High-Dimensional Geometric Streaming for Nearly Low Rank Data

Hossein Esfandiari *1 Praneeth Kacham *12 Vahab Mirrokni *1 David P. Woodruff *2 Peilin Zhong *1

Abstract
We study streaming algorithms for the $\ell_p$ subspace approximation problem. Given points $a_1, \ldots, a_n$ as an insertion-only stream and a rank parameter $k$, the $\ell_p$ subspace approximation problem is to find a $k$-dimensional subspace $V$ such that $(\sum_{i=1}^n d(a_i, V)^p)^{1/p}$ is minimized, where $d(a, V)$ denotes the Euclidean distance between $a$ and $V$ defined as $\min_{v \in V} \|a - v\|_2$. When $p = \infty$, we need to find a subspace $V$ that minimizes $\max_{a \in S} d(a, V)$. For $\ell_\infty$ subspace approximation, we give a deterministic strong coreset construction algorithm and show that it can be used to compute a poly$(k, \log n)$ approximate solution. We show that the distortion obtained by our coreset is nearly tight for any sublinear space algorithm. For $\ell_p$ subspace approximation, we show that suitably scaling the points and then using our $\ell_\infty$ coreset construction, we can compute a poly$(k, \log n)$ approximation. Our algorithms are easy to implement and run very fast on large datasets. We also use our strong coreset construction to improve the results in a recent work of Woodruff and Yasuda (FOCS 2022) which gives streaming algorithms for high-dimensional geometric problems such as width estimation, convex hull estimation, and volume estimation.

1. Introduction
Modern datasets are usually very high-dimensional and have a large number of data points. Storing the entire dataset to analyze them is often impractical and in certain settings impossible. In recent years, streaming algorithms have emerged as a way to process and understand the datasets in both a space and time-efficient manner. In a single-pass streaming setting, an algorithm is allowed to make only a single pass over the entire dataset and is required to output a "summary" of the dataset that is useful to solve a certain problem. In this work, we focus on streaming algorithms for high-dimensional geometric problems such as subspace approximation, width estimation, etc. Suppose we are given a set of $d$-dimensional points $a_1, \ldots, a_n$ and an integer parameter $k \leq d$. Given a subspace $V$, we define $d(a, V)$ to be distance between the point $a$ and subspace $V$ given by $\min_{v \in V} \|a - v\|_2$. The $\ell_p$ subspace approximation problem (Deshpande et al., 2011b), for $p \in [1, \infty]$, asks to find a $k$-dimensional subspace that minimizes $(\sum_{i=1}^n d(a_i, V)^p)^{1/p}$.

Note that for $p = \infty$, we want to find a $k$-dimensional subspace that minimizes the maximum distance from the given set of points. Related to the $\ell_\infty$ subspace approximation problem is the widely studied outer $(d - k)$ radius estimation problem (Varadarajan et al., 2007) which instead asks for a $k$-dimensional flat $F$ that minimizes $\max_{i \in [n]} d(a_i, F)$. The outer $(d - k)$ radius is a measure of how far the point set is from being inside a $k$-dimensional flat. Varadarajan et al. (2007) give a polynomial time algorithm for approximating the outer $(d - k)$ radius up to an $O(\sqrt{\log n})$ multiplicative factor. Their algorithm is based on rounding of a semidefinite program (SDP) relaxation. When $n$ is very large, their algorithm is not practical and cannot be implemented in the streaming setting. We give a time and space-efficient single pass streaming algorithm that approximates the outer $(d - k)$ radius up to an $O(\sqrt{\log(n\kappa)})$ factor, where $\kappa$ is a suitably defined condition number. Typically, the value of $k$ used is much smaller than $n$ and $d$ since in many settings, we have that the $n \times d$ matrix $A$ is a noisy version of an underlying rank $k$ matrix, for a small value of $k$.

Our main contribution is a simple deterministic algorithm that constructs a strong coreset for approximating $\max_{V} d(a_i, V)$ for any $k$-dimensional subspace $V$ in a single-pass streaming setting. We note that this notion of strong coreset is different from the strong/weak coreset definitions in some computational geometry works. When run on the stream of points $a_1, \ldots, a_n$, our algorithm selects a subset $S \subseteq [n]$ of points with $|S| = O(k \log^2(n\kappa))$, such that for all $k$-dimensional subspaces $V$, $\max_{i \in S} d(a_i, V) \leq \max_{i \in [n]} d(a_i, V) \leq O(\sqrt{\log(n\kappa)}) \max_{i \in S} d(a_i, V)$.

We stress that our coreset can be used to approximate the

---

*Equal contribution
1Google Research, USA
2Computer Science Department, Carnegie Mellon University, Pittsburgh, USA.
Correspondence to: Praneeth Kacham <pkacham@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).
max distance of the point set to any \( k \)-dimensional subspace and hence it is termed a strong coreset. We prove:

**Theorem 1.1** (Informal). Given a parameter \( k \) and \( n \) points \( a_1, \ldots, a_n \in \mathbb{R}^d \), Algorithm 1 selects a subset \( S \subseteq [n] \) of points with \( |S| = O(k \log^2 n \kappa) \), such that for all \( k \)-dimensional subspaces \( V \),

\[
1 \leq \frac{\max_{i \in [n]} d(a_i, V)}{\max_{i \in S} d(a_i, V)} \leq O(\sqrt{k} \log n \kappa).
\]

The streaming algorithm requires only enough space to store \( O(k \log^2 n \kappa) \) rows of \( A \) and can be implemented in time \( O(\text{nnz}(A) \log n + d \text{ poly}(k, \log n \kappa)) \) if one is allowed randomization.

In this result and its applications throughout the paper, the condition number \( \kappa \) can be replaced with \( n^{O(k)} \) assuming that all the entries in the input points are integers bounded in absolute value by \( \text{poly}(n) \). We note that some assumption on bit complexity is necessary in order to establish memory bounds in a streaming setting. Under suitable assumptions about the “noise” in the process generating the data, \( \kappa \) can be much smaller than \( n^{O(k)} \).

We then show using a simple reduction that the above theorem can be used to approximate the outer \((d - k)\) radius by running the streaming algorithm on the point set \( a_2 - a_1, \ldots, a_n - a_1 \).

We also prove the following lower bound showing that our coreset obtains near-optimal distortion up to logarithmic factors in \( n \) and \( \kappa \).

**Theorem 1.2** (Informal). Given parameters \( n, d \) and \( k \) with \( k = \Omega(\log n) \), any streaming algorithm that computes a strong coreset with distortion at most \( O(\sqrt{k / \log n}) \) with probability \( \geq 9/10 \) must use \( \Omega(n) \) bits of space.

We then turn to the \( \ell_p \) subspace approximation problem for general \( p \in [1, \infty) \). We observe that an instance of the \( \ell_p \) subspace approximation problem can be turned into an \( \ell_\infty \) subspace approximation problem by using the so-called min-stability property of exponential random variables. We scale each input point with appropriately chosen independent random variables and feed the scaled points to Algorithm 1. We obtain the following result:

**Theorem 1.3** (Informal). Given \( p \geq 1 \), a dimension parameter \( k \), and \( n \) points \( a_1, \ldots, a_n \in \mathbb{R}^d \), there is a randomized streaming algorithm that selects a subset \( S \subseteq [n] \), \( |S| = O(k \log^2 n \kappa) \) and assigns a weight \( w_i \geq 0 \) for \( i \in S \) such that if

\[
\tilde{V} = \arg\min_{k\text{-dim } V} \max_{i \in S} w_i \cdot d(a_i, V),
\]

then

\[
\left( \frac{\sum_{i=1}^n d(a_i, \tilde{V})^p}{\min_{k\text{-dim } V} (\sum_{i=1}^n d(a_i, V)^p)^{1/p}} \right)^{1/p} \leq k^{1/2 + 2/p} \text{poly}(\log^{3+3/p} n \kappa).
\]

The algorithm only uses \( O(d \cdot k \log^2 n \kappa) \) bits of space and runs in \( O(\text{nnz}(A) \log n + d \text{ poly}(k, \log n)) \) time.

While exponential random variables have been previously used in the context of \( \ell_p \) subspace embeddings and \( \ell_p \) moment estimation in streams, as far as we are aware, ours is the first work to use them in the context of subspace approximation.

We then show that recent algorithms of (Woodruff and Yassuda, 2022) can be improved using our coreset construction algorithm when the data points \( a_1, \ldots, a_n \) are “approximately” spanned by a low rank subspace. They give streaming algorithms for a host of geometric problems such as width estimation, volume estimation, Löwner-John ellipsoid computation, etc. The main ingredient of their algorithms is a deterministic \( \ell_\infty \) subspace embedding: their algorithm streams through the rows of an \( n \times d \) matrix \( A \) and selects a subset \( S \subseteq [n] \) of rows, \( |S| = O(d \log n) \) with the property that for all \( x \),

\[
\|A_S x\|_\infty \leq \|A x\|_\infty \leq \sqrt{d \log n} \|A_S x\|_\infty.
\]

Here \( \|x\|_\infty := \max_i |x_i| \) and \( A_S \) is the matrix \( A \) restricted to only those rows in \( S \). When the matrix \( A \) has rank \( d \), their algorithm necessarily needs \( \Omega(d^2) \) bits of space which is prohibitive when \( d \) is very large. In practice, many matrices \( A \) are very well-approximated by a matrix with far lower rank than \( d \) even when the rank of the matrix \( A \) is \( d \). Suppose \( A \) is well-approximated by a rank \( k \) matrix in the sense that there is a \( k \)-dimensional subspace \( V \) such that all the rows of \( A \) are not very far from \( V \). We show that if \( S \) is the coreset constructed by Algorithm 1, then for all unit vectors \( x \), \( \|A_S x\|_\infty \leq \|Ax\|_\infty \leq (C \sqrt{k} \log n \kappa) \|A_S x\|_\infty + C \Delta \log n \kappa \), where \( \Delta \) denotes the optimal rank-\( k \) \( \ell_\infty \) subspace approximation cost of the matrix \( A \). Thus, \( \|A_S x\|_\infty \) can be used to approximate \( \|Ax\|_\infty \) well when \( \Delta \) is small.

### 1.1. Previous Work

The rank-\( k \) \( \ell_\infty \) subspace approximation problem and more generally the rank-\( k \) \( \ell_\infty \) flat approximation problem have been previously studied for different values of \( k \). As discussed earlier, Varadarajan et al. (2007) give an SDP-based algorithm that can compute an \( O(\sqrt{\log n}) \) factor approximation for all values of \( k \). Being SDP-based, the algorithm is impractical in the streaming setting and when the number of points \( n \) is very large. We shall mostly discuss previous works relevant in the streaming setting.

For specific values of \( k = 0 \) and \( k = d - 1 \), Agarwal and Sharathkumar (2015) study upper and lower bounds on streaming algorithms. For \( k = 0 \), also known as the minimum enclosing ball (MEB) problem, they give a streaming algorithm that is a \( (1 + \sqrt{3})/2 \) approximation and show
that there is a small enough constant \( \alpha \) such that any \( \alpha \) approximation algorithm must use \( \min(n, \exp(d^{1/3})) \) space, thereby showing that there are no small-space streaming algorithms with a better than \( \alpha \) approximation. For \( k = d - 1 \), the so-called width estimation problem, they showed that any algorithm that approximates the cost up to a multiplicative \( \Theta(d^{1/3}) \) factor must use \( \Omega(n, \exp(d^{1/3})) \) bits of space, again ruling out small-space algorithms with better than \( d^{1/3} \) approximation factor.

Later, Chan and Pathak (2014) improved the approximation ratio of the algorithm of Agarwal and Sharathkumar (2015) to \( (1 + \sqrt{2})/2 \) for the MEB problem.

Recently, Tukan et al. (2022) give an algorithm to construct a coreset for the \( \ell_\infty \) subspace approximation problem with a size of \( \tilde{O}(k^{3k}) \). While an offline coreset construction can be converted into a streaming coreset construction using the merge-and-reduce procedure, the exponential dependence in \( k \) makes their algorithm impractical compared to our algorithm which needs to store only \( O(k \log(nk)^3) \) input points.

For the \( \ell_1 \) subspace approximation problem, Feldman et al. (2010) give a streaming algorithm to construct a coreset with \( \tilde{O}\left(d \left( \frac{k^2 \log \sqrt{k \log n}}{\varepsilon^3} \right)^{\poly(k)} \right) \) points that can be used to compute a \( 1 + \varepsilon \) approximation. When \( n \) and \( d \) are large, the space requirement of the coreset is infeasible. In comparison, although our algorithms do not give \( 1 + \varepsilon \) approximation, we can compute \( \poly(k, \log nk) \) approximations using only space necessary to store \( \poly(k, \log nk) \) points, which is much smaller than the coreset constructed by their algorithm.

For all values of \( p \), Kerber and Raghvendra (2014) give a dimensionality reduction procedure by showing that projecting the points to a random \( O(k^2 (\log k/\varepsilon \cdot \log n)/\varepsilon^2) \)-dimensional space preserves the \( \ell_p \) subspace approximation\(^2\) cost. For \( p = \infty \), their algorithm combined with the coreset construction algorithm of Woodruff and Yasuda (2022) can be used to approximate the \( \ell_\infty \) subspace approximation up to \( \poly(k, \log n) \) factors. But since the \( d \)-dimensional “information” is destroyed by the projection, we cannot recover a solution in the \( d \)-dimensional space.

In comparison, for \( p = \infty \), we give a practical algorithm to construct a strong coreset that lets us approximate the maximum distance to any \( k \) dimensional subspace and for general \( p \), we give a polynomial time algorithm that can output a “\( d \)-dimensional” approximate solution.

For \( p \not\in \{1, 2, \infty\} \), much less is known in the streaming setting. In the offline setting, Deshpande and Varadarajan (2007) gave a sampling based algorithm for all \( p \geq 1 \) that outputs a bicriteria solution for the \( \ell_p \) subspace approximation problem. Later Deshpande et al. (2011a) gave a polynomial time \( O(\sqrt{p}) \) factor approximation algorithm for the \( \ell_p \) subspace approximation problem for all \( p \geq 2 \). Assuming the Unique Games Conjecture, they show that it is hard to approximate the cost to a smaller than \( O(\sqrt{p}) \) factor. For \( 1 \leq p \leq 2 \), Clarkson and Woodruff (2015) gave an input sparsity time algorithm that computes a \( 1 + \varepsilon \) approximation but they have an \( \exp(\poly(k/\varepsilon)) \) term in their running time. The \( O(\sqrt{p}) \) factor approximation algorithm of (Deshpande et al., 2011a) is based on convex relaxations and is not applicable in the streaming setting of this paper.

In a recent work, Deshpande and Pratap (2023) observed the lack of streaming algorithms for \( \ell_p \) subspace approximation that also have the subset selection property that our coresets have. They give a subset selection algorithm for the \( \ell_p \) subspace approximation problem but their results have a weaker additive error guarantee. They leave open the subset selection algorithms that give a multiplicative approximation to the \( \ell_p \) subspace approximation problem. In a recent work, Woodruff and Yasuda (2023) answered the question of Deshpande and Pratap (2023) in the affirmative by giving a subset selection algorithm that computes a strong coreset with \( O((k/\varepsilon)^{O(p)} \polylog(n)) \) rows that can approximate the cost of any \( k \)-dimensional space up to a \( 1 \pm \varepsilon \) factor. Selecting \( k^{O(p)} \) rows could be problematic when \( p \) is large.

Our work makes progress on this question by removing the exponential dependence in \( p \), although at the cost of only being able to compute a \( \poly(k, \log nk) \) approximation to the problem.

Relevance to Machine Learning. Our work continues the long line of work in the area of subspace approximation and low rank approximation with different error metrics that has been of interest in the machine learning community. Previous works study problems such as \( \ell_1 \) subspace approximation (Hardt and Moitra, 2013), entrywise \( \ell_p \) low rank approximation (Chierichetti et al., 2017; Dan et al., 2019), column subset selection for the entrywise \( \ell_p \) norm, and other error metrics (Song et al., 2019). Our algorithms for geometric streaming problems such as convex hull estimation have applications to robust classification (Provost and Fawcett, 2001; Fawcett and Niculescu-Mizil, 2007).

2. Preliminaries

For integer \( n \geq 1 \), we use \( [n] \) to denote the set \( \{1, \ldots, n\} \).

For an \( n \times d \) matrix \( A \), we use \( a_i \in \mathbb{R}^d \) to denote the \( i \)-th row. If \( S \subseteq [n] \), then \( A_S \) denotes the submatrix formed by the rows in the set \( S \). Given indices \( i < j \), we use \( A_{i:j} \) to denote the matrix formed by the rows \( a_i, \ldots, a_j \). For \( x \in \mathbb{R}^d \) and \( p \geq 1 \), \( \|x\|_p \) denotes the \( \ell_p \) norm of \( x \) defined as \( \left( \sum_{i=1}^d |x_i|^p \right)^{1/p} \) and \( \|x\|_\infty := \max_i |x_i| \). Given a matrix \( A \), we use \( \|A\|_F \) to denote the Frobenius norm.
and $\|A\|_{p,2}$ to denote the $\ell_p$ norm of the $n$-dimensional vector $(\|a_1\|_2, \ldots, \|a_n\|_2)$. Given a matrix $A$, we use $[A]_k$ to denote the best rank-$k$ approximation of $A$ in Frobenius norm. This can be obtained by truncating the singular value decomposition of $A$ to the top $k$ singular values.

For an arbitrary $k$-dimensional subspace $V \in \mathbb{R}^d$, we use $P_V$ to denote the orthogonal projection matrix onto the subspace $V$, i.e., for any $x \in \mathbb{R}^d$, $P_V \cdot x$ is the closest (in Euclidean norm) vector to $x$ in $V$. So, $d(x, V) = \|(I - P_V)x\|_2$ and $\|A(I - P_V)\|_{\infty,2} = \max_i \|(I - P_V)a_i\|_2 = \max_i d(a_i, V)$.

Due to space constraints, we include all the proofs in the appendix.

### 3. $\ell_\infty$ low rank approximation and Outer Radius

As discussed in the introduction, given a matrix $A$ with rows $a_1, \ldots, a_n$ that arrive in a stream, we want to compute a strong coreset, i.e., a subset $S \subseteq [n]$ such that for all $k$-dimensional subspaces $V$,

$$1 \leq \frac{\max_{i \in [n]} d(a_i, V)}{\max_{i \in S} d(a_i, V)} \leq f$$

for a small distortion $f$. Consider the following simple algorithm: we initialize $S \leftarrow \emptyset$ and stream through the rows $a_1, \ldots, a_n$. When processing the row $a_i$, if there exists a $k$-dimensional subspace $V$ such that $d(a_i, V)^2 > \sum_{i \in S} d(a_i, V)^2$, we update $S \leftarrow S \cup \{i\}$. Otherwise, we proceed to the next row without updating $S$. Consider the set $S$ at the end of the stream and let $V$ be an arbitrary $k$-dimensional subspace. We shall now argue that $A_S$ is a strong coreset with a distortion at most $\sqrt{|S|}$.

Let $V$ be an arbitrary $k$-dimensional subspace of $\mathbb{R}^d$. Let $i^* = \arg \max_i d(a_i, V)$ be the index of the row farthest from $V$. Consider the following cases: if $i^* \in S$, then we have $\max_{i \in [n]} d(a_i, V) = d(a_{i^*}, V) = \max_{i \in S} d(a_i, V)$ and therefore $A_S$ has no distortion for $V$. In case the index $i^* \notin S$, then $d(a_{i^*}, V)^2 \leq \sum_{i \in S, i < i^*} d(a_i, V)^2$ since otherwise we would have added $i^*$ to $S$. Thus,

$$\max_i d(a_i, V) = d(a_{i^*}, V) \leq \sqrt{\sum_{i \in S} d(a_i, V)^2} \leq \sqrt{|S|} \max_{i \in S} d(a_i, V)$$

and therefore $A_S$ is a strong coreset with a distortion at most $\sqrt{|S|}$. Now, if we can show that $S$ can not be too large, we obtain that $A_S$ is a strong coreset with a small distortion.

To show that $S$ is not too large, we appeal to rank-$k$ online ridge leverage scores, a generalization of the so-called ridge leverage scores. In the offline setting, ridge leverage scores have been employed by Cohen et al. (2017) as a suitable modification of the usual $\ell_2$-leverage scores to obtain fast algorithms for $\ell_2$ low rank approximation. Later, Braverman et al. (2020) defined online ridge leverage scores and showed that they can be used to compute low rank approximations in the online model. They also showed that for well-conditioned instances, the sum of the online ridge leverage scores is small. Our main observation is that for the set $S$ constructed as described, the online rank-$k$ ridge leverage score of every row in $A_S$ is large. As the sum of online rank-$k$ ridge leverage scores is not large, we obtain that there cannot be too many rows in $A_S$.

One issue we have to solve to implement this algorithm is given $a_i$ and the set $S$ after processing $a_1, \ldots, a_{i-1}$, how can we efficiently know if there exists a rank-$k$ subspace $V$ such that $d(a_i, V)^2 > \sum_{i \in S} d(a_i, V)^2$? Online ridge leverage scores again come to rescue. We show that if we modify the above described algorithm to instead add $i$ to $S$ when its “online rank-$k$ ridge leverage score” is large with respect to $A_S$, then the set $S$ computed at the end of the process is again a strong coreset with a distortion of at most $\sqrt{|S|}$.

#### 3.1. Online Rank-$k$ Ridge Leverage Scores

Let $A$ be an arbitrary matrix with rows $a_1, \ldots, a_n \in \mathbb{R}^d$ and let $k \leq d$ be a rank parameter. Let $\lambda_i = \|a_i - [a_1, \ldots, a_i]\|_2^2$ be the $i$-th ridge parameter. Note that $\lambda_i = 0$ if and only if rank$(A_{1:i}) \leq k$. We define the “rank-$k$ online ridge leverage score” of the row $a_{i+1}$ to be

$$\tau^{\text{OL},k}_{i+1}(A) = \begin{cases} 1 & \text{if } \lambda_i = 0 \text{ and } a_{i+1} \notin \text{rowspace}(A_{1:i}) \\ \min(1, a_{i+1}^T(A_{1:i}^T A_{1:i} + \lambda_i I)^+ a_{i+1}) & \text{o.w.} \end{cases}$$

The online rank-$k$ ridge leverage scores help us capture the “rank-$k$ information” of the matrix $A$ as the rows are revealed.

#### 3.2. An Efficient Algorithm

Our full coreset construction algorithm is described in Algorithm 1. In the algorithm, we select a subset of rows $S$ online in the following way: a new row $a_t$ is added to the set $S$ if the rank-$k$ online ridge leverage score of the row $a_t$ with respect to the matrix $A_{S,t}$ is at least $1/(1 + 1/k)$.

We will first show that the set $S$ computed by the algorithm defines a matrix $A_S$ that is a strong coreset with a distortion at most $\sqrt{|S|}$. Let $S_t := S \cap [t]$ be the subset of rows that have been selected by the algorithm after processing $a_1, \ldots, a_t$ and let $a_{t+1}$ is the row being processed. We prove the following lemma:

**Lemma 3.1.** Let $t$ be arbitrary and let $S_t := S \cap [t]$ be the subset of rows selected by Algorithm 1 after processing the rows $a_1, \ldots, a_t$. If there exists a rank $k$ subspace $V$ such
Algorithm 1: Minimize Distance to a Subspace

Input: A matrix \( A \) as a stream of rows \( a_1, \ldots, a_n \in \mathbb{R}^d \), a rank parameter \( k \)

Output: A subset \( S \subseteq [n] \)

1. \( S \leftarrow \emptyset, \lambda \leftarrow 0 \) // Algorithm stores \( A_S \)
2. for \( t = 1, \ldots, n \) do
3.     if \( \lambda = 0 \) and \( a_t \notin \text{rowspace}(A_S) \) then
4.         \( S \leftarrow S \cup \{t\} \)
5.     else if \( a_t^T(A_S^TA_S + \lambda I)^+a_t \geq 1/(1 + 1/k) \) then
6.         \( S \leftarrow S \cup \{t\} \)
7.     \( \lambda \leftarrow \|A_S - [A_S]_k\|_F^2/k \) // \( \lambda \) changes only when \( S \) changes
8. return \( S \)

that
\[
d(a_{t+1}, V)^2 \geq \sum_{i \in S_t} d(a_i, V)^2,
\]
then the algorithm adds the row \( t + 1 \) to the set \( S \) that it maintains.

The above lemma now directly implies the following from our earlier discussion:

**Lemma 3.2.** Let \( S \) be the set returned by Algorithm 1 after processing the rows \( a_1, \ldots, a_n \). For any \( k \)-dimensional subspace \( V \),
\[
\max_{i \in S} d(a_i, V) \leq \max_{i \in [n]} d(a_i, V) \leq \sqrt{|S|} \cdot \max_{i \in S} d(a_i, V).
\]

Thus the set \( S \) returned by the algorithm is a strong coreset with a distortion bounded by \( \sqrt{|S|} \). Hence, if we show that \( |S| \) is small, then we obtain the two desired properties of a coreset: (i) the distortion of \( A_S \) is small and (ii) the number of rows in \( A_S \) is small.

To bound the size of the set \( S \), we use the fact that the online rank-\( k \) ridge leverage scores of all the rows in the matrix \( A_S \) with respect to \( A_S \) are at least \( 1/(1 + 1/k) \). Thus, the number of rows in \( A_S \) is at most \( 1 + 1/k \) times the sum of online rank-\( k \) ridge leverage scores of the matrix \( A_S \). We shall now prove a bound on the sum of online rank-\( k \) ridge leverage scores of an arbitrary matrix \( B \). The proof of this lemma is similar to that of proof of Lemma 2.11 of (Braverman et al., 2020). First, we define an “online rank-\( k \) condition number” that we use to bound the sum of online rank-\( k \) ridge leverage scores.

**Definition 3.3.** (Online Rank-\( k \) Condition Number). Given a matrix \( B \) with rows \( b_1, \ldots, b_n \), let \( i^* \) be the largest index \( i \) such that \( \text{rank}(B_{1:i}) = k \). The online rank-\( k \) condition number of \( B \) is defined as
\[
\kappa := \min_{i \leq i^* + 1} \sigma_{\text{min}}(B_{1:i})
\]
where \( \sigma_{\text{min}}(\cdot) \) denotes the smallest non-zero singular value.

**Lemma 3.4.** (Sum of online rank-\( k \) ridge leverage scores). Let \( B \in \mathbb{R}^{n \times d} \) be an arbitrary matrix with an online rank-\( k \) condition number \( \kappa \), then
\[
\sum_{i=1}^n r_i^{\text{OL}, k}(B) = O(k \log(k \cdot \kappa)^2).
\]

Applying the above lemma to the matrix \( A_S \), we obtain that \( |S| = O(k \cdot \log(k \cdot \kappa(A_S)^2)) \). Using the strong coreset property of the matrix \( A_S \), we can show that \( \kappa(A_S) \leq \sqrt{n \cdot \kappa(A)} \), thereby showing that the coreset has a size at most \( |S| = O(k \log(n \cdot \kappa(A)^2)) \) and has a distortion at most \( O(\sqrt{k \log(n \cdot \kappa(A)^2)}) \). This gives the following theorem:

**Theorem 3.5.** Given rows of an arbitrary \( n \times d \) matrix \( A \) with an online rank-\( k \) condition number \( \kappa \), Algorithm 1 selects a subset \( S \) of size \( |S| \leq O(k \log(k \cdot \kappa)^2) \) such that for any \( k \)-dimensional subspace \( V \), we have
\[
1 \leq \max_{i \in [n]} d(a_i, V) \leq C\sqrt{k} \cdot \log(nk)
\]
for a large enough constant \( C \). Additionally, the space required of the algorithm is bounded by the amount of space required to store \( O(|S|) \) rows of \( A \).

If we assume that all the rows of \( A \) lie in a Euclidean ball of radius \( R \) and that we are given some \( \delta < \Delta := \min_n \text{diam}_V \max_i d(a_i, V) \), then we can obtain bounds on \( |S| \) that are independent of \( n \) and only depend on the “aspect ratio” \( R/\delta \). A similar aspect ratio has been used in an earlier work of Makarychev et al. (2022). Let \( t \) be a parameter we fix later. We simply feed the vectors \( (\delta/t)e_1, \ldots, (\delta/t)e_{k+1} \) to Algorithm 1 before processing the vectors \( a_1, \ldots, a_n \). We note that the algorithm is guaranteed to select the vectors \( (\delta/t)e_1, \ldots, (\delta/t)e_{k+1} \) since each of these vectors do not lie in the rowspan of the previous vectors. Let \( S \) denote the subset of rows of \( A \) selected by this algorithm. Using (1), we note that for any \( k \)-dimensional subspace \( V \),
\[
\max_{i \in [n]} d(a_i, V) + \max_{i \in [k+1]} d((\delta/t)e_i, V) \leq \sqrt{k + 1} \frac{\delta}{t} + \sum_{i \in S} d(a_i, V)^2
\]
which implies that
\[
\max_{i \in [n]} d(a_i, V) \leq \sqrt{k + \frac{\delta}{t}} + \sum_{i \in S} d(a_i, V)^2.
\]
We now note that the online rank-\( k \) condition number of the coreset computed by the algorithm must be bounded by \( Rt/\delta \) since the first \( k + 1 \) rows of the coreset are guaranteed to be \( (\delta/t)e_1, \ldots, (\delta/t)e_{k+1} \). Thus, using Lemma 3.4 we
obtain \(|S| = O(k \log(t|S|R/\delta)^2)|\), which implies \(|S| \leq O(k \log(t R/\delta)^3)|\). If we pick \(t = 2\sqrt{k + 1}\), we obtain the following theorem.

**Theorem 3.6.** Given that \(\delta < \max_{k-dim V} \max_{i} d(a_i, V)\) and \(|a_i|_2 < R\), we can compute a subset of rows \(A_S\) of \(A\) such that for any \(k\)-dimensional subspace \(V\),

\[
\max_i d(a_i, V) \leq C \sqrt{k} (\log k R/\delta)^{3/2} \max_{i \in S} d(a_i, V)
\]

and \(|S| = O(k \cdot (\log k R/\delta)^3)|\). The space required of the algorithm is bounded by the amount of space required to store \(|S|\) rows of the matrix \(A\).

A coreset \(S\) of size \(|S|\) and a distortion \(\beta\) can also be used to quickly compute an approximate solution to the \(\ell_\infty\) subspace approximation problem as follows. Let \(V^*\) be the optimal solution for the \(\ell_\infty\) subspace approximation problem on \(A\) and \(\tilde{V}\) denote the top-\(k\) singular subspace of the coreset \(A_S\), which can be computed using the singular value decomposition. Then,

\[
\max_i d(a_i, \tilde{V}) \leq \beta \cdot \max_{i \in S} d(a_i, \tilde{V}) \leq \beta \sqrt{\sum_{i \in S} d(a_i, V^*)^2}.
\]

Since, \(\tilde{V}\) is the top-\(k\) singular subspace of the coreset \(A_S\), we have \(\sqrt{\sum_{i \in S} d(a_i, V^*)^2} \leq \sqrt{\sum_{i \in S} d(a_i, V^*)^2} \) which overall implies

\[
\max_i d(a_i, \tilde{V}) \leq \beta \sqrt{\sum_{i \in S} d(a_i, V^*)^2} \leq \beta \sqrt{|S|} \max_{i \in I} d(a_i, V^*).
\]

Hence, a \(\beta \sqrt{|S|}\) approximation to the \(\ell_\infty\) subspace approximation\(^3\) can be obtained without using any SDP based algorithms from previous works. We can additionally initialize an alternating minimization algorithm on the coreset for \(\ell_\infty\) subspace approximation using the SVD subspace of the coreset and use convex optimization solvers to further improve the quality of the solution. We do note that there are no known bounds on the solution quality attained by the alternating minimization algorithm.

By a simple (lossy) reduction of the outer \((d-k)\) radius estimation problem to computing optimal \(\ell_\infty\) subspace approximation of the matrix \(B = A - a_1\), i.e., the matrix obtained by subtracting \(a_1\) from each row of \(A\), we obtain the following theorem using the coreset bounds in Theorem 3.6.

**Theorem 3.7** (Outer \((d-k)\) radius estimation). Given \(z = a_1 - a_1, \ldots, a_n - a_1\), if a streaming algorithm computes a coreset \(S\) with distortion \(\beta\), then the outer \((d-k)\) radius of the point set \(S\) is an \(O(\beta)\) approximation to the outer \((d-k)\) radius of the entire point set.

\(^3\)In our case, the approximation factor is \(O(k \log nk)^2\).

Given that the online rank-\(k\) condition number of the matrix \(A - a_1\) is \(\kappa\), the outer \((d-k)\) radius of the point set can be approximated up to a \(\sqrt{k \cdot \log nk}\) factor by computing the outer \((d-k)\) radius of the coreset points.

### 3.3. Fast Implementation of Algorithm 1

Note that the set \(S\) and hence the value \(\lambda\) are updated only at most \(O(k \log(n \cdot \kappa)^2)\) times in the stream. Hence, if we compute the singular value decomposition of \(A_S\) each time \(S\) is updated, we only spend at most \(O(d \log(k \log nk))\) time in total. Let \(U \Sigma V^T = A_S\) be the \(\cdot\) singular value decomposition of \(A_S\). Then given any vector \(a\), we can compute \(a^T (A_S^T A_S + \lambda I)^{-1} a\) as \(\|\Sigma^{-1} V^T a\|_2^2 + (1/\lambda) \|(I - VV^T) a\|_2^2 = \|Ma\|_2^2\) where \(M\) is defined as the matrix obtained by concatenating \(\Sigma^{-1} V^T\) and \((1/\sqrt{\lambda})(I - VV^T)\).

Now, if \(G\) is a Gaussian matrix with \(O(\log n)\) rows, we can approximate \(\|Ma\|_2^2\) with \(\|GMa\|_2^2\) up to constant factors for all the future rows \(a_i\). Suppose each time \(S\) is updated, we compute the matrix \(M\) and sample a Gaussian matrix \(G\) and then compute \(GM\) which has \(O(\log n)\) rows. Then the online rank-\(k\) ridge leverage score of any row \(a_i\) that appears in the stream can be approximated as \(\|(GM) a_i\|_2^2\) in time \(O(nnz(a_i) \log n)\), since the matrix \(GM\) has only \(O(\log n)\) rows. Thus the overall algorithm can be implemented in time \(O(nnz(A) \log n + d \cdot poly(k \log nk))\). We implement this algorithm and find that it runs very fast on large datasets.

### 4. Lower Bounds

The algorithm in the previous section uses \(O(dk \log nk)^2\) bits of space to process a stream of \(n\) rows in \(\mathbb{R}^d\) and outputs a strong coreset with a distortion at most \(O(C \sqrt{k \log nk})\), where \(\kappa\) is the condition number. We show that any algorithm that constructs a strong coreset with distortion \(O(\sqrt{k \log n})\) must use \(\Omega(n)\) bits of space. This shows that our algorithm obtains the best possible distortion bounds up to \(\log nk\) factors. Our argument is similar to that of (Woodruff and Yassuda, 2022). We state the lower bound in the following theorem.

**Theorem 4.1.** Given parameters \(n\), \(d\) and \(k\) with \(k = \Omega(\log n)\), any streaming algorithm that computes a strong coreset with distortion at most \(O(\sqrt{k \log n})\) with probability \(\geq 9/10\) must use \(\Omega(n)\) bits of space.

**Proof.** Let \(n\), \(d\) and \(k\) be arbitrary. Let \(a_1, \ldots, a_2n \in \mathbb{R}^d\) be random vectors sampled as follows: each of the first \(k\) entries of each \(a_i\) is set to \(+1\) with equal probability. The remaining \(d - k\) coordinates of each \(a_i\) are set to 0.

Note that \(\|a_i\|_2^2 = k\) for all \(i\). For arbitrary \(i \neq j\), consider \(|\langle a_i, a_j \rangle|\). By Hoeffding’s inequality, with probability \(\geq 1 - \delta\), \(|\langle a_i, a_j \rangle| \leq O(\sqrt{\kappa \log 1/\delta})\). Setting \(\delta = 1/10n^2\) and using a union bound, we obtain that with probability
\[ \geq 9/10, \text{ for all } i \neq j, |\langle a_i, a_j \rangle| \leq O(\sqrt{k \log n}). \]
Condition on this event. Let \( S \subseteq [2n] \), \( |S| = n \) be a uniformly random subset of \([2n]\) of size \( n \).

Consider the stream of vectors \((a_i)_{i \in S}\). Let \( C \) be a randomized algorithm that computes a strong coreset with distortion \( \alpha \leq O(\sqrt{k \log n}) \) with probability \( \geq 9/10 \). Let \( C((a_i)_{i \in S}) \) be the output of the algorithm \( C \) on the stream \((a_i)_{i \in S}\). Condition on the event that \( C((a_i)_{i \in S}) \) is a strong coreset. We now argue that if \( \alpha \) is not too large, we can compute the set \( S \) from the coreset \( C((a_i)_{i \in S}) \).

Given a strong coreset \( M \) with distortion \( \alpha \) for the stream \((a_i)_{i \in S}\), and a rank-\( k \) subspace \( V \), let \( M(V) \) be the value computed using the coreset such that

\[ M(V) \leq \max_{i \in S} d(a_i, V) \leq \alpha \cdot M(V). \]

For each \( i \in [2n] \), consider the subspace \( V_i = \text{span}(e_1, \ldots, e_k) \cap a_i^\perp \), where \( a_i^\perp \) denotes the subspace orthogonal to the vector \( a_i \). We now note the following:

- \( d(a_i, V_i) = ||a_i||_2 = \sqrt{k} \)
- For all \( j \neq i \), \( d(a_j, V_i) = |\langle a_j, a_i \rangle|/||a_i||_2 \leq O(\sqrt{k/\alpha}) \).

Therefore if \( i \in S \), then \( C((a_j)_{j \in S})(V_i) \geq \sqrt{k}/\alpha \) and if \( i \notin S \), then \( C((a_j)_{j \in S})(V_i) \leq O(\sqrt{k/\alpha}) \). If the distortion \( \alpha \leq \sqrt{k/\alpha} \), then by enumerating over all \( V_i \) for \( i \in [2n] \) and computing \( C((a_j)_{j \in S})(V_i) \), we can determine the set \( S \).

Let \( S' \) be the set computed by the enumeration algorithm. If \( |S'| \neq n \), set \( S' \) to \( \{1, 2, \ldots, n\} \). By the above discussion, we have \( \Pr[S' = S] \geq 9/10 \). Note that the entropy of the set \( S \) is \( t = \Omega(n) \) where \( 2^t = \binom{2n}{n} \) is the number of subsets of \([2n]\) of size \( n \).

We now upper bound the conditional entropy \( H(S' \mid S) \). Let \( I \) denote the indicator random variable denoting if the coreset construction algorithm succeeds. Note that given \( I = 1 \), we have \( S = S' \). We have \( H((S, S')) = H(S) + I \cdot H(S \mid S') \) and

\[ H((S, S')) \leq H((S, S', I)) \]

\[ = H(S) + H(I \mid S) + H(S' \mid I, S) \]

and therefore, \( I \cdot H(S \mid S') \leq H(I \mid S) + H(S' \mid I, S) \). Since we assumed that the coreset construction algorithm succeeds with probability \( \geq 9/10 \) given any instance, we have \( H(I \mid S) \leq (9/10) \log_2(10/9) + (1/10) \log_2(10) \leq 1/2 \). Now,

\[ H(S' \mid I, S) = \sum_S \Pr[S = S] \cdot [H(S' \mid S = S, I = 0) \cdot \Pr[I = 0 \mid S = S]

+ H(S' \mid S = S, I = 1) \cdot \Pr[I = 1 \mid S = S]] \]

\[ \leq \sum_S \Pr[S = S] \cdot H(S' \mid S = S, I = 0) \cdot (1/10) \]

where we used the fact that if \( I = 1 \), then \( S' = S \) and therefore \( H(S' \mid S = S, I = 1) = 0 \). Since the output \( S' \) is always a subset of \([2n]\) of size \( n \), we have \( H(S' \mid S = S, I = 0) \leq \log_2(2n) = t \) which then implies \( H(S' \mid I, S) \leq t/10 \). Hence the mutual information \( I(S; S') \geq 9t/10 - 1/2 \) and by the data processing inequality, we have

\[ I(C((a_i)_{i \in S}) \mid S) \geq 9t/10 - 1/2 \]

which implies that the space necessary to store the coreset is \( \Omega(n) \) bits since \( t = \log_2(2n) = \Omega(n) \).

\[ \square \]

5. \( \ell_p \) Subspace Approximation

We now show that our coreset construction algorithm for the \( \ell_\infty \) subspace approximation problem, extends to the \( \ell_p \) subspace approximation problem. Fix a matrix \( A \). For any \( k \)-dimensional subspace \( V \), let \( d_V \) denote the non-negative vector satisfying \( d_V = \text{dist}(a_i, V) = ||a_i^\top (I - P_V)||_2 \). Hence, the \( \ell_p \) subspace approximation problem is to find the rank-\( k \) subspace \( V \) that minimizes \( ||d_V||_p \). We use exponential random variables to embed an \( \ell_p \) low rank approximation problem into an \( \ell_\infty \) low rank approximation problem. We then use the coreset construction algorithm for \( \ell_\infty \) LRA to obtain a coreset for the \( \ell_p \) LRA. First, we have the following lemma about exponential random variables that has been used in various previous works to embed \( \ell_p \) problems into an \( \ell_\infty \) problem.

Lemma 5.1. Let \( e_1, \ldots, e_n \) be independent exponential random variables. Then with probability \( \geq 1 - \delta \), \( \max_i e_i^{-1/p} |x_i| \geq \|x\|_p / (\log 1/\delta)^{1/p} \). We also have that with probability \( \geq 1 - \delta \), \( \max_i e_i^{-1/p} |x_i| \leq \delta^{1/p} \cdot \|x\|_p \).

Given \( n \), define \( D \) to be a random matrix with diagonal entries given by independent copies of the random variable \( e^{-1/p} \). For any fixed rank \( k \) projection matrix \( P \), the above lemma implies that \( \|DA(I - P)\|_{\infty, 2} \geq \|A(I - P)\|_{p, 2}/(\log 1/\delta)^{1/p} \). However, we cannot union bound over the net of all \( k \)-dimensional subspaces of \( \mathbb{R}^d \) since the net can have as many as \( \exp(dk) \) subspaces which leads to a distortion of \( d^{1/p} \), which is prohibitive. Here we crucially use the fact that Algorithm 1 only selects a coreset with \( m = O(k \cdot (\log nk)^2) \) rows. Thus, only those \( k \)-dimensional subspaces spanned by at most \( m \) rows of \( A \) are of interest to us. Now, we can union bound over a net of \( \exp(poly(k, \log nk)) \) subspaces and show the following lemma:

Lemma 5.2. Let \( D \) be an \( n \times n \) diagonal matrix with each diagonal entry being an independent copy of the random variable \( e^{-1/p} \). Fix an \( n \times d \) matrix \( A \). With probability \( \geq 98/100 \), for all \( k \)-dimensional subspaces that are in the
span of at most \( m = O(k \log^2 n \kappa) \) rows of \( A \), we have,

\[
\| D \cdot d_v \|_\infty \geq \frac{\| d_v \|_p}{2^M \log(k+1) + m \log n + km \log nk)^{1/p}}.
\]

If \( V^* \) is the optimal solution for the \( \ell_p \) subspace approximation problem, we can also condition on the event that \( \| D \cdot d_v \|_\infty \leq C \| d_v \|_p \) for a large enough constant \( C \).

We can now argue that if \( S \) is the subset of rows selected by Algorithm 1 when run on the matrix \( DA \), if \( V \) is an approximate solution for the \( \ell_\infty \) subspace approximation problem on the points \( (DA)_S \), then \( V \) is also a good solution for the \( \ell_\infty \) subspace approximation problem of \( A \).

**Theorem 5.3.** Let \( D \) be an \( n \times n \) random matrix with each diagonal entry being an independent copy of \( [e^{-1/p}] \) where \( e \) is a standard exponential random variable. If \( S \) is the subset selected by Algorithm 1 when run on the rows of the matrix \( D \cdot A \) and if \( V \) is a \( \beta \) approximate solution to the problem \( \min_{k\text{-dim } V} \| (D A)_S (I - P_V) \|_{\infty,2} \), then with probability \( \geq 9/10 \),

\[
\frac{\| A(I - P_V) \|_{p,2}}{\min_{k\text{-dim } V} \| A(I - P_V) \|_{p,2}} \leq \beta \cdot O(k^{1/2+2/p} \log^{1+3/p} nk).
\]

**6. Applications to Other Geometric Streaming Problems.**

Given a matrix \( A \), suppose that the rows of \( A \) are close to a \( k \)-dimensional subspace in the following sense: \( \Delta := \min_{k\text{-dim } V} \max_i d(a_i, V) \) is small. We now show that if \( S \) is the subset of rows selected by Algorithm 1, then for any vector \( x \), \( \| Ax \|_\infty \) can be approximated using \( \| ASx \|_\infty \). Fix any unit vector \( x \). Let \( i \) be the index such that \( \| Ax \|_\infty = \| (a_i, x) \|_\infty \). If \( i \in S \), we clearly have \( \| Ax \|_\infty = \| ASx \|_\infty \) and we are done. If \( i \notin S \), we obtain that

\[
\max_x \frac{|(a_i, x)|^2}{\| ASx \|_2^2 + \| ASx - [A_S]_k \|_2^2} \leq \frac{1}{1 + 1/k},
\]

which implies

\[
\| Ax \|_\infty^2 = |(a_i, x)|^2 \leq \| ASx \|_2^2 + \frac{\| ASx - [A_S]_k \|_2^2}{k} \leq \| ASx \|_2^2 + \| ASx - [A_S]_k \|_2^2.
\]

Let \( V^* \) be the optimal solution for rank-\( k \) \( \ell_\infty \) subspace approximation of \( A \). We then have,

\[
\| Ax \|_\infty^2 \leq \| ASx \|_2^2 + \| ASx - [A_S]_k \|_2^2 \leq \| ASx \|_2^2 + \| S \| \Delta^2 / k.
\]

Using \( |S| = O(k \log^2 nk) \), we get the following lemma.

**Lemma 6.1.** If \( S \) is the subset of rows selected by Algorithm 1, for any \( k \)-dimensional subspace \( U \) and any unit vector \( x \),

\[
\frac{\| ASx \|_2}{C \sqrt{k \log n k}} \leq \| Ax \|_\infty \leq \| Ax \|_2 + C \Delta \log nk.
\]

Additionally, as \( \| ASx \|_2 \leq \| S \| \| Ax \|_\infty \), we also have

\[
\| Ax \|_\infty \leq \| Ax \|_2 \leq (C \sqrt{k \log nk}) \| Ax \|_\infty + C \Delta \log nk.
\]

**Width Estimation.** Given a point set \( a_1, \ldots, a_n \in \mathbb{R}^d \), the width of the point set in the direction \( x \in \mathbb{R}^d \) for a unit vector \( x \) is defined as \( w(x) := \max_i \langle a_i, x \rangle - \min_i \langle a_i, x \rangle \). Using a coreset for estimating \( \| Ax \|_\infty \), (Woodruff and Yasuda, 2022) gives an \( O(\sqrt{\Delta \log n}) \) approximation to the width estimation problem. Using Lemma 6.1, we show that we get better approximations when \( \Delta \) is small.

Note that \( w(x) = \max_i \langle a_i - a_1, x \rangle - \min_i \langle a_i - a_1, x \rangle \). Now, \( \max_i \langle a_i - a_1, x \rangle \geq 0, x = 0 \) and \( \min_i \langle a_i - a_1, x \rangle \leq 0, x \leq 0 \) which implies that \( \| (A - a_1)x \|_\infty \leq \| w(x) \| / 2 \). Let \( \kappa' \) be the online rank-\( k \) condition number of \( A - a_1 \). If \( S \) is the subset selected by the algorithm when run on the rows \( v = \langle a_i - a_1 \rangle, i = 0, \ldots, n-1 \), then from Lemma 6.1, we have \( \| (A - a_1)Sx \|_\infty \leq \| (A - a_1)x \|_\infty \leq \| w(x) \| / 2 \). Let \( \kappa' = \||a_1 - a_1 \|_\infty \). Using a coreset for estimating \( \| Ax \|_\infty \), (Woodruff and Yasuda, 2022) show that their coreset can be used to compute an ellipsoid \( E' \) such that \( E' \subseteq K \subseteq O(\sqrt{d \log n}) E' \).

When \( k \ll d \), Algorithm 1 selects \( \ll d \) number of rows and does not have the full \( d \)-dimensional view of the point set and hence can not compute an ellipsoid that satisfies the above multiplicative definition if the points span \( \mathbb{R}^d \). Thus, we consider the set \( K \cap B(0, 1) \) and give an algorithm that computes an unbounded ellipsoid \( E' \) such that \( E' \cap B(0, 1) \subseteq K \cap B(0, 1) \subseteq (\alpha E') \cap B(0, 1) \).

By Lemma 6.1, we have that if \( \| Ax \|_\infty \leq 1 \) and \( \| x \|_2 = 1 \), then \( \| ASx \|_2 \leq C \sqrt{k \log nk} \) and if \( \| ASx \|_2 \leq 1 - C \Delta \log nk \) and \( \| x \|_2 \leq 1 \), then \( \| Ax \|_\infty \leq 1 \). Now assuming \( \Delta < 1/(C \log nk) \), define \( E' = \{ x \mid \| ASx \|_2 \leq 1 - (C \log nk) \Delta \} \).
From the above, we have that if \( x \in E' \cap B(0,1) \),
then \( x \in K \cap B(0,1) \). Additionally if \( x \in K \cap B(0,1) \),
then \( \|A_S x\|_2 \leq C \sqrt{k} \log n \kappa \) and therefore \( x \in E' \cap B(0,1) \). Hence,

\[
E' \cap B(0,1) \subseteq K \cap B(0,1) \subseteq \frac{C \sqrt{k} \log n \kappa}{1 - (C \Delta \log n \kappa)} E' \cap B(0,1).
\]

7. Experiments

We implement our coreset construction algorithm (Algorithm 1) and show that the coreset has a low distortion both for the \( \ell_\infty \) low rank approximation problem and for width estimation.

7.1. \( \ell_\infty \) low rank approximation

We run Algorithm 1 on a synthetic data set and a real world dataset. We construct our synthetic dataset as follows: we pick \( n = 40,000, d = 10,000 \) and \( k = 20 \). We sample an \( n \times k \) random matrix \( L \) and a \( k \times d \) random matrix \( R \) each with i.i.d. uniform random variables drawn from \( \{-100, -99, \ldots, 100\} \). We create an \( n \times d \) matrix

\[
A = L \cdot R + G
\]

where \( G \) is a noise matrix with each entry being an i.i.d. uniform random variable drawn from \( \{-5000, \ldots, 5000\} \). With parameter \( k = 20 \), when Algorithm 1 is run on the matrix \( A \), the coreset \( A_S \) computed by the algorithm has only 28 rows. To measure the quality of the coreset, we consider the following candidate subspaces: we define \( V_i \) to be the at most \( k \)-dimensional subspace formed by the first \( i \) rows of \( R \). These are indeed the subspaces for which the rows of \( A \) have a low distance to. We obtain that

\[
1 \leq \max_{i \in [20]} \frac{\|A(I - P_{V_i})\|_\infty}{\|A(I - P_{V_i})\|_\infty} \leq 1.3433
\]

which shows that the \( \ell_\infty \) cost of the interesting subspaces estimated using the coreset is not too small compared to the actual \( \ell_\infty \) cost of the subspace. Another important requirement is that we do not underestimate the cost of uninteresting subspaces by a lot. To see this, we generate random subspaces of \( k = 20 \) dimensions and observe that

\[
\|A(I - P_V)\|_\infty,2/\|A_S(I - P_V)\|_\infty,2 \leq 1.05
\]

with high probability when \( V \) is drawn at random. This can be explained by the fact that random subspaces are so bad in that \( \|A(I - P_V)\|_\infty,2 \approx \|A\|_\infty,2 \) since a random subspace does not capture a large part of the row of \( A \) with the largest norm. We see that when \( V \) is a random matrix,

\[
\|A(I - P_V)\|_\infty,2/\|A_S(I - P_V)\|_\infty,2 = \|A\|_\infty,2/\|A_S\|_\infty
\]

and since all the rows of \( A \) have similar norms, we get that

\[
\|A(I - P_V)\|_\infty,2/\|A_S(I - P_V)\|_\infty,2 \approx 1.
\]

For the real world dataset, we consider a grayscale image

\[ (Leung, 2017) \]

of dimensions 1836 \times 3264 and treat the image as a matrix \( A \) of the same dimensions. We observe that a rank-150 approximation of the image computed using the SVD is very close to the original image (with some artifacts) and therefore set \( k = 150 \) to be the parameter for which we want to solve the \( \ell_\infty \) low rank approximation problem. We run the coreset construction algorithm on \( A \) and obtain a coreset \( A_S \) with 312 rows. Note that the number of rows in the coreset is \( \approx 17\% \) of the original matrix. Again, to measure the quality of the coreset, we consider subspace \( V_i \) defined to be the top \( i \)-dimensional right singular subspace of \( A \) and measure \( \|A(I - P_{V_i})\|_\infty,2/\|A_S(I - P_{V_i})\|_\infty,2 \leq 1.09 \) and hence the coreset gives very accurate cost estimates for these interesting subspaces. We repeat the same experiment on a different grayscale image

\[ (European Space Agency and NASA, 2006) \]

dimensions 4690 \times 6000 and use \( k = 200 \). We obtain a coreset \( A_S \) with 382 rows and for \( V_i \) defined in the same way as before, \( \max_i \|A(I - P_{V_i})\|_\infty,2/\|A_S(I - P_{V_i})\|_\infty,2 \leq 1.12 \).

7.2. Width Estimation

Towards width estimation, Lemma 6.1 shows that if \( A_S \) is the coreset computed by Algorithm 1, then for any unit vector, \( \|Ax\|_\infty \) can be approximated up to a multiplicative/additive error. We again consider synthetic/real-world datasets and use linear programs to obtain an upper bound on \( \|Ax\|_\infty/\|A_Sx\|_\infty \) for \( x \in \text{rowspace}(A_S) \). We note that when the rows of \( A \) are close to a \( k \)-dimensional subspace, then \( A_S \) computed using Algorithm 1 spans a subspace close to this \( k \)-dimensional subspace by Theorem 3.5. Hence, all the important directions are already in \( \text{rowspace}(A_S) \) and bounding \( \|Ax\|_\infty/\|A_Sx\|_\infty \) for \( x \in \text{rowspace}(A_S) \) verifies that the distortion in the important directions is not large.

We construct a synthetic dataset \( A = L \cdot R + G \) in a similar way to the previous section with \( n = 40,000, d = 10,000 \) and \( k = 20 \). To avoid numerical issues when solving linear programs, we now choose the coefficients of the matrices \( L \) and \( R \) to be i.i.d. uniform random variables drawn from \( \{-10, \ldots, 10\} \) and the coefficients of \( G \) to be i.i.d. uniform random variables drawn from \( \{-50, \ldots, 50\} \). The coreset \( A_S \) constructed by Algorithm 1 for the matrix \( A \) has 29 rows and by solving \( n \) linear programs, we find that

\[
\max_{x \in \text{rowspace}(A_S)} \|Ax\|_\infty/\|A_Sx\|_\infty \leq 4.8
\]

We also perform the same experiment on the images from the previous section and find that

\[
\|Ax\|_\infty/\|A_Sx\|_\infty \leq 1.055 \text{ for all } x \in \text{rowspace}(A_S) \text{ for the first image and } \|Ax\|_\infty/\|A_Sx\|_\infty \leq 1.03 \text{ for all } x \in \text{rowspace}(A_S) \text{ for the second image.}
\]

For real-world datasets, the coreset computed is very accurate in approximating \( \|Ax\|_\infty \) for all the interesting directions \( x \). This can be explained by the fact that the value of \( k \) we picked is large and the noise at that value of \( k \) is small enough that many directions are covered by the coreset and hence the coreset has a small error.
Acknowledgments

Part of this work was done while Praneeth Kacham and David P. Woodruff were visiting Google Research. Praneeth and David were also supported in part by a Simons Investigator Award and NSF Grant No. CCF-2335412.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning and Theoretical Computer Science. We do not foresee any potential societal consequences of our work which we feel must be specifically highlighted here.

References


Enoch Leung. Black and White Chessboard, 2017. URL https://commons.wikimedia.org/wiki/File:Black_and_White_Chessboard.jpg. This work is licensed under Creative Commons Attribution-Share Alike 4.0 International license.


A. Missing Proofs from Section 3

A.1. Proof of Lemma 3.1

Assume that there is a $k$-dimensional subspace $V$ such that $d(a_{t+1}, V)^2 > \sum_{i \in S_t} d(a_i, V)^2$ where $S_t = S \cap [t]$ is the set of rows selected by the algorithm after processing the rows $a_1, \ldots, a_t$.

If $\sum_{i \in S_t} d(a_i, V)^2 = 0$, then $\text{rank}(A_{S_t}) \leq k$ and $\text{rowspan}(A_{S_t}) \subseteq V$. Since $d(a_{t+1}, V) > 0$, we have $a_{t+1} \notin V$ which implies $a_{t+1} \notin \text{rowspan}(A_{S_t})$ and therefore the algorithm adds $t+1$ to the set $S$.

Now, suppose $\sum_{i \in S_t} d(a_i, V)^2 > 0$. Let $P_V$ be the orthogonal projection matrix onto the subspace $V$ and define

$$x^* := \frac{(I - P_V)a_{t+1}}{\| (I - P_V)a_{t+1} \|_2}.$$

Using the fact that $(I - P_V)$ is also a projection matrix, we obtain

$$|(a_{t+1}, x^*)|^2 = \frac{(a_{t+1}^T(I - P_V)a_{t+1})^2}{\| (I - P_V)a_{t+1} \|_2^2} = \frac{\| (I - P_V)a_{t+1} \|_2^4}{\| (I - P_V)a_{t+1} \|_2^2} = \| (I - P_V)a_{t+1} \|_2^2 = d(a_{t+1}, V)^2.$$

We also have

$$\| A_{S_t} x^* \|_2^2 = \frac{\| A_{S_t}(I - P_V)a_{t+1} \|_2^2}{\| (I - P_V)a_{t+1} \|_2^2} \leq \frac{\| A_{S_t}(I - P_V) \|_2^2 \| (I - P_V)a_{t+1} \|_2^2}{\| (I - P_V)a_{t+1} \|_2^2} \leq \| A_{S_t}(I - P_V) \|_F^2 = \sum_{i \in S_t} d(a_i, V)^2.$$

Additionally, when processing the row $a_{t+1}$, the value of $\lambda$ used by the algorithm is $\| A_{S_t} - [A_{S_t}]_k \|_F^2/k < \| A_{S_t} (I - P_V) \|_F^2/k$ since the subspace $V$ has a dimension $k$. Now, we consider two cases:

- **Case 1:** $\lambda = 0$. In this case, we have $\text{rank}(A_{S_t}) \leq k$. There are again two cases. If $a_{t+1} \notin \text{rowspan}(A_{S_t})$, then the algorithm adds $t+1$ to the set $S$ and we are done.

  If $a_{t+1} \in \text{rowspan}(A_{S_t})$, then we can write $a_{t+1} = (A_{S_t})^T z$ for some $z$. If $A_{S_t} x^* = 0$, then we get $(x^*, a_{t+1}) = (x^*)^T (A_{S_t})^T z = (z, A_{S_t} x^*) = 0$ which contradicts our assumption that $|(a_{t+1}, x^*)|^2 = d(a_{t+1}, V)^2 > \sum_{i \in S_t} d(a_i, V)^2 > 0$. Thus, $A_{S_t} x^* \neq 0$ and therefore

$$\frac{|(a_{t+1}, x^*)|^2}{\| A_{S_t} x^* \|_2^2} \geq \frac{d(a_{t+1}, V)^2}{\sum_{i \in S_t} d(a_i, V)^2} > 1.$$

Finally, since $a_{t+1} \in \text{rowspan}(A_{S_t})$, we obtain $a_{t+1}^T (A_{S_t}^T A_{S_t} + \lambda I)^+ a_{t+1} > 1/(1 + 1/k)$ and therefore the algorithm adds $t+1$ to the set $S$ and we are done.

- **Case 2:** $\lambda \neq 0$. In this case, we have $\text{rank}(A_{S_t}) > k$ and therefore $\sum_{i \in S_t} d(a_i, V)^2 > 0$. Now,

$$\frac{|(a_{t+1}, x^*)|^2}{\| A_{S_t} x^* \|_2^2 + \lambda \| x^* \|_2^2} \geq \frac{d(a_{t+1}, V)^2}{\sum_{i \in S_t} d(a_i, V)^2 + \lambda} \geq \frac{\sum_{i \in S_t} d(a_i, V)^2}{\sum_{i \in S_t} d(a_i, V)^2 + \sum_{i \in S_t} d(a_i, V)^2/k} = \frac{1}{1 + 1/k}.$$

From the above inequality, we obtain $(a_{t+1})^T ((A_{S_t})^T A_{S_t} + \lambda I)^+ a_{t+1} > 1/(1 + 1/k)$ and therefore the algorithm adds $t+1$ to the set $S$ and we are done.

A.2. Proof of Lemma 3.4

Let $i^*$ be the largest index such that $\text{rank}(B_{1:i^*}) = k$. We note $\text{rank}(B_{1:i^*+1}) = k + 1$. We now separate the sum of online rank-$k$ ridge leverage scores as

$$\sum_{i=1}^n \tau_i^{OL,k}(B) = \sum_{i=1}^{i^*} \tau_i^{OL,k}(B) + \sum_{i=i^*+2}^n \tau_i^{OL,k}(B).$$
and bound both the terms separately. Let \( R_i^4 \subseteq [i^* + 1] \) be the set of coordinates \( i \) such that \( \text{rank}(B_{1:i}) > \text{rank}(B_{1:i+1}) \).

Note that \( |R| \leq k + 1 \). By definition of the rank-\( k \) ridge leverage scores, we have for all \( i \in R, \tau_i^{OL,k}(B) = 1 \). Now consider an \( i < i^* + 1 \) and \( i \notin R \). We have

\[
\tau_i^{OL,k}(B) = \min(1, b_i^\top ((B_{1:i-1})^\top B_{1:i-1})^+ b_i).
\]

We define \( \sigma_{\min,RI} := \min_{i \in RI} \sigma_{\min}(B_{1:i}) \) where \( \sigma_{\min}(\cdot) \) is used to denote the smallest nonzero singular value of the matrix \( B \). We note that for all \( i \in R, \|b_i\|_2 \geq \sigma_{\min,RI} \).

Now consider \( i < i^* + 1 \) and \( i \notin R \). Note that \( b_i \in \text{rowspace}(B_{1:i-1}) \).

Claim A.1. For \( \sigma_{\min,RI} \) defined as above, the following hold:

1. \[
b_i^\top ((B_{1:i-1})^\top (B_{1:i-1}))^+ b_i \leq 2 \cdot b_i^\top ((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I)^+ b_i.
\]

2. \[
\tau_i^{OL,k}(B) = \min(1, b_i^\top ((B_{1:i-1})^\top (B_{1:i-1}))^+ b_i) \leq 2 \cdot \min(1, b_i^\top ((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I)^+ b_i).
\]

Proof. Let \( \Sigma V^\top \) be the “thin” singular value decomposition of the matrix \( B_{1:i-1} \). It is easy to see that \( \sigma_{\min}(B_{1:i}) \geq \sigma_{\min,RI} \). Since \( i \notin R \), we have \( b_i \in \text{rowspace}(B_{1:i-1}) \) and therefore we can write \( b_i = V \cdot z \) for some \( z \) which implies

\[
b_i^\top ((B_{1:i-1})^\top (B_{1:i-1}))^+ b_i = z^\top \Sigma^{-2} z^\top.
\]

We can also write

\[
((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I)^+ = V(\Sigma^2 + \sigma_{\min,RI}^2 \cdot I)^{-1} V^\top + \frac{1}{\sigma_{\min,RI}^2}(I - VV^\top)
\]

from which we obtain

\[
b_i^\top ((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I)^+ b_i = z^\top (\Sigma^2 + \sigma_{\min,RI}^2 \cdot I)^{-1} z \geq \frac{1}{2} \cdot z^\top \Sigma^{-2} z = \frac{1}{2} b_i^\top ((B_{1:i-1})^\top (B_{1:i-1}))^+ b_i,
\]

where the last inequality follows from the fact that \( 0 \prec \Sigma^2 + \sigma_{\min,RI}^2 \cdot I \preceq 2 \cdot \Sigma^2 \).

Note that the second claim directly follows from the first. \( \square \)

For \( i \in R \), we prove the following:

Claim A.2. For all \( i \in R, \)

\[
1 = \tau_i^{OL,k}(B) \leq b_i^\top ((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I)^+ b_i.
\]

Proof. Let \( b_i^\perp \) be the projection of \( b_i \) away from \( \text{rowspace}(B_{1:i-1}) \). Note that \( b_i^\perp \) is in the rowspace of \( B_{1:i} \) and therefore

\[
|\langle b_i, b_i^\perp \rangle| = \| (B_{1:i}) \cdot b_i^\perp \|_2 \geq \sigma_{\min,RI} \cdot \| b_i^\perp \|_2
\]

which implies

\[
\frac{|\langle b_i, b_i^\perp \rangle|^2}{\| B_{1:i-1} \cdot b_i^\perp \|_2^2 + \sigma_{\min,RI}^2 \| b_i^\perp \|_2^2} \geq \frac{\sigma_{\min,RI}^2 \| b_i^\perp \|_2^2}{0 + \sigma_{\min,RI}^2 \| b_i^\perp \|_2^2} \geq 1.
\]
Thus, for all $i < i^* + 1$, we have

$$\tau_{i}^\text{OL}(B) \leq 2 \cdot \min(1, b_i^\top ((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I) + b_i).$$

Hence, it suffices to bound $\sum_{i=1}^{i^*+1} \min(1, b_i^\top ((B_{1:i-1})^\top B_{1:i-1} + \sigma_{\min,RI}^2 \cdot I) + b_i)$. By Theorem 2.2 of (Cohen et al., 2016), we can bound this quantity by $O(k \log \|B_{1:i^*+1}||_2/\sigma_{\min,RI})$. Hence,

$$\sum_{i=1}^{i^*+1} \tau_{i}^\text{OL}(B) = O \left( k \log \|B_{1:i^*+1}\|_2/\sigma_{\min,RI} \right).$$

We now want to bound

$$\sum_{i=i^*+2}^{n} \tau_{i}^\text{OL}(B) = \sum_{i=i^*+2}^{n} \min(1, b_i^\top (B_{1:i-1}^\top B_{1:i-1} + \|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2 \cdot I)^{-1} b_i).$$

Braverman et al. (2020) show a bound on the $\sum_{i=1}^{n} \min(1, b_i^\top (B_{1:i-1}^\top B_{1:i-1} + \lambda I)^{-1} b_i)$ where $\lambda = \|B - [B]_k\|_F^2/k$. The only difference in the above term we want to bound is that, instead of using a fixed $\lambda$ for all the terms as in (Braverman et al., 2020), we require an upper bound when each term has a different multiple of the identity matrix.

We will now state some useful facts, that let us use the upper bounds from (Braverman et al., 2020) to bound the term in (3). Suppose $\alpha$ is such that $\alpha/2 \leq \|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2/k \leq \alpha$. Then, we have from the standard properties of the Löwner ordering that,

$$\frac{1}{2}B_{1:i-1}B_{1:i-1}^\top + \frac{\alpha}{2} \cdot I \preceq B_{1:i-1}B_{1:i-1}^\top + \frac{\alpha}{2} \cdot I \preceq B_{1:i-1}B_{1:i-1}^\top + \frac{\|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2}{k} \leq B_{1:i-1}B_{1:i-1}^\top + \alpha \cdot I.$$

Since all the above matrices are positive definite, assuming $\alpha > 0$, we obtain that

$$2 \cdot (B_{1:i-1}B_{1:i-1}^\top + \alpha \cdot I)^{-1} \preceq (B_{1:i-1}B_{1:i-1}^\top + \frac{\|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2}{k} \cdot I)^{-1} \preceq (B_{1:i-1}B_{1:i-1}^\top + \alpha \cdot I)^{-1}$$

and therefore,

$$\min(1, b_i^\top (B_{1:i-1}B_{1:i-1}^\top + \frac{\|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2}{k} \cdot I)^{-1} b_i) \leq 2 \cdot \min(1, b_i^\top (B_{1:i-1}B_{1:i-1}^\top + \alpha \cdot I)^{-1} b_i).$$

We note that $\|B_{1:i^*+1} - [B_{1:i^*+1}]_k\|_F^2 = \sigma_{\min}(B_{1:i^*+1})^2 \geq \sigma_{\min,RI}^2$ where we used the fact that the rank of $B_{1:i^*+1}$ is exactly $k + 1$. For $j = 1, \ldots$, let $i_j$ be the largest $i$ such that

$$\frac{\|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2}{k} \leq 2^j \cdot \frac{\sigma_{\min,RI}^2}{k},$$

and consider the intervals of integers, $(k + 1 = i_0, i_1, (i_1, i_2), (i_2, i_3), \ldots)$. We note that there are at most

$$O \left( \log \frac{\|B - [B]_k\|_F^2}{\sigma_{\min,RI}^2} \right) = O \left( \log \frac{\|B\|_2}{\sigma_{\min,RI}} \right),$$

such non-empty intervals. Now consider an arbitrary interval $(i_j, i_{j+1})$ and we will bound

$$\sum_{i \in (i_j, i_{j+1})} \min(1, b_i^\top (B_{1:i-1}B_{1:i-1}^\top + \frac{\|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2}{k} \cdot I)^{-1} b_i).$$

Setting $\alpha = 2^{j+1}\sigma_{\min,RI}^2/k$ in (4), we get

$$\sum_{i \in (i_j, i_{j+1})} \min(1, b_i^\top (B_{1:i-1}B_{1:i-1}^\top + \frac{\|B_{1:i-1} - [B_{1:i-1}]_k\|_F^2}{k} \cdot I)^{-1} b_i) \leq 2 \cdot \sum_{i \in (i_j, i_{j+1})} \min(1, b_i^\top (B_{1:i-1}B_{1:i-1}^\top + 2^{j+1}\sigma_{\min,RI}^2/k \cdot I)^{-1} b_i)$$

$$\leq 2 \cdot \sum_{i \in (i_j, i_{j+1})} \min(1, b_i^\top (B_{1:i-1}B_{1:i-1}^\top + 2^{j+1}\sigma_{\min,RI}^2/k \cdot I)^{-1} b_i)$$

14
and since by definition \( \|B_{1:i,j+1-1} - [B_{1:i,j+1-1}]k\|_2^2 \leq 2^{j+1} \frac{\sigma_{\text{min,R}}^2}{k} \), we further obtain
\[
\sum_{i \in (i_1,i_2,j+1)} \min(1,b_i^T(B_{1:i-1}^TB_{1:i-1} + \frac{\|B_{1:i-1} - [B_{1:i-1}]k\|_F^2}{k} \cdot I)^{-1}b_i) \\
\leq \sum_{i \in (i_1,i_2,j+1)} \min(1,m_i^T(B_{1:i-1}^TB_{1:i-1} + \frac{\|B_{1:i,j+1-1} - [B_{1:i,j+1-1}]k\|_F^2}{k} \cdot I)^{-1}m_i).
\]

We can then finally use Lemma 2.11 of (Braverman et al., 2020) to bound the above term by
\[
k \log \left( 1 + \frac{k\|B_{1:i,j+1-1}\|_2^2}{\|B_{1:i,j+1-1} - [B_{1:i,j+1-1}]k\|_F^2} \right) + k + 1 \leq k \log(1 + k\|B\|_2^2/\sigma_{\text{min,R}}^2) + k + 1
\]
where we used the facts that \(\|B_{1:i,j+1-1} - [B_{1:i,j+1-1}]k\|_2^2 \geq \|B_{1:i} - [B_{1:i}]k\|_F^2 \geq \sigma_{\text{min,R}}^2\) and \(\|B_{1:i,j+1} - [B_{1:i,j+1}]k\|_2^2 \leq \|B\|_2^2\).

Overall, we get that
\[
O(k \log(1 + k\|B\|_2^2/\sigma_{\text{min,R}}^2))^2 = O(k \log(k \cdot \kappa)^2).
\]

A.3. Proof of Theorem 3.7

**Proof.** If \(V\) is a \(k\)-dimensional subspace and \(c\) is arbitrary, then the set \(V + c\) is defined as a \(k\)-dimensional flat. Recall that the outer \(d-k\) radius of a point set \(\{a_1, \ldots, a_n\} \subseteq \mathbb{R}^d\) is defined as
\[
\min_{k\text{-dim flat } F} \max_{i} d(a_i,F).
\]
Using the fact that flats are translations of \(k\) dimensional subspaces, we equivalently have that the outer \(d-k\) radius is equal to
\[
\min_{k\text{-dim subspace } V} \min_{c \in \mathbb{R}^d} \max_{i} d(a_i,c,V) = \min_{k\text{-dim subspace } V} \min_{c} \|(A - c)(I - P_V)\|_{\infty,2}.
\]
Here we abuse the notation and use \(A - c\) to denote the matrix with rows given by \(a_i - c\) for \(i \in [n]\). Now define a matrix \(B = A - a_1\) with \(n\) rows given by \(0 = a_1 - a_1, a_2 - a_1, a_3 - a_2, \ldots, a_n - a_1\). For any \(k\)-dimensional subspace \(V\) and any \(c \in \mathbb{R}^d\), we have
\[
\|B(I - P_V)\|_{\infty,2} = \|(A - a_1)(I - P_V)\|_{\infty,2} = \|(A - c + c - a_1)(I - P_V)\|_{\infty,2} \\
\leq \|(A - c)(I - P_V)\|_{\infty,2} + \|(I - P_V)(a_1 - c)\|_2 \\
\leq 2\|(A - c)(I - P_V)\|_{\infty,2}.
\]
Hence, \(\|B(I - P_V)\|_{\infty,2} \leq 2\min_{c} \|(A - c)(I - P_V)\|_{\infty,2}\). We also have \(\|B(I - P_V)\|_{\infty,2} = \|(A - a_1)(I - P_V)\|_{\infty,2} \geq \min_{c} \|(A - c)(I - P_V)\|_{\infty,2}\). Thus, \(\min_{V} \|B(I - P_V)\|_{\infty,2}\) is a 2-approximation for \(\min_{k\text{-dim flat } F} \max_{i} d(a_i,F)\) and if \(S\) is the set of rows selected by Algorithm 1 when run on the rows of the matrix \(B = A - a_1\), then
\[
\min_{V} \|B_S(I - P_V)\|_{\infty,2}\]
is a \(O(\sqrt{k \log(n \kappa)})\) approximation for outer \((d-k)\)-radius estimation of the point set \(\{a_1, \ldots, a_n\}\) where \(\kappa\) is the online rank-\(k\) condition number of \(A - a_1\).

B. Missing Proofs from Section 5

B.1. Proof of Lemma 5.1

**Proof.** By min-stability of exponential random variables, we have that the distribution of \(\max_i e^{-1/|x_i|}p\) is the same as the distribution of \(e^{-1/|x|}p\) where \(e\) is also a standard exponential random variable. With probability \(\geq 1 - \delta\), we have \(e \leq \log 1/\delta\). And hence we have that with probability \(\geq 1 - \delta\),
\[
\max_i e_i^{-1/p} |x_i| = (\max_i e_i^{-1/|x_i|}p)^{1/p} \geq \frac{|x|_p}{(\log 1/\delta)^{1/p}}.
\]
With probability \(\geq 1 - \delta\), we also have that \(e \geq \delta\) which implies that with probability \(\geq 1 - \delta\), \(\max_i e_i^{-1/p} |x_i| = (\max_i e_i^{-1/|x_i|}p)^{1/p} \leq |x|_p\delta^{-1/p}\).
B.2. Proof of Lemma 5.2

Proof. Let $S$ be an arbitrary set of $m \leq K$ rows of $A$ and let $V_S := \text{rowspace}(A_S)$. Let $N_S$ be a $\gamma$ net for the set $V_S \cap S^{d-1}$ i.e., the set of vectors in the subspace $V_S$ with euclidean norm 1. As the subspace $V_S$ has dimension at most $m$, we have that there is a set $N_S$ with size at most $\exp(O(m \log 1/\gamma))$. Let $V$ be an arbitrary $k$ dimensional subspace of $V_S$ and let $\{v_1, \ldots, v_k\}$ be an orthonormal basis for $V$.

Let $\hat{V}$ be the subspace spanned by $\{\hat{v}_i, \ldots, \hat{v}_k\}$, where $\hat{v}_i \in N_S$ and $\|v_i - \hat{v}_i\|_2 < \gamma$ for all $i \in [n]$. Let $a$ be an arbitrary vector. By abusing the notation let $V$ (resp. $\hat{V}$) also denote the matrix with $v_1, \ldots, v_k$ (resp. $\hat{v}_1, \ldots, \hat{v}_k$) as columns. We have

$$d(a, V) = \|a - VV^T a\|_2 \quad \text{and} \quad d(a, \hat{V}) = \|a - \hat{V}\hat{V}^+ a\|_2$$

and therefore $|d(a, V) - d(a, \hat{V})| \leq \|V\hat{V}^+ - VV^+\|_2 \|a\|_2$. If $\gamma \leq 1/4\sqrt{k}$, we can show that $\|VV^T - \hat{V}\hat{V}^+\|_2 \leq 4\sqrt{k}\gamma$ and therefore have that for any $a$, $|d(a, V) - d(a, \hat{V})| \leq \sqrt{k}\gamma \|a\|_2$. Hence,

$$\|d_V - d_{\hat{V}}\|_\infty \leq \max_i |d(a_i, V) - d(a_i, \hat{V})| \leq 4\sqrt{k}\gamma \max \|a_i\|_2 = 4\sqrt{k}\gamma \|A\|_{\infty,2}.$$ 

Overall, this implies that for any arbitrary $k$ dimensional subspace $V$ in the span of rows of $A_S$, there is a $k$ dimensional subspace $\hat{V}$ spanned by some $k$ vectors in the net $N_S$ satisfying

$$\|d_V - d_{\hat{V}}\|_\infty \leq 4\sqrt{k}\gamma \|A\|_{\infty,2}.$$ 

As $d_V \in \mathbb{R}^n$, we have $\|d_V - d_{\hat{V}}\|_p \leq n^{1/p} \|d_V - d_{\hat{V}}\|_\infty \leq 4\sqrt{k}\gamma n^{1/p} \|A\|_{\infty,2}$. Now, let

$$V_S := \{\hat{V} = \text{span}(\hat{v}_1, \ldots, \hat{v}_k) \mid \hat{v}_i \in N_S\}.$$ 

We have $|V_S| \leq |N_S|^k \leq \exp(O(km \log 1/\gamma))$ since $|N_S| \leq \exp(O(m \log 1/\gamma))$. As there are $\binom{n}{m}$ choices for $S$, the total number of subspaces in the set $\bigcup_{S \in \binom{[n]}{m}} V_S$ is upper bounded by $\exp(m \log n + km \log 1/\gamma)$. Using Lemma 5.1, using a union bound over all $\exp(m \log n + km \log 1/\gamma)$ choices of $\hat{V}$, we have that with probability $\geq 99/100$, for all $\hat{V} \in \bigcup_{S \in \binom{[n]}{m}} V_S$,

$$\|D \cdot d_{V}\|_\infty \geq \frac{\|d_{V}\|_p}{(\log 100 + m \log n + km \log 1/\gamma)^{1/p}}.$$ 

Using Lemma 5.1 again, we also have that $\max_i \|D_i\| \leq C_3 n^{1/p}$ for a large enough constant $C_3$ with probability $\geq 99/100$. Condition on both these events. We have that for any $k$ dimensional subspace $V$ in the span of any set of $m$ rows of $A$,

$$\|D \cdot d_V\|_\infty \geq \frac{\|d_V\|_p}{(\log 100 + m \log n + km \log 1/\gamma)^{1/p}} - C_3 n^{1/p} \|d_V - d_{\hat{V}}\|_\infty \geq \frac{\|d_V\|_p}{(\log 100 + m \log n + km \log 1/\gamma)^{1/p}} - \frac{4\sqrt{k}\gamma \|A\|_{\infty,2}}{(\log 100 + m \log n + km \log 1/\gamma)^{1/p}}.$$ 

For any $V$, we have that $\|d_V\|_p \geq \|d_V\|_2 / \sqrt{n} \geq \|A - [A]_k\|_F / \sqrt{n}$ using the fact that $V$ is a $k$ dimensional subspace. Hence, if $\gamma \leq \text{poly}(\|A - [A]_k\|_F / \|A\|_{\infty,2}, 1/n)$, then

$$\|D \cdot d_V\|_\infty \geq \frac{\|d_{V}\|_p}{2(\log 100 + m \log n + km \log 1/\gamma)^{1/p}}.$$ 

Now, $\gamma$ can be taken as $\text{poly}(1/(nk))$ so that

$$\|D \cdot d_V\|_\infty \geq \frac{\|d_{V}\|_p}{C(\log 100 + m \log n + km \log (nk))^{1/p}}$$

for all subspaces $V$ that are in the span of any subset of $m$ rows of $A$. □
B.3. Wrap-up

Let

\[ V^* = \arg \min_{k\text{-dim subspaces } V} \|d_V\|_p. \]

Condition on the event that \( \|D^{1/p} \cdot d_V\|_{\infty} \leq C_1 \|d_V\|_p \) for a large enough constant \( C_1 \). The event holds with probability \( \geq 99/100 \) by Lemma 5.1. Finally, by a union bound, we have all the following events hold simultaneously with probability \( \geq 9/10 \):

1. Algorithm 1, when run on the rows of the matrix \( D \cdot A \), selects at most \( m = O(k \cdot (\log n\kappa)^2) \) rows.
2. For any \( k \) dimensional subspace \( V \) contained in the span of any at most \( m \) rows of \( A \),

\[ \|D \cdot d_V\|_{\infty} \geq \frac{\|d_V\|_p}{C_2 k^{2/p} \log^{3/p} n\kappa}. \]
3. If \( V^* \) is the optimal subspace that minimizes the \( \ell_p \) norm of the distance vector to a \( k \) dimensional subspace, then

\[ \|D \cdot d_{V^*}\|_{\infty} \leq C_1 \|d_{V^*}\|_p. \]

Conditioned on the above events, let \( S \subseteq [n] \) be the coreset computed for the matrix \( D \cdot A \) by Algorithm 1. From Theorem 3.5, we have that for any rank \( k \) projection matrix \( P \),

\[ \|(DA)_S(I - P)\|_{\infty,2} \leq \|(DA)(I - P)\|_{\infty,2} \leq C\sqrt{\kappa}(\log n\kappa)\|(DA)_S(I - P)\|_{\infty,2}. \]

Let \( \hat{V} \) be a \( k \) dimensional subspace such that

\[ \min_{k\text{-dim } V} \|(DA)_S(I - P_V)\|_{\infty,2} \beta \cdot \min_{k\text{-dim } V} \|(DA)_S(I - P_V)\|_{\infty,2}. \]

Without loss of generality, we can assume that \( \hat{V} \) is contained in the rowspace of \( (D \cdot A)_S \) and hence the row space of \( A_S \). Therefore,

\[ \|A(I - P_{\hat{V}})\|_{p,2} = \|d_{\hat{V}}\|_p \]
\[ \leq C_2 k^{2/p} \log^{3/p}(n\kappa)\|D \cdot d_{\hat{V}}\|_{\infty} \]
\[ = C_2 k^{2/p} \log^{3/p}(n\kappa)\|(DA)(I - P_{\hat{V}})\|_{\infty,2} \]
\[ \leq C_2 \cdot C \cdot k^{2/p+1/2} \log^{1+3/p}(n\kappa)\|(DA)_S(I - P_{\hat{V}})\|_{\infty,2} \]
\[ \leq \beta \cdot C_2 \cdot C \cdot k^{2/p+1/2} \log^{1+3/p}(n\kappa)\|D \cdot d_{V^*}\|_{\infty,2} \]
\[ \leq \beta \cdot C_1 \cdot C_2 \cdot C \cdot k^{2/p+1/2} \log^{1+3/p}(n\kappa)\|d_{V^*}\|_{p,2}. \]

Thus, \( \hat{V} \) is an \( O(\beta \cdot k^{2/p+1/2} \log^{1+3/p}(n\kappa)) \) approximate solution for the \( \ell_p \) low rank approximation problem over the matrix \( A \).

Algorithm 2: Minimize the \( \ell_p \) norm of the vector of distances to a \( k \) dimensional subspace

**Input:** A matrix \( A \) as a stream of rows \( a_1, \ldots, a_n \in \mathbb{R}^d \), \( p \geq 1 \) and a rank parameter \( k 

**Output:** Rank \( k \) projection matrix \( P \)

9 Feed the stream \( e_i^{-1/p} a_1, \ldots, e_i^{-1/p} a_n \) to Algorithm 1 and obtain the set \( S \subseteq [n] \) \( \hat{V} \leftarrow \beta \)-approximate solution for \( \min_{\text{rank-}k, \text{max}_{i \in S} d(e_i^{-1/p} \cdot a_i, V)} \) return \( \hat{V} \)

**Theorem B.1.** Given a stream of rows \( a_1, \ldots, a_n \), Algorithm 2 uses space necessary to store \( O(k \log^2 n\kappa) \) rows and outputs a rank \( k \) subspace \( \hat{V} \) satisfying

\[ \|A(I - P_{\hat{V}})\|_{p,2} \leq O(\beta \cdot k^{1/2+2/p} \log^{1+3/p}(n\kappa)) \min_{k\text{-dim } V} \|A(I - P_V)\|_{p,2}. \]
C. Missing Details about Experiments

C.1. Measuring Distortion with in the Subspace

Given a matrix $A$ and a parameter $k$, Algorithm 1 returns a coreset $S$. In our experiments we measure the maximum distortion defined as $\max_{x \in \text{rowspace}(A_S)} \|Ax\|_\infty / \|A_Sx\|_\infty$. Since any vector in the rowspace of $A_S$ can be written as $A_S^Ty$ for some $y$, we want to measure $\max_y \|AA_S^Ty\|_\infty / \|A_S A_S^Ty\|_\infty$. Let the distortion be maximized at $y^*$ and that

$$\|AA_S^Ty^*\|_\infty = \|A_S A_S^Ty^*\|_\infty = 1.$$ 

Further let $i$ be the coordinate such that $\|AA_S^Ty^*_i\|_\infty = (AA_S^Ty^*)_i$. Now for each $j \in [n]$, consider the following linear program:

$$\begin{align*}
\min_{(y,t)} & \quad t \\
\text{s.t.} & \quad A_i^TA_S^Ty = 1 \\
& \quad A_S A_S^Ty \leq t \cdot 1 \\
& \quad -A_S A_S^Ty \leq t \cdot 1.
\end{align*}$$

If $(y_j, t_j)$ is the optimum solution for the above problem, we note that $t_j = \|A_S A_S^Ty_j\|_\infty$. Since we have $A_i^TA_S^Ty_j = 1$, we have that $\|AA_S^Ty_j\|_\infty \geq 1$ and therefore we have that $t_j = \|A_S A_S^Ty_j\|_\infty \geq \|AA_S^Ty_j\|_\infty / \phi \geq 1 / \phi$. Thus for each $j \in [n]$, $1 / t_j$ gives a lower bound on the maximum distortion $\phi$.

Now consider the linear program corresponding to $i \in [n]$ is defined above. Consider the vector $y = y^* / (AA_S^Ty^*)_i$. By definition, we have $a_i^TA_S^Ty = a_i^TA_S^Ty^*/(AA_S^Ty^*)_i = 1$ and $\|A_S A_S^Ty\|_\infty = \|AA_S^Ty^*\|_\infty / (AA_S^Ty^*)_i = \|A_S A_S^Ty^*\|_\infty / \|AA_S^Ty^*_i\|_\infty = 1 / \phi$. Hence, $(y, 1 / \phi)$ is a feasible solution for the linear program corresponding to index $i$. Since we proved above that $t_j \geq 1 / \phi$ for all $j$, we get that $t_i = 1 / \phi$ and hence $\max_j 1 / t_j = \phi = \max_{x \in \text{rowspace}(A_S)} \|AA_S^Ty\|_\infty / \|A_S A_S^Ty\|_\infty$. In our experiments, we solve these linear programs and find the max-distortion within the rowspace of $A_S$.

C.2. Grayscale Images Used

We use images from (Leung, 2017) and (European Space Agency and NASA, 2006) for our experiments. The compressed versions of the images used are in Figure 1.