |R_i \cap A|\right) \leq \frac{|R_i \cap S| \cdot n}{|S|} \leq |R_i \cap A| + \varepsilon \cdot \max\left(n \cdot \eta,\ |R_i \cap A|\right).$$

Furthermore, let d as in the algorithm, i.e., such that $\frac{2}{3} \cdot \varepsilon \cdot \eta \cdot |S| \leq \left|S \setminus (B(q,d) \cap S)\right|$. Then $d < \left(3(\varepsilon\eta)^{-1} \cdot \frac{\alpha \cdot OPT}{n}\right)^{1/z}$

Since we do not know r, we also do not know which of the rings with $i < 0$, if any, satisfy $2^{i+1} \cdot d > (\varepsilon/\alpha 5z) \cdot (\eta/3\alpha)^{1/z}$. However, the maximum number of these rings is at most $\log(\frac{5\alpha z}{\varepsilon} \cdot \left(\frac{3\alpha}{\eta}\right)^{1/z})$, which will turn out to be of the order $O(\log \varepsilon^{-1})$. For all of the rings that are not light, i.e. we cannot discard or snap to q, we now use Theorem 4. To ensure that we can call Theorem 4, we invoke Lemma 2 for every ring.

Probability Amplification While the aforementioned algorithm is guaranteed to produce a $(1+\varepsilon)$-approximation with constant probability, amplifying this is non-trivial. Indeed, when running the algorithm multiple times, it is not clear how to distinguish a successful run from an unsuccessful one. The main issue in amplifying the probability lies in the initial solution q, as any invocation of Lemma 2 or Theorem 4 allows us to control the failure probability. The simplest way to achieve a success probability $1-\delta$ is to condition on $\|q-c\|^z \leq \delta \cdot \frac{OPT}{n}$. Unfortunately, this makes ε dependent on δ, which significantly increases the sampling complexity.

Instead, we use the following algorithm. We sample $m \in O(\log 1/\delta)$ points $q_1, \ldots q_m$ uniformly at random. For each point, we additionally sample $O(\varepsilon^{-2}(\log 1/\varepsilon + \log 1/\delta))$ points S_{q_i}. For q_i, let d_i

denote the minimum radius such that the points in $\frac{n}{|S_{q_i}|} \cdot |B(q_i, d_i) \cap S_{q_i}| > \frac{n}{2}$. We output the point with minimal d_i.

The following lemma shows that this point is, with probability at least $1 - \delta$, a 8^z approximation.

Lemma 12. *Given query access to A, we can identify with probability $1 - \delta$ a 8^z-approximate solution using $O(\varepsilon^{-2}(\log 1/\varepsilon + \log 1/\delta) \log 1/\delta)$ samples.*

To achieve an overall success probability of $1 - \delta$, we only need to sample from non-cheap rings. Thereafter, a high probability bound can be obtained by applying Theorem 4 applied to all R_i. The range space induced by rings centered around a single point q has constant VC dimension. Hence, Lemma 2 guarantees that a constant size sample will allow us to distinguish cheap rings from non-cheap ones.

4 Improved Coreset Constructions

Our algorithm is as follows. First, we compute a point q that is a reasonably good approximation to the optimum[2]. In the following, let $\alpha = \frac{\text{cost}(q)}{\text{OPT}}$. Let $\Delta = \frac{\text{cost}(q)}{n}$ be the average cost of the input points when clustering them to q. We now partition the points into rings. We define ring $R_i = \{p \in A \mid \left(\frac{\varepsilon}{2z}\right)^z \cdot \frac{1}{\alpha} \cdot 2^i \cdot \Delta \leq \|p - q\|^z \leq \left(\frac{\varepsilon}{2z}\right)^z \cdot \frac{1}{\alpha} \cdot 2^{i+1} \cdot \Delta\}$. To simplify the notation, we defined $\Delta_i := \left(\frac{\varepsilon}{2z}\right)^z \cdot \frac{1}{\alpha} \cdot 2^i$. Let $R_M = \bigcup_{i=1}^{\log\left(\frac{2z}{\varepsilon}\right)^{3z}} R_i$. For each ring R_i with $1 \leq i \leq \log\left(\frac{2z}{\varepsilon}\right)^{3z}$, we sample a subset S_i of s points uniformly at random, where each point is weighted by $\frac{|R_i|}{s}$. We weigh q by the number of points in $A \setminus R_M$. Finally, we set $\kappa = \sum_{i > \log\left(\frac{2z}{\varepsilon}\right)^{3z}} \sum_{p \in R_i} \|p - q\|^z$. We claim, for $s \in \tilde{O}(\varepsilon^{-2})$ that for the weights defined above, $\bigcup_{i=1}^{\log\left(\frac{2z}{\varepsilon}\right)^{3z}} S_i \cup \{q\}$ together with the constant κ is a coreset.

The analysis for every ring $R_i \in R_M$ is merely an application of Theorem 4. For the remaining points, we use the following lemma.

Lemma 13. *Let q, R_M and κ be defined as above. Then for any point $c \in \mathbb{R}^d$, we have*

$$|\mathit{cost}(A, c) - (\mathit{cost}(R_M, c) + \kappa + |A \setminus R_M| \cdot \|q - c\|^2)| \leq \varepsilon \cdot \mathit{cost}(A, c).$$

5 Experimental Evaluation

While we can prove that the algorithm can compute a good solution in constant time for every constant ε and z, even for moderately small ε (e.g. $\varepsilon = 1/2$) the sampling complexity becomes quite large for even small values of z, as indeed our lower bound shows is necessarily the case. Our experiments therefore aim at evaluating the performance of the sublinear algorithm on realistic, not necessarily worst case data sets.

As baseline algorithm, we implemented a simple version of a batched gradient descent. Since all considered objectives are convex, we can expect such an algorithm to find a good solution in a reasonable time. The sublinear algorithm ran Algorithm 1 before calling the batched gradient descent.

We selected two data sets from the UCI repository [12], both of which are under the Creative Commons license. The first data set is the 3D Road Network data set from [21]. It consists of elevation information with the attributes longitude, latitude and altitude. The total number of points is 434,874. For this data set, we considered all powers from $z = 3$ to 7. The second data set is the USCensus data set, consisting of the records from a 1990 census. The total size of the data set was 2,458,285 samples, each with 68 attributes. For this data set, we considered the powers $z = 3, 4, 5$.

The results essentially confirmed that the sublinear algorithm succeeded in finding a good candidate solution in a fraction of the time as batch gradient descent for essentially all considered problems.

A more extensive discussion can be found in the supplementary material.

[2] We described an option in detail for the sublinear algorithm. In the interest of keeping the presentation succinct, we defer to that part of the paper and omit further discussion.

References

[1] Olivier Bachem, Mario Lucic, S Hamed Hassani, and Andreas Krause. Approximate k-means++ in sublinear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[2] Mihai Badoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Comput. Geom.*, 40(1):14–22, 2008.

[3] Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1039–1050, 2019.

[4] Shai Ben-David. A framework for statistical clustering with constant time approximation algorithms for K-median and K-means clustering. *Mach. Learn.*, 66(2-3):243–257, 2007.

[5] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[6] Vladimir Braverman, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2679–2696. SIAM, 2021.

[7] Ke Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, 2009.

[8] Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. *J. ACM*, 59(5):23:1–23:49, 2012.

[9] Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2016.

[10] Artur Czumaj and Christian Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Structures & Algorithms*, 30(1-2):226–256, 2007.

[11] Hu Ding. A sub-linear time framework for geometric optimization with outliers in high dimensions. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, pages 38:1–38:21, 2020.

[12] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[13] Michael Elkin, Arnold Filtser, and Ofer Neiman. Terminal embeddings. *Theor. Comput. Sci.*, 697:1–36, 2017.

[14] Dan Feldman. Core-sets: An updated survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 10(1), 2020.

[15] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578, 2011.

[16] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM J. Comput.*, 49(3):601–657, 2020.

[17] Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

[18] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 291–300, 2004.

[19] Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in euclidean spaces: importance sampling is nearly optimal. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, 2020.

[20] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering (extended abstract). In *Proceedings of the Tenth Annual Symposium on Computational Geometry, Stony Brook, New York, USA, June 6-8, 1994*, pages 332–339, 1994.

[21] Manohar Kaul, Bin Yang, and Christian S. Jensen. Building accurate 3d spatial networks to enable next generation intelligent transportation systems. In *2013 IEEE 14th International Conference on Mobile Data Management, Milan, Italy, June 3-6, 2013 - Volume 1*, pages 137–146, 2013.

[22] Michael Langberg and Leonard J. Schulman. Universal ε-approximators for integrals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 598–607, 2010.

[23] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *J. Comput. Syst. Sci.*, 62(3):516–527, 2001.

[24] Gábor Lugosi and Shahar Mendelson. Mean estimation and regression under heavy-tailed distributions: A survey. *Found. Comput. Math.*, 19(5):1145–1190, 2019.

[25] Sepideh Mahabadi, Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. Nonlinear dimension reduction via outer bi-lipschitz extensions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1088–1101, 2018.

[26] Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. Performance of johnson-lindenstrauss transform for k-means and k-medians clustering. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1027–1038, 2019.

[27] Adam Meyerson, Liadan O'callaghan, and Serge Plotkin. A k-median algorithm with running time independent of data size. *Machine Learning*, 56(1):61–87, 2004.

[28] Shyam Narayanan and Jelani Nelson. Optimal terminal dimensionality reduction in euclidean space. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1064–1069. ACM, 2019.

[29] Jelani Nelson. Chaining introduction with some computer science applications. *Bull. EATCS*, 120, 2016.

[30] Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 764–773. ACM, 2014.

[31] Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 802–813, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] The potential societal impacts are the ones of any clustering algorithm; since this would be a paper by itself, we did not discussed it.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Limitations We make no assumptions on the data set. The sublinear algorithm has no assumptions other than random access to data points. While all sublinear algorithms make this assumption, it may not always be justified. The coreset algorithm has no assumptions at all, other than having access to random bits.

Finally, we only consider clustering problems for a single center. We believe that our techniques can lead to improved algorithms for multiple centers, but chose not to include those problems for two reasons. First, any sublinear algorithm with guarantees for k-clustering requires additional assumptions on the data set, which we wanted to avoid. Second, while we believe that the coreset result can be extended to $\tilde{O}(k^2/\varepsilon^2 \cdot 2^{O(z)})$ for k-clustering, it requires a non-uniform sampling strategy and thus we decided it to be out of scope.