Robust Subspace Approximation in a Stream

Roie Levin[1]
roiel@cs.cmu.edu

Anish Sevekari[2]
asevekar@andrew.cmu.edu

David P. Woodruff[1]
dwoodruf@cs.cmu.edu

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213
[2] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213

December 4, 2018

Abstract

We study robust subspace estimation in the streaming and distributed settings. Given a set of n data points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^d$ and an integer k, we wish to find a linear subspace S of dimension k for which $\sum_i M(\text{dist}(S, a_i))$ is minimized, where $\text{dist}(S, x) := \min_{y \in S} \|x - y\|_2$, and $M(\cdot)$ is some loss function. When M is the identity function, S gives a subspace that is more robust to outliers than that provided by the truncated SVD. Though the problem is NP-hard, it is approximable within a $(1 + \epsilon)$ factor in polynomial time when k and ϵ are constant. We give the first sublinear approximation algorithm for this problem in the turnstile streaming and arbitrary partition distributed models, achieving the same time guarantees as in the offline case. Our algorithm is the first based entirely on oblivious dimensionality reduction, and significantly simplifies prior methods for this problem, which held in neither the streaming nor distributed models.

1 Introduction

A fundamental problem in large-scale machine learning is that of subspace approximation. Given a set of n data points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^d$ and an integer k, we wish to find a linear subspace S of dimension k for which $\sum_i M(\text{dist}(S, a_i))$ is minimized, where $\text{dist}(S, x) := \min_{y \in S} \|x - y\|_2$, and $M(\cdot)$ is some loss function. When $M(\cdot) = (\cdot)^2$, this is the well-studied least squares subspace approximation problem. The minimizer in this case can be computed exactly by computing the truncated SVD of the data matrix.

Otherwise M is often chosen from $(\cdot)^p$ for some $p \geq 0$, or from a class of functions called M-estimators, with the goal of providing a more robust estimate than least squares in the face of outliers. Indeed, for $p < 2$, since one is not squaring the distances to the subspace, one is placing less emphasis on outliers and therefore capturing more of the remaining data points. For example, when M is the identity function, we are finding a subspace so as to minimize the sum of distances to it, which could arguably be more natural than finding a subspace so as to minimize the sum of squared distances. We can write this problem in the following form:

$$\min_{S \dim k} \sum_i \text{dist}(S, a_i) = \min_{X \text{ rank } k} \sum_i \|(A - AX)_{i*}\|_2$$

where A is the matrix in which the i-th row is the vector a_i. This is the form of robust subspace approximation that we study in this work. We will be interested in the approximate version of the problem for which the goal is to output a k-dimensional subspace S' for which with high probability,

$$\sum_i \operatorname{dist}(S', a_i) \leq (1+\epsilon) \sum_i \operatorname{dist}(S, a_i) \tag{1}$$

The particular form with M equal to the identity was introduced to the machine learning community by Ding et al. [11], though these authors employed heuristic solutions. The series of work in [8],[16] and [9, 13, 21, 6] shows that if $M(\cdot) = |\cdot|^p$ for $p \neq 2$, there is no algorithm that outputs a $(1 + 1/\operatorname{poly}(d))$ approximation to this problem unless P = NP. However, [6] also show that for any p there is an algorithm that runs in $O(\operatorname{nnz}(A) + (n+d)\operatorname{poly}(k/\epsilon) + \exp(\operatorname{poly}(k/\epsilon)))$ time and outputs a k-dimensional subspace whose cost is within a $(1+\epsilon)$ factor of the optimal solution cost. This provides a considerable computational savings since in most applications $k \ll d \ll n$. Their work builds upon techniques developed in [14] and [12] which give $O\left(nd \cdot \operatorname{poly}(k/\epsilon) + \exp\left((k/\epsilon)^{O(p)}\right)\right)$ time algorithms for the $p \geq 1$ case. These in turn build on the weak coreset construction of [10]. In other related work [7] give algorithms for performing regression with a variety of M-estimator loss functions.

Our Contributions. We give the first sketching-based solution to this problem. Namely, we show it suffices to compute $Z \cdot A$, where Z is a $\operatorname{poly}(\log(nd)k\epsilon^{-1}) \times n$ random matrix with entries chosen obliviously to the entries of A. The matrix Z is a block matrix with blocks consisting of independent Gaussian entries, while other blocks consist of independent Cauchy random variables, and yet other blocks are sparse matrices with non-zero entries in $\{-1, 1\}$. Previously such sketching-based solutions were known only for $M(\cdot) = (\cdot)^2$. Prior algorithms [9, 13, 21, 6] also could not be implemented as single-shot sketching algorithms since they require first making a pass over the data to obtain a crude approximation, and then using (often adaptive) sampling methods in future passes to refine to a $(1+\epsilon)$-approximation. Our sketching-based algorithm, achieving $O(\operatorname{nnz}(A) + (n+d)\operatorname{poly}(\log(nd)k/\epsilon) + \exp(\operatorname{poly}(k\epsilon^{-1})))$ time, matches the running time of previous algorithms and has considerable benefits as described below.

Streaming Model. Since Z is linear and oblivious, one can maintain $Z \cdot A$ in the presence of insertions and deletions to the entries of A. Indeed, given the update $A_{i,j} \leftarrow A_{i,j} + \Delta$ for some $\Delta \in \mathbb{R}$, we simply update the j-th column ZA_j in our sketch to $ZA_j + \Delta \cdot Z \cdot e_i$, where e_i is the i-th standard unit vector. Also, the entries of Z can be represented with limited independence, and so Z can be stored with a short random seed. Consequently, we obtain the first algorithm with $d \operatorname{poly}(\log(nd)k\epsilon^{-1})$ memory for this problem in the standard turnstile data stream model [20]. In this model, $A \in \mathbb{R}^{n \times d}$ is initially the zero matrix, and we receive a stream of updates to A where the i-th update is of the form (x_i, y_i, c_i), which means that A_{x_i, y_i} should be incremented by c_i. We are allowed one pass over the stream, and should output a rank-k matrix X' which is a $(1+\epsilon)$ approximation to the robust subspace estimation problem, namely $\sum_i \|(A - AX')_{i*}\|_2 \leq (1+\epsilon) \min_{X \text{ rank } k} \sum_i \|(A - AX)_{i*}\|_2$. The space complexity of the algorithm is the total number of words required to store this information during the stream. Here, each word is $O(\log(nd))$ bits. Our algorithm achieves $d \operatorname{poly}(\log(nd)k\epsilon^{-1})$ memory, and so only logarithmically depends on n. This is comparable to the memory of streaming algorithms when $M(\cdot) = (\cdot)^2$ [4, 15], which is the only prior case for which streaming algorithms were known.

Distributed Model. Since our algorithm maintains $Z \cdot A$ for an oblivious linear sketch Z, it is parallelizable, and can be used to solve the problem in the distributed setting in which there are s machines holding $A^1, A^2, \ldots, A^s$, respectively, and $A = \sum_{i=1}^s A^i$. This is called the *arbitrary partition model* [18]. In this model, we can solve the problem in one round with $s \cdot d \operatorname{poly}(\log(nd)k\epsilon^{-1})$ communication by having each machine agree upon (a short seed describing) Z, and sending ZA^i to a central coordinator who computes and runs our algorithm on $Z \cdot A = \sum_i ZA^i$. The arbitrary partition model is stronger than the so-called row partition model, in which the points (rows of A) are partitioned across machines. For example, if each machine corresponds to a shop, the rows of A correspond to customers, the columns of A correspond to items, and $A^i_{c,d}$ indicates how many times customer c purchased item d at shop i, then the row partition model requires customers to make purchases at a single shop. In contrast, in the arbitrary partition model, customers can purchase items at multiple shops.

2 Notation and Terminology

For a matrix A, let A_{i*} denote the i-th row of A, and A_{*j} denote the j-th column of A.

Definition 2.1 ($\|\cdot\|_{2,1}$, $\|\cdot\|_{1,2}$, $\|\cdot\|_{1,1}$, $\|\cdot\|_{\mathrm{med},1}$, $\|\cdot\|_F$)**.** *For a matrix $A \in \mathbb{R}^{n\times m}$, let:*

$$\|A\|_{2,1} \equiv \sum_i \|A_{i*}\|_2 \qquad \|A\|_{1,2} \equiv \|A^\intercal\|_{2,1} = \sum_j \|A_{*j}\|_2$$

$$\|A\|_F \equiv \sqrt{\sum_i \|A_{i*}\|_2^2} \qquad \|A\|_{1,1} \equiv \sum_i \|A_{i*}\|_1 \qquad \|A\|_{\mathrm{med},1} \equiv \sum_j \|A_{*j}\|_{\mathrm{med}}$$

where $\|\cdot\|_{\mathrm{med}}$ denotes the function that takes the median of absolute values.

Definition 2.2 (X^*, Δ^*)**.** *Let:*

$$\Delta^* \equiv \min_{X \text{ rank } k} \|A - AX\|_{2,1} \qquad X^* \equiv \operatorname*{argmin}_{X \text{ rank } k} \|A - AX\|_{2,1}$$

Definition 2.3 ((α, β)-coreset)**.** *For a matrix $A \in \mathbb{R}^{n\times d}$ and a target rank k, W is an (α, β)-coreset if its row space is an α-dimensional subspace of $\mathbb{R}^d$ that contains a β-approximation to X^*. Formally:*

$$\operatorname*{argmin}_{X \text{ rank } k} \|A - AXW\|_{2,1} \le \beta\Delta^*$$

Definition 2.4 (Count-Sketch Matrix)**.** *A random matrix $S \in \mathbb{R}^{r\times t}$ is a Count-Sketch matrix if it is constructed via the following procedure. For each of the t columns S_{*i}, we first independently choose a uniformly random row $h(i) \in \{1, 2, ..., r\}$. Then, we choose a uniformly random element of $\{-1, 1\}$ denoted $\sigma(i)$. We set $S_{h(i),i} = \sigma(i)$ and set $S_{j,i} = 0$ for all $j \ne i$.*

For the applications of Count-Sketch matrices in this paper, it suffices to use $O(1)$-wise instead of full independence for the hash and sign functions. Thus these can be stored in $O(1)$ space, and multiplication SA can be computed in $\mathrm{nnz}(A)$ time. For more background on such sketching matrices, we refer the reader to the monograph [25].

We also use the following notation: $[n]$ denotes the set $\{1, 2, 3, \cdots n\}$. $[\![E]\!]$ denotes the indicator function for event E. $\mathrm{nnz}(A)$ denotes the number of non-zero entries of A. A^- denotes the pseudoinverse of A. $\mathcal{I}$ denotes the identity matrix.

3 Algorithm Overview

At a high level we follow the framework put forth in [6] which gives the first input sparsity time algorithm for the robust subspace approximation problem. In their work Clarkson and Woodruff first find a crude $(\mathrm{poly}(k), K)$-coreset for the problem. They then use a non-adaptive implementation of a residual sampling technique from [10] to improve the approximation quality but increase the dimension, yielding a $(K \operatorname{poly}(k), 1+\epsilon)$-coreset. From here they further use dimension reducing sketches to reduce to an instance with parameters that depend only polynomially on k/ϵ. Finally they pay a cost exponential only in $\operatorname{poly}(k/\epsilon)$ to solve the small problem via a black box algorithm of [3].

There are several major obstacles to directly porting this technique to the streaming setting. For one, the construction of the crude approximation subspace uses leverage score sampling matrices which are non-oblivious and thus not usable in 1-pass turnstile model algorithms. We circumvent this difficulty in Section 4.1 by showing that if T is a sparse $\operatorname{poly}(k) \times n$ matrix of Cauchy random variables, the row span of TA contains a rank-k matrix which is a $\log(d) \operatorname{poly}(k)$ approximation to the best rank-k matrix under the $\|\cdot\|_{2,1}$ norm.

Second, the residual sampling step requires sampling rows of A with respect to probabilities proportional to their distance to the crude approximation (in our case TA). This is challenging because one does not know TA until the end of the stream, much less the distances of rows of A to TA. We handle this in Section 4.2 using a row-sampling data structure of [1, 22, 24], which for a matrix B maintains a sketch HB in a stream from which one can extract samples of rows of B according to probabilities given by their norms. By linearity, it suffices to maintain HA and TA in parallel in the stream, and apply the sample extraction procedure to $HA \cdot (\mathcal{I} - P_{TA})$, where $P_{TA} = (TA)^\intercal (TA(TA)^\intercal)^{-1} TA$ is the projection onto the rowspace of TA. Unfortunately, the extraction procedure only returns noisy perturbations of the original rows which majorly invalidates the analysis in [6] of the residual sampling. In Section 4.2 we give an analysis of non-adaptive noisy residual sampling which we name BOOTSTRAPCORESET. This gives a procedure for transforming our $\operatorname{poly}(k)$-dimensional space containing a $\operatorname{poly}(k)\log(d)$ approximation into a $\operatorname{poly}(k)\log(d)$-dimensional space containing a $3/2$ factor approximation.

Third, requiring the initial crude approximation to be oblivious yields a coarser $\log(d)\operatorname{poly}(k)$ initial approximation than the constant factor approximation of [6]. Thus the dimension of the subspace after residual sampling is $\operatorname{poly}(k)\log(d)$. Applying dimension reduction techniques reduces the problem to an instance with $\operatorname{poly}(k)$ rows and $\log(d)\operatorname{poly}(k)$ columns. Here the black box algorithm of [3] would take time $d^{\operatorname{poly}(k)}$ which is no longer fixed parameter tractable as desired. Our key insight is that finding the best rank-k matrix under the Frobenius norm, which can be done efficiently, is a $\sqrt{\log d}(\log\log d)\operatorname{poly}(k)$ approximation to the $\|\cdot\|_{2,1}$ norm minimizer. From here we can repeat the residual sampling argument which this time yields a small instance with $\operatorname{poly}(k)$ rows by $\sqrt{\log d}(\log\log d)\operatorname{poly}(k/\epsilon)$ columns. Sublogarithmic in d makes all the difference and now enumerating can be done in time $(n+d)\operatorname{poly}(k/\epsilon) + \exp(\operatorname{poly}(k/\epsilon))$. All this is done in parallel in a single pass of the stream.

Lastly, the sketching techniques applied after the residual sampling are not oblivious in [6]. We instead use an obvlious median based embedding in Section 5.1, and show that we can still use the black box algorithm of [3] to find the minimizer under the $\|\cdot\|_{\mathrm{med},1}$ norm in Section 5.2.

We present our results as two algorithms for the robust subspace approximation problem. The first runs in fully polynomial time but gives a coarse approximation guarantee, which corresponds to stopping before repeating the residual sampling a second time. The second algorithm captures the entire procedure, and uses the first as a subroutine.

Algorithm 1 COARSEAPPROX

Input: $A \in \mathbb{R}^{n \times d}$ as a stream
Output: $X \in \mathbb{R}^{d \times d}$ such that $\|A - AX\|_{2,1} \leq \sqrt{\log d}(\log \log d) \operatorname{poly}(k)\Delta^*$

$T \in \mathbb{R}^{\operatorname{poly}(k) \times n} \leftarrow$ Sparse Cauchy matrix // as in Thm. 4.1
$C_1 \in \mathbb{R}^{\operatorname{poly}(k) \times n} \leftarrow$ Sparse Cauchy matrix // as in Thm. 4.4
$S_1 \in \mathbb{R}^{\log d \cdot \operatorname{poly}(k) \times d} \leftarrow$ Count Sketch composed with Gaussian // as in Thm. 4.3
$R_1 \in \mathbb{R}^{\operatorname{poly}(k) \times d} \leftarrow$ Count Sketch matrix // as in Thm. 4.3
$G_1 \in \mathbb{R}^{\log d \cdot \operatorname{poly}(k) \times \log d \cdot \operatorname{poly}(k)} \leftarrow$ Gaussian matrix // as in Thm. 4.4
Compute TA online
Compute $C_1 A$ online
$U_1^{\intercal} \in \mathbb{R}^{\log d \operatorname{poly}(k) \times d} \leftarrow$ BOOTSTRAPCORESET$(A, TA, 1/2)$ // as in Alg. 3
$\hat{X} \in \mathbb{R}^{\operatorname{poly}(k) \times \log d \operatorname{poly}(k)} \leftarrow \operatorname{argmin}_{X \text{ rank } k} \|C_1(A - AR_1^{\intercal} X U_1^{\intercal}) S_1^{\intercal} G_1\|_F$ // as in Fact 4.2
return $R_1^{\intercal} \hat{X} U^{\intercal}$

Theorem 3.1 (Coarse Approximation in Polynomial Time)**.** *Given a matrix $A \in R^{n \times d}$, Algorithm 1 with constant probability computes a rank k matrix $X \in \mathbb{R}^{d \times d}$ such that:*

$$\|A - AX\|_{2,1} \leq \sqrt{\log d}(\log \log d) \cdot \operatorname{poly}(k) \cdot \|A - AX^*\|_{2,1}$$

that runs in time $O(nnz(A)) + d \operatorname{poly}(k \log(nd))$. Furthermore, it can be implemented as a one-pass streaming algorithm with space $O\left(d \operatorname{poly}(k \log(nd))\right)$ and time per update $O(\operatorname{poly}(\log(nd)k))$.

Proof Sketch We show the following are true in subsequent sections:

1. The row span of TA is a $(\operatorname{poly}(k), \log d \cdot \operatorname{poly}(k))$-coreset for A (Section 4.1) with probability $24/25$.
2. BOOTSTRAPCORESET$(A, TA, 1/2)$ is a $(\log d \cdot \operatorname{poly}(k), 3/2)$-coreset with probability $49/50$ (Section 4.2).
3. If:

$$\hat{X} = \underset{X \text{ rank } k}{\operatorname{argmin}} \|C_1 A S_1^{\intercal} G_1 - C_1 A R_1^{\intercal} X U_1^{\intercal} S_1^{\intercal} G_1\|_F$$

then with probability $47/50$:

$$\left\| A - AR_1^{\intercal} \hat{X} U_1^{\intercal} \right\|_{2,1} \leq \operatorname{poly}(k) \sqrt{\log d} \log \log d \cdot \Delta^*$$

(Sections 4.3 and 4.4, with $\epsilon = 1/2$).

By a union bound, with probability 88/100 all the statements above hold, and the theorem is proved. BOOTSTRAPCORESET requires $d \operatorname{poly}(k \log(nd))$ space and time. Left matrix multiplications by Sparse Cauchy matrices TA and $C_1 A$ can be done in $O(\operatorname{nnz}(A))$ time (see Section J of [23] for a full description of Sparse Cauchy matrices). Computing remaining matrix products and $\hat{X}$ requires time $d \operatorname{poly}(k \log d)$. $\square$

Algorithm 2 $(1+\epsilon)$-APPROX

Input: $A \in \mathbb{R}^{n\times d}$ as a stream

Output: $X \in \mathbb{R}^{d\times d}$ such that $\|A - AX\|_{2,1} \le (1+\epsilon)\Delta^*$

1: $\hat{X} \in \mathbb{R}^{\mathrm{poly}(k)\times \log d\, \mathrm{poly}(k)} \leftarrow$ COARSEAPPROX(A) // as in Thm. 3.1

2: $C_2 \in \mathbb{R}^{\sqrt{\log d}(\log\log d)\,\mathrm{poly}(k/\epsilon)\times n} \leftarrow$ Cauchy matrix // as in Thm. 5.1

3: $S_2 \in \mathbb{R}^{\sqrt{\log d}(\log\log d)\cdot\mathrm{poly}(k/\epsilon)\times d} \leftarrow$ Count Sketch composed with Gaussian // as in Thm. 4.3

4: $R_2 \in \mathbb{R}^{\mathrm{poly}(k/\epsilon)\times d} \leftarrow$ Count Sketch matrix // as in Thm. 4.3

5: $G_2 \in \mathbb{R}^{\sqrt{\log d}(\log\log d)\cdot\mathrm{poly}(k/\epsilon)\times\sqrt{\log d}(\log\log d)\cdot\mathrm{poly}(k/\epsilon)} \leftarrow$ Gaussian matrix // as in Thm. 5.1

6: Compute $AR_2^\intercal$ online

7: Compute $AS_2^\intercal$ online

8: Let $V \in \mathbb{R}^{\log d\,\mathrm{poly}(k)\times k}$ be such that $\hat{X} = WV^\intercal$ is the rank-k decomposition of $\hat{X}$

9: $U_2^\intercal \in \mathbb{R}^{\mathrm{poly}(k/\epsilon)\sqrt{\log d}\log\log d\times d} \leftarrow$ BOOTSTRAPCORESET$(A, V^\intercal U_1^\intercal, \epsilon')$ // as in Alg. 3, U_1 as computed during COARSEAPPROX in line 1.

10: $\hat{X}' \in \mathbb{R}^{\mathrm{poly}(k/\epsilon)\times\mathrm{poly}(k/\epsilon)\sqrt{\log d}\log\log d} \leftarrow \mathrm{argmin}_{X\text{ rank }k}\|C_2(A - AR_2^\intercal X U_2^\intercal)S_2^\intercal G_2\|_{\mathrm{med},1}$ // as in Thm. 5.2

11: **return** $R_2^\intercal \hat{X}' U'^\intercal$

Theorem 3.2 ($(1+\epsilon)$-Approximation)**.** *Given a matrix $A \in R^{n\times d}$, Algorithm 2 with constant probability computes a rank k matrix $X \in \mathbb{R}^{d\times d}$ such that:*

$$\|A - AX\|_{2,1} \le (1+\epsilon)\|A - AX^*\|_{2,1}$$

that runs in time

$$O(nnz(A)) + (n+d)\,\mathrm{poly}\left(\frac{k\log(nd)}{\epsilon}\right) + \exp\left(\mathrm{poly}\left(\frac{k}{\epsilon}\right)\right)$$

Furthermore, it can be implemented as a one-pass streaming algorithm with space $O\left(d\,\mathrm{poly}\left(\frac{k\log(nd)}{\epsilon}\right)\right)$ and time per update $O(\mathrm{poly}(\log(nd)k/\epsilon))$.

Proof Sketch We show the following are true in subsequent sections:

1. If V is such that $\hat{X} = WV^\intercal$, then $V^\intercal$ is a $(\mathrm{poly}(k), \mathrm{poly}(k)\sqrt{\log d}\log\log d)$-coreset with probability $88/100$ (Theorem 3.1).
2. BOOTSTRAPCORESET$(A, V^\intercal U_1^\intercal, \epsilon')$ is a $(\mathrm{poly}(k/\epsilon')\sqrt{\log d}\log\log d, (1+\epsilon'))$-coreset with probability $49/50$ (Reusing Section 4.2).
3. If:
$$\hat{X}' \leftarrow \underset{X}{\mathrm{argmin}}\|C_2(A - AR_2^\intercal X U_2^\intercal)S_2^\intercal G_2\|_{\mathrm{med},1}$$
then with probability $19/20$:
$$\left\|A - AR_2^\intercal \hat{X}' U_2^\intercal\right\|_{2,1} \le (1 + O(\epsilon'))\Delta^*$$
(Reusing Section 4.3 and Section 5.1).
4. A black box algorithm of [3] computes $\hat{X}'$ to within $(1+O(\epsilon'))$ (Section 5.2).

By a union bound, with probability $81/100$ all the statements above hold. Setting ϵ' appropriately small as a function of ϵ, the theorem is proved.

COARSEAPPROX and BOOTSTRAPCORESET together require $d\,\mathrm{poly}(k\log(nd)/\epsilon)$ space and $O(\mathrm{nnz}(A)) + d\,\mathrm{poly}(k\log(nd)/\epsilon)$ time. Right multiplication by the sketching matrices $AS_2^\intercal$ and $AR_2^\intercal$ can be done in time $\mathrm{nnz}(A)$. Computing remaining matrix products and $\hat{X}'$ requires time $(n+d)\,\mathrm{poly}(\log(d)k/\epsilon) + \exp(\mathrm{poly}(k/\epsilon))$ (See end of Section 5.2 for details on this last bound). □

We give further proofs and details of these theorems in subsequent sections.

4 Coarse Approximation

4.1 Initial Coreset Construction

We construct a $(\mathrm{poly}(k), \log d \cdot \mathrm{poly}(k))$-coreset which will serve as our starting point.

Theorem 4.1. *If $T \in \mathbb{R}^{\mathrm{poly}(k)\times n}$ is a sparse Cauchy matrix, then the row space of TA contains a k dimensional subspace with corresponding projection matrix X' such that with probability $24/25$:*

$$\left\|A - AX'\right\|_{2,1} \le \log d \cdot \mathrm{poly}(k) \min_{X \text{ rank } k} \|A - AX\|_{2,1} = \log d \cdot \mathrm{poly}(k) \cdot \Delta^*$$

Proof. In order to deal with the awkward $\|\cdot\|_{2,1}$ norm, we make use of a well known theorem due to Dvoretzky to convert it into an entrywise 1-norm.

Fact 4.1 (Dvoretzky's Theorem (Special Case), Section 3.3 of [17])**.** *There exists an appropriately scaled Gaussian Matrix $G \in \mathbb{R}^{d\times \frac{d\log(1/\epsilon)}{\epsilon^2}}$ such that w.h.p. the following holds for all $y \in \mathbb{R}^d$ simultaneously*

$$\|y^\intercal G\|_1 \in (1\pm\epsilon)\,\|y^\intercal\|_2$$

Applying this to all rows at once: $\|AX - A\|_{2,1} \in (1\pm\epsilon)\,\|AXG - AG\|_{1,1}$.

We also use some existing machinery for input sparsity time ℓ_1 subspace embeddings.

Fact 4.1.1 (Theorem 4 from [19])**.** *For any given $D \in \mathbb{R}^{s\times t}$, let $\Pi \in \mathbb{R}^{r\times s}$ be a random Sparse Cauchy matrix with $r = O(t^5\log^5 t)$ defined as follows: $\Pi = SC$ where $S \in \mathbb{R}^{r\times s}$ has each column uniformly and independently chosen from the r standard basis vectors in $\mathbb{R}^r$, and where $C \in \mathbb{R}^{s\times s}$ is a diagonal matrix with diagonal entries chosen independently from the standard Cauchy distribution. Then with probability $99/100$ simultaneously for all $x \in \mathbb{R}^t$:*

$$\frac{1}{O(t^2\log^2 t)} \cdot \|Dx\|_1 \le \|\Pi Dx\|_1 \le O(t\log t)\cdot\|Dx\|_1$$

Fact 4.1.2 (Lemma D.25 from [23])**.** *If $\Pi \in \mathbb{R}^{r\times s}$ is a Sparse Cauchy matrix as defined above, and $B \in \mathbb{R}^{s\times t}$ is a fixed matrix, then with probability at least $99/100$:*

$$\|\Pi B\|_1 \le O(\log(rt))\,\|B\|_1$$

Finally, we also need a couple of structural lemmas which we state here without proof:

Lemma 4.1.1 (Lemma 29 from [6])**.** *For a fixed (B, D) pair such that $B \in \mathbb{R}^{r\times s}, D \in \mathbb{R}^{r\times t}$, if $S \in \mathbb{R}^{s/\mathrm{poly}(\epsilon)\times r}$ is a CountSketch Matrix composed with a matrix of i.i.d. Gaussians (for background on such sketching matrices, we refer the reader to the monograph [25]), then with probability $99/100$ both of the properties below hold:*

1. $\|S(BX-D)\|_{1,2} \geq (1-\epsilon)\|BX-D\|_{1,2}$ *for any* X.
2. *If* $X^* = \operatorname{argmin}_{X \text{ rank } k} \|BX-D\|_{1,2}$*, then* $\|S(BX^*-D)\|_{1,2} \leq (1+\epsilon)\|BX^*-D\|_{1,2}$.

Clarkson and Woodruff [6] call such an S a lopsided embedding for (B, D) with respect to the $(1, 2)$-norm.

Lemma 4.1.2 (Lemma 31 from [6])**.** *If R is a lopsided embedding for $(A_k^\intercal, A^\intercal)$, then:*

$$\min_{X \text{ rank } k} \|AR^\intercal X - A\|_{2,1} \leq (1+3\epsilon)\Delta^*$$

Let $X' = \operatorname{argmin}_X \|TAR^\intercal X - TA\|_{2,1}$, $R^\intercal \in \mathbb{R}^{d \times \text{poly}(k)}$ as in the lemma above and $\epsilon = O(1)$.

Define E_1 to be the event that the condition in Dvoretzky's theorem is satisfied, E_2 to be the event that Fact 4.1.1 holds for $D = AR^\intercal$, E_3 to be the event that Fact 4.1.2 holds for $B = AR^\intercal X^* G - AG$, and E_4 to be the event that R satisfies Lemma 4.1.2.

E_1 holds w.h.p., E_2, E_3, E_4 each separately hold with probability $99/100$ (for a suitable choice of K). By a union bound, they all hold simultaneously with probability at least $24/25$. Conditioned on this happening:

$$\left\|AR^\intercal X' - A\right\|_{2,1} \leq \|AR^\intercal X^* - A\|_{2,1} + \left\|AR^\intercal(X^* - X')\right\|_{2,1} \tag{1}$$

$$\leq \|AR^\intercal X^* - A\|_{2,1} + \text{poly}(k) \cdot \left\|TAR^\intercal(X^* - X')G\right\|_{1,1} \tag{2}$$

$$\leq \text{poly}(k)\left(\|AR^\intercal X^* - A\|_{2,1} + \|T(AR^\intercal X^* - A)G\|_{1,1} + \left\|T(AR^\intercal X' - A)G\right\|_{1,1}\right) \tag{3}$$

$$\leq \text{poly}(k)\left(\|AR^\intercal X^* - A\|_{2,1} + 2\|T(AR^\intercal X^* - A)G\|_{1,1}\right) \tag{4}$$

$$\leq \text{poly}(k)\left(\|AR^\intercal X^* - A\|_{2,1} + O(\log d)\|(AR^\intercal X^* - A)G\|_{1,1}\right) \tag{5}$$

$$\leq \log d \cdot \text{poly}(k)\|AR^\intercal X^* - A\|_{2,1} \tag{6}$$

(1) and (3) hold by the triangle inequality, (2) since E_1 and E_2 hold, (4) by E_1 again and since X' is the minimizer of the expression $\|TAR^\intercal X - TA\|_{2,1}$, (5) since E_3 holds, and (6) by E_1 again.

X' lies in the rowspace of TA, since otherwise there is a rank-k projection Z onto the rows of TA with $\|TAX'Z - TAZ\|_{2,1} = \|TAX'Z - TA\|_{2,1}$ smaller than $\|TAX' - TA\|_{2,1}$. Since E_4 holds,

$$\|AR^\intercal X^* - A\|_{2,1} \leq O(1)\Delta^*$$

and thus the rowspace of TA contains a $\log d \cdot \text{poly}(k)$ approximation. □

Thus the rowspace of TA with T as in Theorem 4.1 above is a $(\text{poly}(k), \log d \cdot \text{poly}(k))$-coreset for A.

4.2 Bootstrapping a Coreset

Given a poor coreset Q for A, we now show how to leverage known results about residual sampling from [10] and [6] to obtain a better coreset of slightly larger dimension.

Theorem 4.2. *Given Q, an (α, β)-coreset for A, with probability* $49/50$ BOOTSTRAPCORESET *returns an $(\alpha + \beta\,\text{poly}(k/\epsilon), (1+\epsilon))$-coreset for A. Furthermore* BOOTSTRAPCORESET *runs in space and time* $O(d\,\text{poly}(\beta \log(nd)k/\epsilon))$*, with* $\text{poly}(\beta\log(nd)k/\epsilon)$ *time per update in the streaming setting.*

Algorithm 3 BOOTSTRAPCORESET

Input: $A \in \mathbb{R}^{n \times d}$, $Q \in \mathbb{R}^{\alpha \times d}$ (α, β)-coreset, $\epsilon \in (0, 1)$
Output: $U \in \mathbb{R}^{(\alpha + \beta \operatorname{poly}(k/\epsilon)) \times d}$ $(\alpha + \beta \operatorname{poly}(k/\epsilon), (1+\epsilon))$-coresets

Compute HA online // as in Lem. 4.2.2
$P \leftarrow \beta \operatorname{poly}(k/\epsilon)$ samples of rows of $A(\mathcal{I} - Q)$ according to $\mathcal{P}(HA(\mathcal{I} - Q))$ // as in Lem. 4.2.2
$U^\intercal \leftarrow$ Orthonormal basis for $\operatorname{RowSpan}\left(\begin{bmatrix} Q \\ P \end{bmatrix}\right)$
return $U^\intercal$

Proof. Consider the following idealized noisy sampling process that samples rows of a matrix B. Sample a row B_i of B with probability $\frac{\|B_i\|_2}{\|B\|_{2,1}}$ and add an arbitrary noise vector E_i such that $\|E_i\|_2 \le \nu \|B_i\|_2$, where we fix the parameter $\nu = \frac{\epsilon}{100k\beta}$. Supposing we had such a process $\mathcal{P}^*(B)$, we can prove the following lemma.

Lemma 4.2.1. *Suppose Q is an (α, β)-coreset for A, and P is a noisy subset of rows of the residual $A(\mathcal{I} - Q)$ of size $\beta(\operatorname{poly} k/\epsilon)$ each sampled according to $\mathcal{P}^*(A(\mathcal{I} - Q))$. Then with probability $99/100$, $\operatorname{RowSpan}(Q) \cup \operatorname{RowSpan}(P)$ is an $(\alpha + \beta \operatorname{poly}(k/\epsilon))$ dimensional subspace containing a k-dimensional subspace with corresponding projection matrix X' such that:*

$$\left\|A - AX'\right\|_{2,1} \le (1+\epsilon)\Delta^*$$

Proof. Our theorem is identical to Theorem 45 from [6], which is in turn an adaptation of Theorem 9 from [10], except that our sampling procedure produces noisy samples instead of actual rows of $A(\mathcal{I} - Q)$. We highlight the difference between our proof and the originals, and refer the reader to the sources for a full description.

Let H_ℓ denote the span of the rows of Q adjoined with ℓ samples from $\mathcal{P}^*(A(\mathcal{I} - Q))$. The analysis considers $k+1$ phases during the construction of H_ℓ, where phase j is defined such that there exists a subspace X_j with:

(i) the dimension of $\operatorname{RowSpan}(X_j) \cap H_\ell \ge j$.
(ii) and letting $\delta = \epsilon/2k$ we have: $\|A(\mathcal{I} - X_j)\|_{2,1} \le (1+\delta)^j \min_{X \text{ rank } k} \|A - AX\|_{2,1}$

In other words, the cost of the solution X_j slowly gets worse with j, but H_ℓ recovers more of it. Note that in phase k, $\|A(\mathcal{I} - X_k)\|_{2,1} \le (1+\epsilon) \min_{X \text{ rank } k} \|A - AX\|_{2,1}$, and furthermore $X_k \subseteq H_\ell$.

Let Y_ℓ denote the rank-k projection whose row space is that of X_j, but rotated about the intersection $\operatorname{RowSpan}(X_j) \cap H_\ell$ such that it also contains the vector in H_ℓ realizing the smallest nonzero principle angle with X_j. Note that Y_ℓ satisfies condition (i) for some $j' > j$, so it remains to show that with high probability, with a small number of new samples, condition (ii) is also satisfied. In particular, we show that if condition (ii) is violated, and thus if:

$$\|A(\mathcal{I} - Y_\ell)\|_{2,1} > (1+\delta) \|A(\mathcal{I} - X_j)\|_{2,1} \tag{1}$$

then with probability greater than $\delta/5K$ we sample a witness noisy-row $\hat{A}_{\ell'*}$ with the property:

$$\left\|\hat{A}_{\ell'*}(\mathcal{I} - Y_\ell)\right\|_2 \ge (1+\delta/2) \left\|\hat{A}_{\ell'*}(\mathcal{I} - X_j)\right\|_2 \tag{2}$$

By the Angle Drop Lemma (Lemma 13 of [10]), this witness implies that the smallest nonzero principle angle between X_j and H_ℓ (and so Y_ℓ) decreases. By the analysis of Theorem 9 of [10], once the angle is small enough, Y_ℓ will satisfy (ii). We now prove this fact.

By the assumption on $\mathcal{P}^*$, $E_{\ell'}$ satisfies $\|E_{\ell'}\|_2 \leq \nu \|A_{\ell' *}(\mathcal{I} - Q)\|_2$. Recall we set the noise parameter $\nu = \frac{\epsilon}{100k\beta} = \frac{\delta}{50\beta}$.

Let W denote the set of indices of witness *noisy* rows, in other words the set of all i such that $\hat{A}_i$ satisfies (2). It suffices to show that:

$$\sum_{i \in W} \|A_{i*}(\mathcal{I} - Q)\|_2 \geq \frac{\delta}{5\beta} \|A(\mathcal{I} - Q)\|_{2,1} \tag{3}$$

Suppose that (3) is false. Let $\tilde{X}_\ell$ be the matrix projecting onto H_ℓ.

$$\begin{aligned}
\left\|\hat{A}_{i*}(\mathcal{I} - Y_\ell)\right\|_2 &\leq \left\|\hat{A}_{i*}(\mathcal{I} - \tilde{X}_\ell)\right\|_2 + \left\|\hat{A}_{i*}\tilde{X}_\ell(\mathcal{I} - Y_\ell)\right\|_2 && (4)\\
&\leq \left\|\hat{A}_{i*}(\mathcal{I} - \tilde{X}_\ell)\right\|_2 + \left\|\hat{A}_{i*}\tilde{X}_\ell(\mathcal{I} - X_j)\right\|_2 && (5)\\
&\leq 2\left\|\hat{A}_{i*}(\mathcal{I} - \tilde{X}_\ell)\right\|_2 + \left\|\hat{A}_{i*}(\mathcal{I} - X_j)\right\|_2 && (6)\\
&\leq 2\left\|\hat{A}_{i*}(\mathcal{I} - Q)\right\|_2 + \left\|\hat{A}_{i*}(\mathcal{I} - X_j)\right\|_2 && (7)
\end{aligned}$$

(4) and (6) follow from the triangle inequality, (5) since the definitions of X_j, Y_ℓ and H_ℓ imply that all elements of H_ℓ are closer to $\mathrm{RowSpan}(Y_\ell)$ than to $\mathrm{RowSpan}(X_j)$, and (7) since $\mathrm{RowSpan}(Q) \subseteq H_\ell$.

For $i \notin W$, by definition $\left\|\hat{A}_{i*}(\mathcal{I} - Y_\ell)\right\|_2 < (1 + \delta/2)\left\|\hat{A}_{i*}(\mathcal{I} - X_j)\right\|_2$. Combining both the bounds we have that for all i;

$$\left\|\hat{A}_{i*}(\mathcal{I} - Y_\ell)\right\|_2 \leq (1 + \delta/2)\left\|\hat{A}_{i*}(\mathcal{I} - X_j)\right\|_2 + [\![i \in W]\!] \cdot 2 \cdot \left\|\hat{A}_{i*}(\mathcal{I} - Q)\right\|_2$$

Summing over all i,

$$\left\|\hat{A}(\mathcal{I} - Y_\ell)\right\|_{2,1} \leq (1 + \delta/2)\left\|\hat{A}(\mathcal{I} - X_j)\right\|_{2,1} + 2\left\|\hat{A}_{W*}(\mathcal{I} - Q)\right\|_2$$

By triangle inequality:

$$\|A(\mathcal{I} - Y_\ell)\|_{2,1} - \|E(\mathcal{I} - Y_\ell)\|_{2,1} \leq \begin{bmatrix} (1 + \delta/2)\|A(\mathcal{I} - X_j)\|_{2,1} + 2\|A_{W*}(\mathcal{I} - Q)\|_{2,1} \\ +(1 + \delta/2)\|E(\mathcal{I} - X_j)\|_{2,1} + 2\|E(\mathcal{I} - Q)\|_{2,1} \end{bmatrix}$$

Finally, rearranging:

$$\begin{aligned}
\|A(\mathcal{I} - Y_\ell)\|_{2,1} &\leq \left(1 + \frac{\delta}{2}\right)\|A(\mathcal{I} - X_j)\|_{2,1} + \frac{2\delta}{5\beta} \cdot \beta\|A(\mathcal{I} - X_j)\|_{2,1} + 5\|E\|_{2,1} && (8)\\
&\leq \left(1 + \frac{9\delta}{10} + 5\nu\beta\right)\|A(\mathcal{I} - X_j)\|_{2,1} && (9)\\
&\leq (1 + \delta)\|A(\mathcal{I} - X_j)\|_{2,1}
\end{aligned}$$

Which contradicts our assumption that (1) held. (8) follows from the assumption that (3) is false and the fact that $\|A(\mathcal{I}-Q)\|_{2,1} \le \beta \|A(\mathcal{I}-X_j)\|_{2,1}$ and (9) since $\|E\|_{2,1} \le \nu \|A(\mathcal{I}-Q)\|_{2,1}$.
Note that this proof goes through for any error matrix E satisfying $\|E_i\| \le \nu \|A_i\|$ for all i. Also, as written in [6], the proof guarantees success with constant probability. We can repeat the sampling a constant number of times, keep all samples, and guarantee success with probability $99/100$. □

It remains to show that we can sample from $\mathcal{P}^*$ in a stream.

Lemma 4.2.2. *Let $B \in \mathbb{R}^{n\times d}$ be a matrix, and let $\delta, \nu \in (0,1)$ be given. Also let s be a given integer. Then there is an oblivious sketching matrix $H \in \mathbb{R}^{\mathrm{poly}(s/(\delta\nu))\times n}$ and a sampling process $\mathcal{P}$, such that $\mathcal{P}(HB)$ returns a collection of $s' = O(s)$ distinct row indices $i_1, \dots, i_{s'} \in [n]$ and approximations $\tilde{B}_{i_j} = B_{i_j} + E_{i_j}$ with $\|E_{i_j}\|_2 \le \nu \cdot \|B_{i_j}\|_2$ for $j = 1, \dots, s$. With probability $1-\delta$ over the choice of H, the probability an index i appears in the sampled set $\{i_1, \dots, i_{s'}\}$ is at least the probability that i appears in a set of s samples without replacement from the distribution $\left(\frac{\|B_{1,*}\|_2}{\|B\|_{2,1}}, \dots \frac{\|B_{n,*}\|_2}{\|B\|_{2,1}}\right)$. Furthermore the multiplication HB and sampling process $\mathcal{P}$ can be done in nnz$(B) + d \cdot \mathrm{poly}(s/(\delta\nu))$ time, and can be implemented in the streaming model with $d \cdot \mathrm{poly}(s/(\delta\nu))$ bits of space.*

The theorem builds on the work of [1], [22] and [24].

Proof. We will show that with probability $O(1)\cdot\delta'$, we produce a set of $1 - O(1)\cdot s$ samples such that the probability a noisy row $\tilde{B}_i = B_i + E_i$ with $\|E_i\|_2 \le \nu \|B_i\|_2$ appears in this set is at least $\frac{\|B_i\|_2}{\|B\|_{2,1}}$. Fixing $\delta' = \delta/O(1)$ will give the claim.

Algorithm 4 H-SKETCH

Input: $B \in \mathbb{R}^{n\times d}$

Output: $HB \in \mathbb{R}^{d\,\mathrm{poly}\left(\frac{s\log(nd)}{\nu\delta'}\right)\times d}$

for level $j \in [\ell]$ **do**
  $H_{j,*} \leftarrow$ new hash table with $w = O\left(\left(\frac{s\log n}{\nu\delta'}\right)^{15}\right)$ buckets and independent hash function $h_j \in ([n] \to [w])$ (each bucket stores a d dimensional vector).
  Sample a set $J_j \subset [n]$ where each $i \in [n]$ is included with probability $p_j = \frac{1}{2^j}$.
  for $v \in [w]$ **do**
    $H_{j,v} = \sum_{i\in J_j} [\![h_j(i) = v]\!] \cdot \varepsilon_j(i) \cdot B_{i*}$ where $\varepsilon_j(i)$ are 2-wise ind. uniform ± 1 random variables.
  end for
end for
return $\begin{bmatrix} H^{(1)} \\ H^{(2)} \\ \vdots \\ H^{(\ell)} \end{bmatrix}$

Before describing the algorithm we define a number of parameters.

- M is an estimate for $\|B\|_{2,1}$ such that $\|B\|_{2,1} \le M \le O(1)\|B\|_{2,1}$ (we show in Appendix A.2 that we can calculate such an M with high probability).
- Setting $T_j = M/2^j$, define $S_j = \{i \in [n] \mid \|B_i\|_2 \in (T_j, 2T_j]\}$ to be the jth level set of B.

- Define $s_j = |S_j|$ to be the number of rows in level j.
- For convenience, we also use the notation $S_{\geq j} = \bigcup_{j' \geq j} S_{j'}$ and $S_{\leq j} = \bigcup_{j' \leq j} S_{j'}$.
- Let $\ell = 4\log(n/\delta)$ be the set of levels we consider in our sketch.
- Define a level $j \in [\ell]$ to be important if $s_j \geq \frac{\delta' 2^j}{\ell s}$. Informally, j is an important level if the set of rows in in level j contribute a significant fraction of $\|B\|_{2,1}$.
- Let $\mathcal{J} \subset [\ell]$ denote the set of all important levels.

Observe that for any level j we have $s_j \leq 2^j$. It will suffice to consider only levels $j \in [\ell]$, since $s \leq n$ implies that these rows necessarily capture a $(1 - \delta'/s)$ fraction of the mass of B. By definition, the idealized process sampling from the distribution $\left(\frac{\|B_{1,*}\|_2}{\|B\|_{2,1}}, \ldots \frac{\|B_{n,*}\|_2}{\|B\|_{2,1}}\right)$ will sample a level $j \in [\ell]$ with probability $1 - \delta'/s$, and by a union bound all s samples come from such a level with probability $1 - \delta'$. Similarly, the idealized process will take a single sample from an important level with probability $(1 - \sum_{j \in [\ell]} T_j s_j / \|B\|_{2,1}) \geq 1 - \delta'/s$, meaning it only every samples from important levels also with probability $1 - \delta'$.

The main idea of the sketch is the following. For every level j, we subsample every row of B independently at random with probability proportional to 2^{-j}, and then hash the subsampled rows independently at random into buckets (each bucket is a vector that is the sum of the vectors assigned to it). Doing so guarantees that with high enough probability, for every important level j, there is a nearby level k such that with high probability at least one row from j is sampled in level k. Furthermore, all sufficiently heavy rows in level k hash to different buckets, and all light rows contribute at most νT_j to any one bucket. In particular, this means that if any bucket in important level k has norm in the range $((1-\nu)T_j, (2+\nu)T_j]$, that bucket is of the form $\tilde{B}_i = B_i + E_i$ where B_i is a row of B and E_i has small norm. We defer formal descriptions of these guarantees to Appendix A.1.

Next we argue that we can use the sketch from Algorithm 4 to produce samples from the idealized process with high enough probability. The general idea of the sampling algorithm SAMPLER is the following. Partition the rows of B by assigning each row to one of $t = 100s^3$ pieces uniformly at random: $B^{(1)}$, $B^{(2)}$, ..., $B^{(t)}$. We can bound the probability that any two out of s samples from the idealized process come from the same piece by $\binom{s}{2} \cdot \frac{\delta'}{100s^3} \leq \delta'$ so we can condition on this being the case. Sketch each piece using H-SKETCH to obtain: $H^{(1)}$, $H^{(2)}$, ..., $H^{(t)}$. Let $\ell^{(p)}$, $s_j^{(p)}$, $M^{(p)}$, $T_j^{(p)}$ denote the quantities ℓ, s_j, M and T_j respectively for piece $B^{(p)}$. Using Lemma A.4.1, with constant probability we can calculate simultaneously for all p an $O(1)$ estimate $\tilde{b}_p$ for $\left\|B^{(p)}\right\|_{2,1}$. Using Lemma A.3, we can also calculate simultaneously for all $j \in [\ell^{(p)}]$ a $O(1)$ estimate $\tilde{s}_j^{(p)}$ for $s_j^{(p)}$. Now repeat the following until we have generated s' samples.

Sample a piece with probability proportional to $\tilde{b}_p$, and within that piece sample a level j with probability proportional to $\tilde{s}_j T_j$. Examine level j of the output of Algorithm 4. If at least one bucket of this level has a norm that is in the target range $(T_j, 2T_j]$, then output a uniform random choice of such a bucket. We show that this process generates samples with probabilities sufficiently close to those of the idealized process.

Let C_i be the event that noisy row $\tilde{B}_i = B_i + E_i$ is extracted on line 7. Let G_p be the event that piece p is sampled on line 9. Let D_j be the event that level j is sampled on line 10. Finally let E denote the event that any noisy row of the form $B_i + E_i$ with $\|E_i\|_2 \leq \nu \|B_i\|_2$ is extracted at all in iteration z. We wish to understand the probability of C_i:

$$\mathbb{P}[C_i] = \mathbb{P}\left[C_i \mid E \wedge D_{j'} \wedge G_{p'}\right] \cdot \mathbb{P}\left[E \mid D_{j'} \wedge G_{p'}\right] \cdot \mathbb{P}\left[D_{j'} \mid G_{p'}\right] \cdot \mathbb{P}\left[G_{p'}\right]$$

Algorithm 5 SAMPLER

Input: $HB \in \mathbb{R}^{d\,\mathrm{poly}\left(\frac{s\log(nd)}{\nu\delta'}\right)\times d}$
Output: $B_{i_1}, \ldots, B_{i_{s'}}$ samples

Partition rows of B uniformly at random into $t = 100s^3/\delta'$ pieces: $B^{(1)}$, $B^{(2)}$, ..., $B^{(t)}$.
for $p \in [t]$ **do**
  $H^{(p)} \leftarrow$ H-SKETCH($B^{(p)}$), computed online.
  Calculate estimates $\tilde{b}_p \in \left[\left\|B^{(p)}\right\|_{2,1}, O(1)\cdot\left\|B^{(p)}\right\|_{2,1}\right]$ // as in Lemma A.4.1.
  For all $j \in [\ell^{(p)}]$, calculate estimates $\tilde{s}_j^{(p)} \in \left[s_j^{(p)}, O(1)\cdot s_j^{(p)}\right]$ // as in Lemma A.3.
  Set $M^{(p)} = \tilde{b}_p$.
end for
$F \leftarrow \emptyset$
while $|F| < s'$ **do**
  Sample a piece $p' \in [t]$ with probability $\frac{\tilde{b}_{p'}}{\sum_p \tilde{b}_p}$ (without replacement).
  Sample a level $j' \in [\ell]$ in $B^{(p')}$ with probability $\frac{\tilde{s}_{j'}^{(p')}T_{j'}^{(p')}}{\sum_j \tilde{s}_j^{(p')}T_j^{(p')}}$.
  Let $k = \max\left(0, j' - 2\log\left(\frac{s\log n}{\delta'\nu}\right)\right)$.
  if at least one bucket v of $H_k^{(p')}$ has $\left\|H_{k,v}^{(p')}\right\|_2 \in ((1-\nu)T_{j'}, (2+\nu)T_{j'})$ **then**
    $F \leftarrow F \cup \left\{\text{uniform random } H_{k,v'}^{(p')} \text{ such that } \left\|H_{k,v'}^{(p')}\right\|_2 \in ((1-\nu)T_{j'}, (2+\nu)T_{j'})\right\}$
  end if
end while
return F

We have straightforward bounds on the last two probabilities:

$$\mathbb{P}\left[G_p\right] = \frac{\tilde{b}_{p'}}{\sum_p \tilde{b}_p} = \Theta(1)\frac{\left\|B^{(p)}\right\|_{2,1}}{\|B\|_{2,1}}$$

$$\mathbb{P}\left[D_{j'} \mid G_p\right] = \frac{\tilde{s}_{j'}^{(p)}T_{j'}^{(p)}}{\sum_j \tilde{s}_j^{(p)}T_j^{(p)}} = \Theta(1)\cdot\frac{s_{j'}^{(p)}\|B_i\|_2}{\left\|B^{(p)}\right\|_{2,1}}$$

Now we can also lower bound $\mathbb{P}\left[E \mid D_j \wedge G_p\right]$. E will not hold if either:

(i) a noisy row $\tilde{B}$ is sampled but $\tilde{B}$ cannot be written $B_i + E_i$ with $\|E_i\|_2 \leq \nu$.

(ii) no row is sampled at all.

If Lemmas A.1 and A.2 hold (i) will not occur. If Lemmas A.1 and Corollary A.1 hold (ii) will not occur. All these hold individually with probability at least $1 - O(1)/\log n$, so E holds with probability at least $1 - O(1)/\log n$. Finally, since conditioned on $E \wedge D_{j'} \wedge G_{p'}$ we pick any row in level j' from piece p' with the same probability i.e. $1/s_j^{(p)}$. Putting all of this together, we get that:

$$\mathbb{P}\left[C_i\right] = \Theta(1)\cdot\frac{1}{s_{j'}^{(p)}}\cdot\frac{s_{j'}^{(p)}\|B_i\|_2}{\left\|B^{(p)}\right\|_{2,1}}\cdot\frac{\left\|B^{(p)}\right\|_{2,1}}{\|B\|_{2,1}} = \Theta(1)\frac{\|B_i\|_2}{\|B\|_{2,1}}$$

To conclude, the sampling procedure samples noisy rows B_i such that i is sampled with probability at most a multiplicative constant from its probability under the distribution $\left(\frac{\|B_1\|_2}{\|B\|_{2,1}}, \dots \frac{\|B_n\|_2}{\|B\|_{2,1}}\right)$. Sampling $O(s)$ times guarantees that each row appears in the sampled set with at least the probability it would appear in s samples of the idealized process.

Finally note that H-SKETCH, and the $\|\cdot\|_{2,1}$-norm estimation procedure of Lemma A.4.1, can be implemented as oblivious linear sketches. Since no two distinct pieces share any rows in common, all matrix multiplications can be done in $\mathrm{nnz}(B) + d \cdot \mathrm{poly}(s/(\delta\nu))$ time. Furthermore they can be implemented in the streaming model with $d \cdot \mathrm{poly}(s/(\delta\nu))$ bits of space. $\square$

Setting $b = \log(nd)$, $\delta = 1/100$, $\nu = \frac{\epsilon}{100k\beta}$ and $s = \beta \,\mathrm{poly}(k/\epsilon)$, it follows that P contains $\beta \,\mathrm{poly}(k/\epsilon)$ samples from $\mathcal{P}^*(A(\mathcal{I} - Q))$ with probability $99/100$. By Lemma 4.2.1 and a union bound, the projection matrix of $\mathrm{RowSpan}(Q) \cup \mathrm{RowSpan}(P)$ is an $(\alpha + \beta \,\mathrm{poly}(k/\epsilon), (1+\epsilon))$-coreset for A with probability $49/50$. BOOTSTRAPCORESET takes total time $O(\mathrm{nnz}(A)) + O(d \,\mathrm{poly}(\beta \log(nd)k/\epsilon))$ and space $O(d \,\mathrm{poly}(\beta \log(nd)k/\epsilon))$. $\square$

Note that in our main algorithm we cannot compute the projection $A(\mathcal{I} - Q)$ until the after the stream is finished. Fortunately, since H is oblivious, we can right multiply HA by $(\mathcal{I} - Q)$ once Q is available, and only then perform the sampling procedure $\mathcal{P}$.

4.3 Right Dimension Reduction

We show how to reduce the right dimension of our problem. This result is used in both Algorithm 1 and Algorithm 2.

Theorem 4.3. *If $U^\intercal$ is an (α, β)-coreset, $S \in \mathbb{R}^{\alpha \cdot \mathrm{poly}(k/\epsilon) \times d}$ is a CountSketch matrix composed with a matrix of i.i.d. Gaussians, and $R \in \mathbb{R}^{d \times \mathrm{poly}(k/\epsilon)}$ is a CountSketch matrix, then with probability* $49/50$*, if $X' = \mathrm{argmin}_X \|AS^\intercal - AR^\intercal X U^\intercal S^\intercal\|_{2,1}$ then:*

$$\left\|A - AR^\intercal X' U^\intercal\right\|_{2,1} \leq (1 + O(\epsilon)) \min_{X \text{ rank } k} \|A - AXU^\intercal\|_{2,1}$$

Proof. Here we apply reasoning similar to that at the bottom of page 32 from [6]. We need a couple of lemmas from [6].

Lemma 4.3.1 (Lemma 30 from [6])**.** *If S is a lopsided embedding for (B, D), then if X'' has the property that $\|SBX'' - SD\|_{1,2} \leq \kappa \min_{X \in \mathcal{C}} \|SBX - SD\|_{1,2}$ for some κ, then:*

$$\left\|BX'' - D\right\|_{1,2} \leq \kappa(1 + 3\epsilon) \min_{X \in \mathcal{C}} \|BX - D\|_{1,2}$$

Lemma 4.3.2. *If $U \in \mathbb{R}^{d \times \alpha}$ and $R \in \mathbb{R}^{\mathrm{poly}(k/\epsilon) \times d}$ is a CountSketch matrix, then with probability* $99/100$*:*

$$\min_{X \text{ rank } k} \|A - AR^\intercal XU^\intercal\|_{2,1} \leq (1 + 3\epsilon) \min_{X \text{ rank } k} \|A - AXU^\intercal\|_{2,1}$$

Proof. Let $V^* = \mathrm{argmin}_{V \text{ rank } k} \|UV - A^\intercal\|_{1,2}$ and let $V = V_1 V_2$ be its rank factorization. Applying Lemmas 4.1.1 and 4.3.1, R is a lopsided embedding for $(UV_1, A^\intercal)$ with probability $99/100$. If $Y = \mathrm{argmin}_{Y \text{ rank } k} \|R(UV_1 Y - A^\intercal)\|_{1,2}$ then:

$$\|UV_1 Y - A^\intercal\|_{2,1} \leq (1 + 3\epsilon) \|UV^* - A^\intercal\|_{1,2} \leq (1 + 3\epsilon) \min_{X \text{ rank } k} \|A - AXU^\intercal\|_{2,1}$$

But $Y = (RUV_1)^- RA^\intercal$, and taking transposes this means that:

$$\min_{X \text{ rank } k} \|A - AR^\intercal X U^\intercal\|_{2,1} \leq \left\|A - AR^\intercal((RUV_1)^-)^\intercal V_1^\intercal U^\intercal\right\|_{2,1} \leq (1+3\epsilon) \min_{X \text{ rank } k} \|A - AXU^\intercal\|_{2,1}$$

□

From the last lemma, a solution to $\min_{X \text{ rank } k} \|A - AR^\intercal XU^\intercal\|_{2,1}$ will yield a $(1+O(\epsilon))$-approximate solution to the problem $\min_{X \text{ rank } k} \|A - AXU^\intercal\|_{2,1}$. Lemma 4.3.2 holds with probability $99/100$. Applying Lemma 4.1.1, with probability $99/100$, $S \in \mathbb{R}^{d \times \alpha \operatorname{poly}(k/\epsilon)}$ CountSketch composed with a Gaussian is a lopsided embedding for $(U, A^\intercal)$. Union bounding over these events, and applying Lemma 4.3.1 with $\mathcal{C}$ as the set of matrices in RowSpan$(RA^\intercal)$ proves the claim with probability $49/50$. □

4.4 Left Dimension Reduction

We show how to reduce the left dimension of our problem. Together with results from Section 4.3, this preserves the solution to X^* to within a coarse $\sqrt{\log d} \log\log d \cdot \operatorname{poly}(k/\epsilon)$ factor.

Theorem 4.4. *Suppose the matrices S_1, R_1 and U_1 are as in Algorithm 1. If $C_1 \in \mathbb{R}^{\operatorname{poly}(k/\epsilon) \times n}$ is a Sparse Cauchy matrix, and $G_1 \in \mathbb{R}^{\log d \operatorname{poly}(k/\epsilon) \times \log d \operatorname{poly}(k/\epsilon)}$ is a matrix of appropriately scaled i.i.d. Gaussians (as in Fact 4.1), and*

$$\hat{X} = \operatorname*{argmin}_{X \text{ rank } k} \|C_1 A S_1^\intercal G_1 - C_1 A R_1^\intercal X U_1^\intercal S_1^\intercal G_1\|_F$$

then with probability $24/25$*:* $\left\|AS_1^\intercal - AR_1^\intercal \hat{X} U_1^\intercal S_1^\intercal\right\|_{2,1} \leq \sqrt{\log d} \log\log d \cdot \operatorname{poly}(k/\epsilon) \cdot \Delta^*$

Proof. Define E_1 to be the event that the condition in Dvoretzky's theorem is satisfied, E_2 to be the event that Fact 4.1.1 holds for $D = AR_1^\intercal$, and E_3 to be the event that Fact 4.1.2 holds for $B = (AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal)G_1$. E_1 holds w.h.p., E_2, E_3 each separately hold with probability $99/100$ (for a suitable choice of K). By a union bound, they all hold simultaneously with probability at least $24/25$. Conditioned on this happening:

$$\left\|AS_1^\intercal - AR_1^\intercal \hat{X} U_1^\intercal S_1^\intercal\right\|_{2,1} \leq \|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} + \left\|AR(X^* - \hat{X})U_1^\intercal S_1^\intercal\right\|_{2,1} \tag{1}$$

$$\leq \|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} + \operatorname{poly}(k/\epsilon)\left\|CAR(X^* - \hat{X})U_1^\intercal S_1^\intercal G_1\right\|_{1,1} \tag{2}$$

$$\leq \operatorname{poly}(k/\epsilon)\begin{bmatrix} & \|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} + \|C(A - AR_1^\intercal X^* U_1^\intercal)S_1^\intercal G_1\|_{1,1} \\ + & \left\|C(A - AR_1^\intercal \hat{X} U_1^\intercal)S_1^\intercal G_1\right\|_{1,1} \end{bmatrix} \tag{3}$$

$$\leq \operatorname{poly}(k/\epsilon)\begin{bmatrix} & \|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} + \|C(AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal)G_1\|_{1,1} \\ + & \sqrt{\log d}\left\|C(A - AR_1^\intercal \hat{X} U_1^\intercal)S_1^\intercal G_1\right\|_F \end{bmatrix} \tag{4}$$

$$\leq \operatorname{poly}(k/\epsilon)\left[\|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} + \sqrt{\log d}\,\|C(AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal)G_1\|_{1,1}\right] \tag{5}$$

$$\leq \operatorname{poly}(k/\epsilon)\begin{bmatrix} & \|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} \\ + & \sqrt{\log d}\log\log d\,\|(AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal)G_1\|_{1,1} \end{bmatrix} \tag{6}$$

$$\leq \sqrt{\log d}\log\log d \operatorname{poly}(k/\epsilon)\,\|AS_1^\intercal - AR_1^\intercal X^* U_1^\intercal S_1^\intercal\|_{2,1} \tag{7}$$

(1) and (3) hold by triangle inequality, (2) since E_1 and E_2 hold, (4) comes from the relationship between the 1-norm and 2-norm, (5) since $\hat{X}$ is the minimizer of the expression $\|C_1(A - C_1AR_1^\intercal XU_1^\intercal)S_1^\intercal G_1\|_F$ and p-norms decrease with p, (6) since E_3 holds, and (7) by E_1 again. □

The rank constrained Frobenius norm minimization problem above has a closed form solution.

Fact 4.2. *For a matrix M, let $U_M\Sigma_M V_M^\intercal$ be the SVD of M. Then:*

$$\operatorname*{argmin}_{X \text{ rank } k} \|Y - ZXW\|_F = Z^-[U_Z U_Z^\intercal Y V_W V_W^\intercal]_k W^-$$

5 $(1+\epsilon)$-Approximation

5.1 Left Dimension Reduction

The following median based embedding allows us to reduce the left dimension of our problem. Together with results from Section 4.3, this preserves the solution to X^* to within a $(1+O(\epsilon))$ factor.

Theorem 5.1. *Suppose S_2, R_2 and U_2 are as in Algorithm 2. If $C_2 \in \mathbb{R}^{\sqrt{\log d}\log\log d\,\mathrm{poly}(k/\epsilon)\times n}$ is a Cauchy matrix, and $G_2 \in \mathbb{R}^{\sqrt{\log d}\log\log d\,\mathrm{poly}(k/\epsilon)\times\sqrt{\log d}\log\log d\,\mathrm{poly}(k/\epsilon)}$ is a matrix of appropriately scaled i.i.d. Gaussians (as in Fact 4.1), and:*

$$\hat{X}' = \operatorname*{argmin}_{X \text{ rank } k} \|C_2AS_2^\intercal G_2 - C_2AR_2^\intercal XU_2^\intercal S_2^\intercal G_2\|_{\mathrm{med},1}$$

then with probability $99/100$:

$$\left\|AS_2^\intercal G_2 - AR_2^\intercal \hat{X}' U_2^\intercal S_2^\intercal G_2\right\|_{1,1} \le (1+\epsilon)\min_{X \text{ rank } k} \|AS_2^\intercal G_2 - AR_2^\intercal XU_2^\intercal S_2^\intercal G_2\|_{1,1}$$

Proof. The following fact is known:

Fact 5.1 (Lemma F.1 from [2])**.** *Let L be a t dimensional subspace of $\mathbb{R}^s$. Let $C \in \mathbb{R}^{m\times s}$ be a matrix with $m = O\left(\frac{1}{\epsilon^2}t\log\frac{t}{\epsilon}\right)$ and i.i.d. standard Cauchy entries. With probability $99/100$, for all $x \in L$ we have*

$$(1-\epsilon)\|x\|_1 \le \|Cx\|_{\mathrm{med}} \le (1+\epsilon)\|x\|_1$$

The theorem statement is simply the lemma applied to $L = \mathrm{ColSpan}\left([AS_2^\intercal \mid AR_2^\intercal]\right)$. □

5.2 Solving Small Instances

Given problems of the form $\hat{X} = \operatorname{argmin}_{X \text{ rank } k} \|Y - ZXW\|_{\mathrm{med},1}$, we leverage an algorithm for checking the feasibility of a system of polynomial inequalities as a black box.

Lemma 5.1. *[3] Given a set $K = \{\beta_1, \cdots, \beta_s\}$ of polynomials of degree d in k variables with coefficients in $\mathbb{R}$, the problem of deciding whether there exist $X_1, \cdots X_k \in \mathbb{R}$ for which $\beta_i(X_1, \cdots, X_k) \ge 0$ for all $i \in [s]$ can be solved deterministically with $(sd)^{O(k)}$ arithmetic operations over $\mathbb{R}$.*

Theorem 5.2. *Fix any $\epsilon \in (0,1)$ and $k \in [0, \min(m_1, m_2)]$. Let $Y \in \mathbb{R}^{n\times m''}$, $Z \in \mathbb{R}^{n\times m_1}$, and $W \in \mathbb{R}^{m_2\times m''}$ be any matrices. Let $C \in \mathbb{R}^{m'\times n}$ be a matrix of i.i.d. Cauchy random variables, and $G \in \mathbb{R}^{m''\times m''\,\mathrm{poly}(1/\epsilon)}$ be a matrix of scaled i.i.d. Gaussian random variables. Then conditioned on C satisfying Fact 5.1 for the adjoined matrix $[Y, Z]$ and G satisfying the condition of Fact 4.1, a rank-k projection matrix X can be found that minimizes $\|C(Y - ZXW)G\|_{\mathrm{med},1}$ up to a $(1+\epsilon)$-factor in time $\mathrm{poly}(m'm''/\epsilon)^{O(mk)+(m''+m')\,\mathrm{poly}(1/\epsilon)}$, where $m = \max(m_1, m_2)$.*

Proof. We write $X = PQ$, where P is $m_1 \times k$ and Q is $k \times m_2$, to ensure that X is rank $\leq k$.

Guess a permutation π_j for each column j of $C(ZXW - Y)G$ and define constraints enforcing the permutation. Since the (i,j)-th entry of the matrix is $\sum_{k,\ell}(CZ)_{ik}X_{k\ell}(WG)_{\ell j} - (CYG)_{ij}$ these constraints are of the form $((C(ZXW - Y)G)_{\pi_j(i)j})^2 \leq ((C(ZXW - Y)G)_{\pi_j(i+1)j})^2$. Then define the median of the j-th column to be:

$$M_j = \left(|(C(ZXW - Y)G)_{\pi_j(\lfloor m''/2 \rfloor)j}| + |(C(ZXW - Y)G)_{\pi_j(\lceil m''/2 \rceil)j}| \right) /2$$

which can be expressed via polynomial constraints. Thus we have $O(mk) + m'' \operatorname{poly}(1/\epsilon)$ variables in our polynomial inequality system, $O(mk)$ variables to describe P and Q, and $m'' \operatorname{poly}(1/\epsilon)$ variables to describe the column medians M_j. We have $\operatorname{poly}(m'm''/\epsilon)$ constraints, each involving polynomials of $O(1)$ degree. By Lemma 5.1, checking the feasibility of this system takes time $\operatorname{poly}(m'm''/\epsilon)^{O(mk)+m'' \operatorname{poly}(1/\epsilon)}$.

We can minimize the objective $\sum_j M_j$ using binary search. This requires a lower bound on the objective value, which we can get by noting from Fact 5.1 that:

$$\min_X \|CZXWG - CYG\|_{\text{med},1} \geq (1-\epsilon) \min_X \|ZXW - Y\|_{1,1} \geq (1-\epsilon) \min_X \|ZXW - Y\|_{2,1}$$

As in the proof of Theorem 51 in [6], when the solution is constrained to be rank k, the right hand side is lower bounded by $\frac{1}{\operatorname{poly}(d)}(\sigma_{k+1}(Y))^{1/2}$ (where $\sigma_{k+1}(Y)$ is the $k+1$st singular value of Y), which itself is lower bounded by $\left(\frac{1}{\exp(\operatorname{poly}(m'm''))}\right)^k$. Thus we can do binary search in $\operatorname{poly}(m'm''/\epsilon)$ steps.

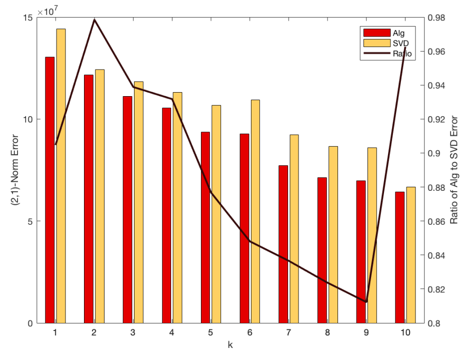
Finally, since there are $m'' \cdot m'!$ possible permutation guesses, the entire procedure takes time $\operatorname{poly}(m'm''/\epsilon)^{O(mk)+(m''+m')\operatorname{poly}(1/\epsilon)}$. $\square$

We remark that if, as we do in our algorithm, we set the all the parameters m, m' and m'' to be $\log\log d\sqrt{\log d}\cdot \operatorname{poly}(k/\epsilon)$, we can write the runtime of this step (Line 9 of Algorithm 2) as $(n+d)\operatorname{poly}(k/\epsilon)+\exp(\operatorname{poly}(k/\epsilon)))$. If $\operatorname{poly}(k/\epsilon) \leq \sqrt{\log d}/(\log\log d)^2$, then this step is captured in the $(n+d)\operatorname{poly}(k/\epsilon)$ term. Otherwise this step is captured in the $\exp(\operatorname{poly}(k/\epsilon))$ term.

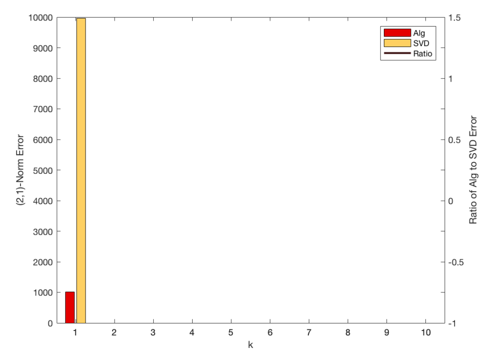
6 Experiments

In this section we empirically demonstrate the effectiveness of CoarseApprox compared to the truncated SVD. We experiment on synthetic and real world data sets. Since the algorithm is randomized, we run it 20 times and take the best performing run. For a fair comparison, we use an input sparsity time approximate SVD as in [5].

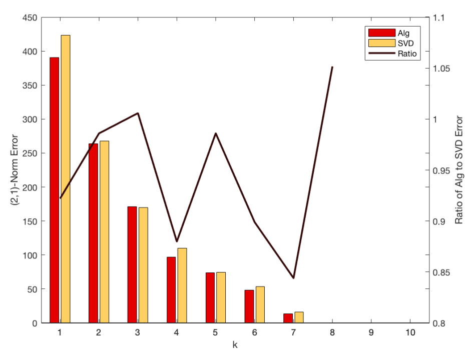
For the synthetic data, we use two example matrices all of dimension 1000×100. In Figure 1a we use a Rank-3 matrix with additional large outlier noise. First we sample U random 100×3 matrix and V random 3×10 matrix. Then we create a random sparse matrix W with each entry nonzero with probability 0.9999 and then scaled by a uniform random variable between 0 and $10000 \cdot n$. We use $10 \cdot UV + W$. In Figure 1b we create a simple Rank-2 matrix with a large outlier. The first row is n followed by all zeros. All subsequent rows are 0 followed by all ones.



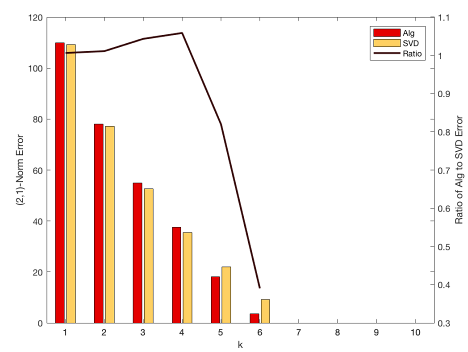
(a) Random Rank-3 Matrix Plus Large Outliers



(b) Large Outlier Rank-2 Matrix



(c) Glass



(d) E. Coli

Figure 1: Comparison of Algorithm 1 on synthetic and real world examples.

While the approximation guarantee of CoarseApprox is weak, we find that it performs well against the SVD baseline in practice on our examples, namely when the data has large outliers rows. The second example in particular serves as a good demonstration of the robustness of the (2,1)-norm to outliers in comparison to the Frobenius norm. When $k = 1$, the truncated SVD which is the Frobenius norm minimizer recovers the first row of large magnitude, whereas our algorithm recovers the subsequent rows. Note that both our algorithm and the SVD recover the matrix exactly when k is greater than or equal to rank.

We have additionally compared our algorithm against the SVD on two real world datasets from the UCI Machine Learning Repository: Glass is a 214×9 matrix representing attributes of glass samples, and E.Coli is a 336×7 matrix representing attributes of various proteins. For this set of experiments, we use a heuristic extension of our algorithm that performs well in practice. After running CoarseApprox, we iterate solving $Y_t = \min_Y \|CAS^\intercal G - YZ_{t-1}\|_{1,1}$ and $Z_t = \min_Z \|CAS^\intercal G - Y_t Z\|_{1,1}$ (via Iteratively Reweighted Least Squares for speed). Finally we output the rank k Frobenius minimizer constrained to RowSpace$(Y_t Z_t)$. In Figure 1c we consistently outperform the SVD by between 5% and 15% for nearly all k, and nearly match the SVD otherwise. In Figure 1d we are worse than the SVD by no more than 5% for $k = 1$ to 4, and beat the SVD by up to 50% for $k = 5$ and 6. We have additionally implemented a greedy column selection algorithm which performs worse than the SVD on all of our datasets.

Acknowledgements: We would like to thank Ainesh Bakshi for many helpful discussions. D. Woodruff thanks partial support from the National Science Foundation under Grant No. CCF-1815840. Part of this work was also done while D. Woodruff was visiting the Simons Institute for the Theory of Computing.

References

[1] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David P. Woodruff. Efficient sketches for earth-mover distance, with applications. *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 324–330, 2009.

[2] Arturs Backurs, Piotr Indyk, Ilya P. Razenshteyn, and David P. Woodruff. Nearly-optimal bounds for sparse recovery in generic norms, with applications to k-median sketching. In *SODA*, 2016.

[3] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. In *J. ACM*, 1994.

[4] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 205–214, 2009.

[5] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 81–90, New York, NY, USA, 2013. ACM.

[6] Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 310–329, 2015.

[7] Kenneth L. Clarkson and David P. Woodruff. Sketching for m-estimators: A unified approach to robust regression. In *SODA*, 2015.

[8] Amit Deshpande, Madhur Tulsiani, and Nisheeth K. Vishnoi. Algorithms and hardness for subspace approximation. In *SODA*, 2011.

[9] Amit Deshpande and Kasturi R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 641–650, 2007.

[10] Amit Deshpande and Kasturi R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In *STOC*, 2007.

[11] Chris H. Q. Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R1-pca: rotational invariant l1-norm principal component analysis for robust subspace factorization. In *ICML*, 2006.

[12] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, 2011.

[13] Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 630–649, 2010.

[14] Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *SODA*, 2010.

[15] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM J. Comput.*, 45(5):1762–1792, 2016.

[16] Venkatesan Guruswami, Prasad Raghavendra, Rishi Saket, and Yi Wu. Bypassing ugc from some optimal geometric inapproximability results. *ACM Trans. Algorithms*, 12:6:1–6:25, 2010.

[17] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 10–, Washington, DC, USA, 2001. IEEE Computer Society.

[18] Ravi Kannan, Santosh Vempala, and David P. Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, pages 1040–1057, 2014.

[19] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. *CoRR*, abs/1210.3135, 2012.

[20] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005.

[21] Nariankadu D. Shyamalkumar and Kasturi R. Varadarajan. Efficient subspace approximation algorithms. *Discrete & Computational Geometry*, 47(1):44–63, 2012.

[22] Christian Sohler and David P. Woodruff. Subspace embeddings for the l1-norm with applications. In *STOC*, 2011.

[23] Zhao Song, David P. Woodruff, and Peilin Zhong. Low rank approximation with entrywise l1-norm error. *CoRR*, abs/1611.00898, 2016.

[24] Elad Verbin and Qin Zhang. Rademacher-sketch: A dimensionality-reducing embedding for sum-product norms, with an application to earth-mover distance. In *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part I*, ICALP'12, pages 834–845, Berlin, Heidelberg, 2012. Springer-Verlag.

[25] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

A Elided Proofs

A.1 H-SKETCH Guarantees

Note that in the proofs that follow, we require that our sketching matrices have small representations. For each level j of each H-SKETCH, we can store the matrix as three different $O(1)$-wise independent hashes:

1. $P_j : [n] \to \{0, 1\}$ determines whether the row is subsampled
2. $h_j : [n] \to [w]$ determines the bucket
3. $\varepsilon_j : [n] \to \pm 1$ determines the sign.

Since our sketching matrices are stored with small seeds and do not guarantee full independence of the entries of the matrix, we cannot easily use Chernoff-Hoeffding bounds.

For ease of notation below, we define a parameter $C = O(\frac{s \log n}{\delta' \nu})$ with a sufficiently large constant.

For any level j, we separate the rows of B into "light" and "heavy" rows with respect to j:

$$S_L^j = S_{\geq j+\log(C^6)+1}$$
$$S_H^j = S_{\leq j+\log(C^6)}$$

We first analyze the contribution of light and heavy elements to level j of the sketch.

Lemma A.1 (Noise from light elements is small)**.** *For any $j \in [\ell]$, with probability $1 - O(1)/\log n$:*

$$\max_v \sum_{i \in S_L^j} [\![i \in I_j]\!][\![h_j(i) = v]\!] \cdot \|B_i\|_2 \leq \frac{0.1 \cdot \nu \cdot T_j}{C^2}$$

Proof. Define the variables $Z_i^v = [\![i \in J_j]\!] \cdot [\![h_j(i) = v]\!] \cdot \varepsilon_j(i) \cdot \|B_i\|_2$, where $i \in S_L^j$. For a fixed v, we first study σ^2, the second moment of $\sum_{i \in S_L^j} Z_i^v$.

$$\sigma^2 = \mathbb{E}\left[\left(\sum_i [\![i \in J_j]\!] \cdot [\![h_j(i) = v]\!] \cdot \varepsilon_j(i) \cdot \|B_i\|_2\right)^2\right] \tag{1}$$
$$= \sum_{i,i' \in S_L^j} \mathbb{E}\left[[\![i \in J_j]\!][\![i' \in J_j]\!] \cdot [\![h_j(i) = v]\!][\![h_j(i') = v]\!]\right] \cdot \mathbb{E}\left[\varepsilon_j(i)\varepsilon_j(i')\right] \cdot \|B_i\|_2 \cdot \|B_{i'}\|_2 \tag{2}$$
$$= \sum_{i \in S_L^j} \frac{p_j}{w} \cdot \|B_i\|_2^2 \tag{3}$$
$$\leq \frac{p_j}{w} \max_i(\|B_i\|_2) \cdot \sum_i \|B_i\|_2 \tag{4}$$
$$\leq \frac{T_j^2}{wC^6} \tag{5}$$

In step (2) we used the fact that the two variables $[\![i \in J_j]\!][\![i' \in J_j]\!][\![h_j(i) = v]\!][\![h_j(i') = v]\!]$ and $\varepsilon_j(i)\varepsilon_j(i')$ are independent. In step (3), we used the 2-wise independence of ε_j and the fact that $\varepsilon_j(i)\varepsilon_j(i') = 1$ if $i = i'$ and 0 otherwise. In step (5) we used the fact that $\frac{p_j}{w}\|B\|_{2,1} \leq \frac{M}{w2^j} = \frac{T_j}{w}$ and:

$$\|B_i\|_2 \leq T_{j+\log(C^6)} = T_j/2^{\log(C^6)} = \frac{T_j}{C^6}$$

By Chebyshev's inequality:

$$\mathbb{P}\left[\sum_{i \in S_L^j} Z_i \geq \frac{0.1 \cdot \nu \cdot T_j}{C^2}\right] \leq \frac{O(1)}{w \log n}$$

The desired bound follows by a union bound over the w buckets. □

Lemma A.2 (Heavy Elements do not collide)**.** *For any level $j \in [\ell]$, with probability at least $1 - \frac{O(1)}{\log n}$ no two elements from S_H^j hash to the same bucket.*

Proof. We can bound the expected number of samples:

$$\mathbb{E}\left[\sum_{i \in S_H^j} [\![i \in I_j]\!]\right] \leq \left(\sum_{j'=1}^{j+\log(C^6)} s_{j'}\right) \cdot p_j \leq 2^{j+\log(C^6)+1} \cdot 2^{-j} = 2C^6$$

Thus, by Markov's bound,

$$\mathbb{P}\left[\sum_{i\in S_H^j} [\![i \in J_j]\!] > C^7\right] \leq \frac{O(1)}{\log n}$$

Thus, no more than C^7 heavy elements are subsampled with high probability. Conditioned on this happening, we can bound the probability that any two of them hash into the same bucket by:

$$w^{-1} \cdot \binom{C^7}{2} \leq \frac{C^{14}}{w} \leq \frac{1}{\log n}$$

since we chose $w = O(C^{15})$. The claim follows by a union bound over the two $1/\log n$ probability events. $\square$

Lemma A.3 (Level estimates)**.** *For any important level $j \in \mathcal{J}$, let $\hat{s}_j$ be the number of buckets in $H^{(k)}$ with norm in the interval $[(1-\nu)T_j, (2+\nu)T_j]$, where $k = \max\left(0, j - \log C^2\right)$. Let $\tilde{s}_j = 2\hat{s}_j p_k^{-1}$. Then with probability $1 - O(1)/\log n$ it holds that $\tilde{s}_j \in [s_j, 4s_j]$.*

Proof. By Lemma A.2, with probability $1 - O(1)/\log n$ the rows in S_H^k do not collide, and by Lemma A.1, with probability $1 - O(1)/\log n$ the contribution of elements in S_L^k is less than $\frac{0.1\nu T_k}{C^2} = 0.1\nu T_j$ (due to the relationship between j and k). Conditioned on this holding, $\hat{s}_j = J_k^j$, where J_k^j is the number of elements of S_j subsampled in level k.

If $k = 0$, all rows of S_j are subsampled, $p_k = 1$ and the claim is proved. Otherwise, we use a second moment method. $\mathbb{E}\left[J_k^j\right] = s_j p_k$ and additionally:

$$\begin{aligned}
\mathbb{E}\left[\left|J_k^j\right|^2\right] &= \mathbb{E}\left[\left(\sum_{i\in S_j} [\![i \in J_k^j]\!]\right)^2\right] \\
&= \sum_{i,i'\in S_j} \mathbb{E}\left[[\![i \in J_k^j]\!][\![i' \in J_k^j]\!]\right] \\
&= \sum_{i\in S_j} \mathbb{E}\left[[\![i \in J_k^j]\!]\right] + \sum_{i\neq i'\in S_j} \mathbb{E}\left[[\![i \in J_k^j]\!][\![i' \in J_k^j]\!]\right] \\
&\leq s_j p_k + s_j^2 p_k^2
\end{aligned}$$

Note that we only use the 2-wise independence of the subsampling function P_k. Thus $\left|J_k^j\right|$ has variance $\sigma^2 \leq s_j p_k$. Using Chebyshev's inequality:

$$\mathbb{P}\left[\left|\left|J_k^j\right| - s_j p_k\right| \geq \frac{s_j p_k}{2}\right] \leq \frac{2}{s_j p_k}$$

Since $s_j p_k \geq \log n$, the claim follows from a union bounding guaranteeing that Lemmas A.1 and A.2 hold together with the last event that the number of subsampled elements from S_j is within a factor of two of its expectation. $\square$

Corollary A.1 (At least one element is sampled)**.** *For any important level $j \in \mathcal{J}$, with probability at least $1 - O(1)/\log n$, at least one element of S_j is subsampled in J_k when $k = \max\left(0, j - \log C^2\right)$.*

A.2 High Probability $\|\cdot\|_{2,1}$ Estimation

Lemma A.4. *Given a matrix $B \in \mathbb{R}^{n\times d}$, there is an algorithm that with probability $1 - 1/(50t)$ outputs an estimate M such that $\|B\|_{2,1} \le M \le 20010\,\|B\|_{2,1}$. Furthermore this algorithm runs in time $O(nnz(B) + O(n+d)(\mathrm{poly}\log(ndt)))$ and can be implemented in the streaming model with $d \cdot \mathrm{poly}(\log(ndt))$ bits of space.*

Proof. First we prove an intermediate result:

Lemma A.4.1. *If S is a Count Sketch matrix with $O(t^2)$ rows, then with probability $1 - \frac{1}{100t}$, it holds that $\|B\|_{2,1}/2 \le \|BS^\intercal\|_{2,1} \le 2\,\|B\|_{2,1}$.*

Proof. For any fixed row i, if S has $O(1/\delta)$ rows, then with probability at least $(1-\delta)$ it holds that $\|B_i S^\intercal\|_2 \in (1 \pm 0.5)\,\|B_i\|_2$. For a proof of this, see e.g. Theorem 2.6 of [25] which shows that with probability $(1-\delta)$, S is a (1 ± 0.5) subspace embedding for B_i^T.

Let T be the set of rows i for which $\|B_i S^\intercal\|_2 \notin (1 \pm 0.5)\,\|B_i\|_{2,1}$. Then:

$$\begin{aligned}\|BS^\intercal\|_{2,1} &= \sum_{i\in T}\|B_iS^\intercal\|_2 + \sum_{i\notin T}\|B_iS^\intercal\|_2\\ &\le \sum_{i\in T}\|B_iS^\intercal\|_2 + \sum_{i\notin T}\frac{3}{2}\|B_i\|_2\end{aligned}$$

Since $\mathbb{E}\left[\sum_{i\in T}\|B_iS^\intercal\|_2\right] \le \delta\,\|B\|_{2,1}$, by a Markov Bound:

$$\mathbb{P}\left[\sum_{i\in T}\|BS^\intercal\|_{2,1} \ge (3/2 + \gamma\delta)\,\|B\|_{2,1}\right] \le \frac{1}{\gamma}$$

For a lower bound, let $y_i = \begin{cases} 0 & \text{if } i \in T \\ \|B_i\|_2/2 & \text{if } i \notin T\end{cases}$, and let $z_i = \|B_i\|_2/2 - y_i$. Then $\mathbb{E}\left[\sum_i z_i\right] \le \frac{\delta\|B\|_{2,1}}{2}$.

Note that $\|BS^\intercal\|_{2,1} \ge \sum_{i\notin T}\frac{1}{2}\|B_i\|_2 = \sum_i y_i$. Again by a Markov Bound:

$$\begin{aligned}\mathbb{P}\left[\|BS^\intercal\|_{2,1} \le (1-\gamma\delta)\,\|B\|_{2,1}\right] &\le \mathbb{P}\left[\sum_i y_i \le (1-\gamma\delta)\,\|B\|_{2,1}\right]\\ &= \mathbb{P}\left[\sum_i z_i \ge \gamma\delta\,\|B\|_{2,1}\right]\\ &\le \frac{1}{\gamma}\end{aligned}$$

Setting $\gamma = 100t$ and $\delta = 1/10000t^2$, the claim is proved. □

Calculating $BS^\intercal$ in a stream normally requires $\Omega(n)$ bits of space which exceeds our desired space bound. However we can remedy this with some lemmas from [24].

By Sections 4.1 and 4.2 of [24], for any matrix B' with probability at least 0.9 it holds that:

$$\frac{2\,\|B'\|_{2,1}}{5} \le \left\|\text{H-SKETCH}(B')\right\|_{2,1} \le 2001\,\left\|B'\right\|_{2,1}$$

for sufficiently large $\|B'\|_{2,1}$. Consequently, if we repeat this $O(\log t)$ times and let M_0 be the median of these trials, then M_0 achieves the same guarantee but with probability $1 - 1/(100t)$.

Letting $B' = B$ would naively require time $O(\text{nnz}(B') \cdot \log t)$, which exceeds our desired time bounds. However, using Lemma A.4.1, if we set $B' = BS^\intercal$ where S is a Count Sketch matrix with $O(t^2)$ rows, we get a similar constant factor guarantee. By a union bound over the event that M_0 is a good estimator for $\|B'\|_{2,1} = \|BS^\intercal\|_{2,1}$ and the event that $\|BS^\intercal\|_{2,1}$ is itself a good estimator for $\|B\|_{2,1}$, with probability $1 - 1/(50t)$:

$$\frac{\|B\|_{2,1}}{5} \leq M_0 \leq 4002 \|B\|_{2,1}$$

Note that we can afford to store $O(\log t)$ copies of H-SKETCH$(BS^\intercal)$ in the stream. Outputting $M = 5 \cdot M_0$ yields the claim. $\square$