Near-Optimal Algorithms for Linear Algebra in the Current Matrix Multiplication Time

Nadiia Chepurko* Kenneth L. Clarkson† Praneeth Kacham‡ David P. Woodruff§

Abstract

In the numerical linear algebra community, it was suggested that to obtain nearly optimal bounds for various problems such as rank computation, finding a maximal linearly independent subset of columns (a *basis*), regression, or low-rank approximation, a natural way would be to resolve the main open question of Nelson and Nguyen (FOCS, 2013). This question is regarding the logarithmic factors in the sketching dimension of existing oblivious subspace embeddings that achieve constant-factor approximation. We show how to bypass this question using a refined sketching technique, and obtain optimal or nearly optimal bounds for these problems. A key technique we use is an explicit mapping of Indyk based on uncertainty principles and extractors, which after first applying known oblivious subspace embeddings, allows us to quickly spread out the mass of the vector so that sampling is now effective. We thereby avoid a logarithmic factor in the sketching dimension that is standard in bounds proven using the matrix Chernoff inequality. For the fundamental problems of rank computation and finding a basis, our algorithms improve Cheung, Kwok, and Lau (JACM, 2013), and are optimal to within a constant factor and a poly($\log\log(n)$)-factor, respectively. Further, for constant-factor regression and low-rank approximation we give the first optimal algorithms, for the current matrix multiplication exponent.

1 Introduction

We obtain several new results for fundamental problems in numerical linear algebra, in many cases removing, in particular, the *last* log factor to obtain a running time that is truly linear in the input sparsity, and with lower-order terms that are close to optimal. We note that the bottleneck in improving prior work, including such removal of last logarithmic factors, involved well-known conjectures to construct *Sparse Johnson-Lindenstrauss transforms* (see Conjecture 14 in [29]).

To sidestep these conjectures we introduce a new simple matrix sketching technique which allows for multiplication by a random sparse matrix whose randomly chosen nonzero entries are random signs. The key idea is to compose this matrix with an appropriate Flattening transform based on explicit embeddings of ℓ_2 into ℓ_1, together with OSNAP embeddings. Using this, we obtain the first oblivious subspace embedding for k-dimensional subspaces that has $o(k\log(k))$ rows and that can be applied to a matrix A in time asymptotically less than both $\mathtt{nnz}(A)\log k$ and $k^\omega \log k$, where $\mathtt{nnz}(A)$ is the number of nonzero entries in the matrix A, and $\omega \approx 2.37$ is the exponent of fast matrix multiplication [1]. This scheme removes a log factor that has thus far remained both a nuisance and an impediment to optimal algorithms. Our main embedding result is as follows:

Theorem 1.1. (Fast Subspace Embedding, informal Theorem 6.3) *Given an $n \times k$ matrix, there is a distribution $\mathcal{S}$ over matrices with $k\,\mathrm{poly}(\log\log k)$ rows such that, for $\boldsymbol{S} \sim \mathcal{S}$, with probability $\geq 99/100$, for all vectors $x \in \mathbb{R}^k$*

$$\|Ax\|_2 \leq \|\boldsymbol{S}Ax\|_2 \leq \exp(\mathrm{poly}(\log\log k))\|Ax\|_2.$$

For $\boldsymbol{S} \sim \mathcal{S}$, with probability $\geq 95/100$, the matrix $\boldsymbol{S}A$ can be computed in time $O(\gamma^{-1}\mathtt{nnz}(A) + k^{2+\gamma+o(1)})$ for any constant $\gamma > 0$.

[*] Massachusetts Institute of Technology. Email: nadiia@mit.edu.
[†] IBM Research Almaden. Email: klclarks@us.ibm.com.
[‡] Carnegie Mellon University. Email: pkacham@cs.cmu.edu.
[§] Carnegie Mellon University. Email: dwoodruf@cs.cmu.edu.

Using our subspace embedding, together with additional ideas, we obtain nearly optimal (up to $\log\log$ factors in the sub-linear terms) running times for fundamental problems in classical linear algebra including computing matrix rank, finding a set of linearly independent rows, and linear regression. Further, for regression and low-rank approximation, we obtain the first optimal algorithms for the current matrix multiplication exponent. We begin with least-squares regression:

Theorem 1.2. (Least-Squares Regression, informal Theorem 7.3) *Given a full rank $n \times k$ matrix A, $k \leq n$, and vector b, there exists an algorithm that computes $\hat{x}$ such that $\|A\hat{x} - b\|_2 \leq (1 + \varepsilon)\min_x \|Ax - b\|_2$ in time*

$$O\left(\frac{\mathtt{nnz}(A)}{\gamma} + k^\omega \operatorname{poly}(\log\log(k)) + \frac{1}{\operatorname{poly}(\varepsilon)} k^{2+o(1)} n^{\gamma+o(1)}\right)$$

for any constant $\gamma > 0$ small enough.

We note that for constant ε and $k = n^{\Omega(1)}$, the running time obtained is within a $\operatorname{poly}(\log\log(n))$ factor of optimal, for the current matrix multiplication constant. Further, it improves on prior work [4, 10, 14, 15, 27, 29] describing algorithms with an additional $\log(n)$ factor multiplying either the leading $\mathtt{nnz}(A)$ term, or that is $\mathtt{nnz}(A)$ time but has a $k^\omega \log k$ additive term or worse. We note that our additive term is only $k^\omega \operatorname{poly}(\log\log k)$, for the current matrix multiplication exponent ω, when $k = n^{\Omega(1)}$. Importantly, up to a $\operatorname{poly}(\log\log k)$ factor, our bound is best possible, and thus we remove the last logarithmic factor even in the additive term. As we explain more below, the issue with previous work is that to obtain a sketching dimension of $O(k)$, for constant ε, one needs either $\mathtt{nnz}(A)k$ time to directly perform a multiplication with a dense Sub-Gaussian matrix, or at least $k^\omega \log k$ time to compose a dense Sub-Gaussian sketch with a sparse sketch. We avoid this using our new subspace embedding, given by Theorem 6.3.

We note that simply sketching on the left with a CountSketch matrix and solving the sketched problem attains an optimal $O(\mathtt{nnz}(A))$ running time for $k = O(n^c)$ for a sufficiently small constant $c > 0$, and so our theorems are most interesting when $k = \Omega(n^c)$.

Next, we show a similar result holds for low-rank approximation (LRA):

Theorem 1.3. (LRA in Current Matrix Multiplication Time, informal Theorem 7.8) *Given $\varepsilon > 0$, an $n \times d$ matrix A and $k \leq \min(n, d)$, $k = \max(n, d)^{\Omega(1)}$, there exists an algorithm that runs in*

$$O\left(\mathtt{nnz}(A) + \frac{(n+d)k^{\omega-1}}{\varepsilon} + \frac{(n+d)k^{1.01}}{\varepsilon} + \operatorname{poly}(\varepsilon^{-1}k)\right)$$

time and outputs two matrices $V \in \mathbb{R}^{n\times k}$ and $\tilde{X} \in \mathbb{R}^{k\times d}$, with $V^\top V = I_k$, such that

$$\|A - V \cdot \tilde{X}\|_\mathsf{F} \leq (1+\varepsilon)\|A - [A]_k\|_\mathsf{F}.$$

For the current matrix multiplication exponent, the running time is $O(\mathtt{nnz}(A) + (n + d)k^{\omega-1})$ for constant ε. In contrast, existing low rank approximation algorithms [4, 10, 13, 14, 15, 16, 27, 29] take time at least $\mathtt{nnz}(A)\log n$ or $dk^{\omega-1}\log k$ or worse. Thus, as with least squares regression, we remove the last logarithmic factor in both the $\mathtt{nnz}(A)$ term and the leading additive term.

We also give constructions of $1 + \varepsilon$ subspace embeddings with $O(k\log(k)/\varepsilon^2)$ rows that have better running times than earlier subspace embeddings with $O(k\log(k)/\varepsilon^2)$ rows, such as approximate leverage score sampling and OSNAP embeddings.

Theorem 1.4. (Subspace Embeddings, informal Theorem 7.2) *Given a matrix $A \in \mathbb{R}^{n\times k}$, there is a* non-oblivious *subspace embedding $\boldsymbol{S}$ with $O(k\log(k)/\varepsilon^2)$ rows that can be applied to the matrix A in time $O(\mathtt{nnz}(A) + k^\omega \operatorname{poly}(\log\log k) + \operatorname{poly}(\varepsilon^{-1})k^{2.1+o(1)})$ for $k = n^{\Omega(1)}$.*

Finally, we obtain faster algorithms for computing the rank of a matrix and finding a full-rank set of rows.

Theorem 1.5. (Matrix Rank and Finding a Basis, informal Theorem 7.6 and 7.7) *Given an $n \times d$ matrix A, there exists a randomized algorithm to compute $k = \operatorname{rank}(A)$ in $O(\mathtt{nnz}(A) + k^\omega)$ time, where ω is the matrix multiplication constant. Further, the algorithm can find a set of k linearly independent rows in $O(\mathtt{nnz}(A) + k^\omega \log\log(n))$ time.*

We note that this result improves prior work by Cheung et al. [8], in the case of matrices with real numbers, who obtain an $O(\mathtt{nnz}(A)\log(k)+k^{\omega})$ time algorithm to compute matrix rank and an $O(\log(n)(\mathtt{nnz}(A)+k^{\omega}))$ time algorithm to find a full-rank set of rows.

The following table lists our running times for $k \le n$ and $k = n^{\Omega(1)}$, assuming $\omega > 2$, and putting some terms to constant values (such as 2.1 instead of $2+\gamma$). See theorem statements for exact running times.

Application	Running time (up to constant factors)
ε Subspace Embeddings	$\mathtt{nnz}(A) + \varepsilon^{-3}k^{2.1+o(1)} + k^{\omega}\operatorname{poly}(\log\log(k))$
ε approximate linear regression	$\mathtt{nnz}(A) + \varepsilon^{-3}k^{2.1+o(1)} + k^{\omega}\operatorname{poly}(\log\log(k))$
Linearly Independent Rows	$\mathtt{nnz}(A) + k^{\omega}\operatorname{poly}(\log\log(k)) + k^{2+o(1)}$
0.01 Low-Rank Approximation	$\mathtt{nnz}(A) + (n+d)k^{\omega-1}$

2 Related Work

Matrix Sketching. The *sketch and solve* paradigm [9, 38] was designed to reduce the dimensionality of a problem, while maintaining enough structure such that a solution to the smaller problem remains an approximate solution to the original one. This approach has been pivotal in speeding up basic linear algebra primitives such as least-squares regression [9, 31, 33], ℓ_p regression [12, 37], low-rank approximation [16, 26, 29], linear and semi-definite programming [17, 23, 24], solving non-convex optimization problems such as ℓ_p low-rank approximation [3, 34, 35], and training neural networks [2, 7]. For a comprehensive overview we refer the reader to the aforementioned papers and citations therein. Several applications use rank computation, finding a full rank subset of rows/columns, leverage score sampling, and computing subspace embeddings, as key algorithmic primitives. In addition to being used as a black box, we believe our techniques will be useful in sharpening bounds for several such applications.

3 Preliminaries

Computational Model Throughout the paper, we work with matrices having real numbers and assume that all elementary arithmetic operations on real numbers can be computed in $O(1)$ time.

Let A^{+} denote the Moore-Penrose pseudo-inverse of matrix $A \in \mathbb{R}^{n\times d}$, equal to $V\Sigma^{-1}U^{\top}$ when A has "thin" Singular Value Decomposition (SVD) $A = U\Sigma V^{\top}$, so that Σ is a square invertible matrix. We note that AA^{+} is the projection matrix onto the column span of the matrix A. Let $\|A\|_2$ denote the spectral norm ($\ell_2 \to \ell_2$ operator norm) of A and $\|A\|_{\mathsf{F}}$ denote the Frobenius norm $(\sum_{i,j} A_{ij}^2)^{1/2}$. Let $\kappa(A) = \|A^{+}\|_2\|A\|_2$ denote the condition number of A. We write $a \pm b$ to denote the set $\{c \mid |c-a| \le |b|\}$, and $c = a \pm b$ to denote the condition that c is in the set $a \pm b$. Let $[m] = \{1 \ldots m\}$ for an integer $m \ge 1$. For $i \in [n]$, A_{i*} denotes the i-th row of A and for $j \in [d]$, A_{*j} denotes the j-th column of A. We use bold symbols such as $\boldsymbol{A}$, $\boldsymbol{S}$ to emphasize that these objects are explicitly sampled from an appropriate distribution.

As mentioned, $\mathtt{nnz}(A)$ is the number of nonzero entries of A, and we assume $\mathtt{nnz}(A) \ge n$, i.e., there are no rows composed entirely of zeros. We let $[A]_k$ denote the best rank-k approximation to A in Frobenius norm and operator norm. Further, for an $n \times d$ matrix A and $S \subseteq [n]$, we use the notation A_S to denote the restriction of the rows of A to the subset indexed by S, and for $S \subseteq [d]$ we use the notation A^S to denote the restriction of the columns of A to the subset indexed by S.

Let n^{ω} be the time needed to multiply two $n \times n$ matrices. See [18] and references therein for ways of computing other linear algebra primitives such as QR decomposition, SVD, and a matrix inverse, in $O(n^{\omega})$ time. Given an $n \times d$ matrix A, $n \ge d$, we can orthogonalize its columns in time $O(nd^{\omega-1})$ as follows: first compute the product $A^{\mathsf{T}}A$ in time $nd^{\omega-1}$, compute SVD of $A^{\mathsf{T}}A$ in time $O(d^{\omega})$ to obtain V, Σ such that $A^{\mathsf{T}}A = V\Sigma^2V^{\mathsf{T}}$, and then compute $AV\Sigma^{-1}$ in time $O(nd^{\omega-1})$ to obtain an orthonormal basis.

For a matrix A, let U be a matrix with orthonormal columns and $\operatorname{colspan}(A) = \operatorname{colspan}(U)$. The leverage score of the i-th row of A, ℓ_i^2, is defined as $\|U_{i*}\|_2^2$.

Lemma 3.1. (Known constructions of sketching matrices) *For a given matrix $A \in \mathbb{R}^{n\times d}$ with $k = \operatorname{rank}(A)$, these constructions give ε-embeddings with failure probability $1-c$, for given constant c. Here the sketching matrix S is an ε-embedding if with constant probability, $\|SAx\|_2 = (1\pm\varepsilon)\|Ax\|_2$ simultaneously for all $x \in \mathbb{R}^d$.*

- *There is a sketching matrix $T \in \mathbb{R}^{m_T\times n}$ with sketching dimension $m_T = O(\varepsilon^{-2}k^{1+\mu}\log k)$ such that TA can* *be computed in $O(\mu^{-1}\mathtt{nnz}(A)/\varepsilon)$ time (see, e.g., [11]), with $1/\mu\varepsilon$ non-zero entries per column, in this form called here an OSNAP, and in earlier forms with 1 non-zero per column called a* CountSketch *[10] matrix, or* sparse embedding*. The sparsest version $\hat{T} \in \mathbb{R}^{m_{\hat{T}} \times n}$ has $m_{\hat{T}} = O(\varepsilon^{-2}k^2)$, with $\hat{T}A$ computable in $O(\mathtt{nnz}(A))$ time; $\hat{T}$ has one nonzero entry per column. A less sparse version $\bar{T}$ of OSNAP has $m_T = O(\varepsilon^{-2}k\log(nd))$, $O(\log(nd)/\varepsilon)$ entries per column, and failure probability $1/\operatorname{poly}(nd)$.*

- *There is a sketching matrix $H \in \mathbb{R}^{m_H \times n}$ with $m_H = O(\varepsilon^{-2}k\log(nk))$ such that HA can be computed in $O(nd\log n)$ time (see e.g. [5]). This is called an SRHT (Sampled Randomized Hadamard Transform) matrix. The matrix $H = \hat{H}D$, where the rows of $\hat{H}$ are a random subset of the rows of a Hadamard matrix, and D is a diagonal matrix whose diagonal entries are ± 1.*

- *If matrix $L \in \mathbb{R}^{m_L \times n}$ is chosen using leverage score sampling (see Theorem 7.1), then there is $m_L = O(\varepsilon^{-2}k\log k)$ so that L is an ε-embedding [30, 32].*

- *If matrix $G \in \mathbb{R}^{m_G \times n}$ with $m_G = O(\varepsilon^{-2}k)$ is an appropriately scaled matrix with i.i.d normal or Sub-Gaussian random variables, then G is an ε embedding.*

These embeddings can be composed, so that for example $S = H_S T_S$ is a "two-stage" ε-embedding for A, where T_S is an OSNAP matrix, and H_S is an SRHT, so that $H_S T_S A$ can be computed in $O(\varepsilon^{-1}\mu\mathtt{nnz}(A) + \varepsilon^{-2}nk^{1+1/\mu}\log^2(k/\varepsilon))$ time, and the sketching dimension is $m_{H_S} = O(\varepsilon^{-2}k\log(k/\varepsilon))$. The space needed is $O(n + m_{H_S}d)$.

We also require the following notions of projection cost preserving sketches and affine embeddings.

Definition 3.1. (Projection Cost Preserving Sketch[13]) *Given a matrix $A \in \mathbb{R}^{n\times d}$, $\varepsilon > 0$ and an integer $k \in [d]$, a sketch $SA \in \mathbb{R}^{s\times d}$ is a projection-cost preserving sketch of A if for all rank-k projection matrices P,*

$$(1-\varepsilon)\|A(I-P)\|_{\mathsf{F}}^2 \le \|SA(I-P)\|_{\mathsf{F}}^2 \le (1+\varepsilon)\|A(I-P)\|_{\mathsf{F}}^2$$

We note that sometimes Projection Cost Preserving Sketches allow an additive scalar in the definition, see, e.g., [28]. We do not need such an additive term here.

Definition 3.2. (Affine Embeddings[10]) *Given matrices A, B, let $X^* = argmin_X\|AX - B\|_{\mathsf{F}}$ and $\tilde{B} = AX^* - B$. A matrix S is an affine embedding for (A, B) if for all matrices X,*

$$\|S(AX-B)\|_{\mathsf{F}}^2 - \|S\tilde{B}\|_{\mathsf{F}}^2 = (1\pm\varepsilon)\|AX-B\|_{\mathsf{F}}^2 - \|\tilde{B}\|_{\mathsf{F}}^2.$$

Many subspace embedding distributions for the column space of A satisfy the affine embedding property. Importantly, the number of rows in S depends only on the rank of the matrix A and has no dependence on number of columns in the matrix B. See [10] for properties required of a distribution to be an affine embedding.

Throughout the paper, we use the following fact numerous times: for any matrices A, B, and C, we have $\|A - BC\|_{\mathsf{F}}^2 \ge \|A - AC^+C\|_{\mathsf{F}}^2$. This is just the Pythagorean theorem, which says that the best approximation of A inside the rowspace of C is obtained by projecting each of the rows of A onto the rowspace of matrix C.

4 Technical Overview

The only known oblivious subspace embedding for a k dimensional subspace with $o(k\log(k))$ rows is a dense matrix of $O(k)$ rows with independent Sub-Gaussian random variables. This embedding can be applied to a matrix A in time $\Omega(\mathtt{nnz}(A)\cdot k)$. All other subspace embedding constructions that are faster to apply have at least $\Omega(k\log(k))$ rows. Obtaining a subspace embedding with few rows is important to speed up the further downstream tasks such as finding a maximal set of linearly independent rows of a matrix, computing approximate leverage scores, low rank approximation, etc.

We analyze the properties required of a k-dimensional subspace $V \subseteq \mathbb{R}^d$, $d = \tilde{O}(k)$, such that a sparse random sign matrix with $o(k\log(k))$ rows can be a subspace embedding for V. The advantage of the sparsity is that the embedding can be applied to a vector quickly. Suppose every unit vector in the subspace V has at least a constant c fraction of coordinates that have a magnitude of at least $\tilde{\Omega}(1/\sqrt{k})$. Let x be an arbitrary unit vector in the subspace V. Now consider a random matrix $\boldsymbol{G}$ where each entry is either 0 with probability $1-p$ and ± 1 with probability $p/2$ each. For $p = \Theta(1/d)$, as at least a constant c fraction of the coordinates of the vector x have a magnitude $\tilde{\Omega}(1/\sqrt{k})$, each row of the matrix $\boldsymbol{G}$ has $\Omega(1)$ probability of hitting one of the large coordinates of the vector x. Conditioned on a row $\boldsymbol{G}_{i*}$ hitting one of the large coordinates of x, we have $|\boldsymbol{G}_{i*}x| \geq \tilde{\Omega}(1/\sqrt{k})$ with probability $\geq 1/2$ by using the random signs. Thus, with at least a constant probability, for a row $\boldsymbol{G}_{i*}$, $|\boldsymbol{G}_{i*}x|^2 \geq \tilde{\Omega}(1/k)$. If the matrix $\boldsymbol{G}$ has $\Omega(k)$ rows, using the Chernoff bound, we have that with very high probability, $\|\boldsymbol{G}x\|_2^2 \geq \tilde{\Omega}(1)$, which suffices to union bound over a suitable net of unit vectors in a k-dimensional subspace. On the other hand, showing that $\|\boldsymbol{G}\|_2$ is small and that it does not increase the norm of any unit vector by a lot is much easier. For the probability p that we consider, each row and column of the matrix $\boldsymbol{G}$ only has $O(1)$ nonzero entries with high probability. As all the nonzero entries are at either ± 1, we can bound the operator norm $\|\boldsymbol{G}\|_2$ by $O(1)$. This implies that for any unit vector x, $\|\boldsymbol{G}x\|_2^2 \leq O(1)$.

The above argument shows that if a subspace has the property that every unit vector in the subspace has a *large* number of *large* coordinates, then a random sparse sign matrix is a subspace embedding with small distortion for that subspace. We call subspaces having this property *flat*. But of course, the column space of the matrix to which we want to apply the embedding may not have this property. Let $V_1 \subseteq \mathbb{R}^n$ be the column space of the given matrix A. If we can find a linear map $\mathcal{F}$ that maps vectors in the subspace V_1 to a *flat* subspace V_2 and if $\mathcal{F}$ preserves the Euclidean norms of the vectors, then we have that $\|\boldsymbol{G}\mathcal{F}x\|_2 \approx \|\mathcal{F}x\|_2 \approx \|x\|_2$ for all vectors $x \in V_1$. As we show later, by paying some cost in running time, we can assume that $n = O(k\log(k))$ by first applying a series of suitable OSNAP embeddings. To obtain such a mapping $\mathcal{F}$, we use the $\ell_2 \to \ell_1$ embedding F of [22]. We show that recursively applying the linear map F gives a linear map $\mathcal{F} : n \to n^{1+o(1)}$ with the property that for all unit vectors x, $\|\mathcal{F}x\|_2 \approx 1$ and $\|\mathcal{F}x\|_1 \geq \tilde{\Omega}(\sqrt{n})$. This property immediately shows that the vector $\mathcal{F}x$ must have a large number of large coordinates and therefore that the subspace range$(\mathcal{F})$ is *flat*. We only obtain that a $1/n^{o(1)}$ fraction of the coordinates are large but it is sufficient for our purposes. We also show that the sequence of OSNAP, the mapping of [22] which we call Indyk, and the sparse random sign embeddings can be applied to a matrix $A \in \mathbb{R}^{n\times k}$ in time $O(\gamma^{-1}\mathtt{nnz}(A) + k^{2+\gamma+o(1)})$ for any constant $\gamma > 0$.

$1+\varepsilon$ **Subspace Embeddings.** We use our $\exp(\mathrm{poly}(\log\log k))$ distortion subspace embedding construction to obtain $1+\varepsilon$ *non-oblivious* subspace embeddings using approximate leverage scores obtained by using a preconditioner. Let $A \in \mathbb{R}^{n\times k}$. Earlier algorithms to compute approximate leverage scores can be described as follows : (i) Compute $\boldsymbol{S}A$ where $\boldsymbol{S}$ is a subspace embedding for the column space of A, (ii) Compute an orthonormal matrix Q and matrix R^{-1} such that $\boldsymbol{S}A = QR^{-1}$, and (iii) Compute the approximate leverage scores $\tilde{\ell}_i^2 = \|A_iR\|_2^2$.

Thus, to make computing approximate leverage scores faster, we need a subspace embedding $\boldsymbol{S}$ that can be quickly applied to matrix A to make step (i) faster while also having a fewer number of rows to make the computation of the QR-decomposition in step (ii) faster. As discussed, our subspace embedding construction $\boldsymbol{S}$ has both of these desired properties. In step (iii), instead of computing $\|A_{i*}R\|_2^2$ exactly, a Gaussian matrix $\boldsymbol{G}$ with $O(\log(n))$ columns is used so that for all the rows $i \in [n]$, $\|A_{i*}R\boldsymbol{G}\|_2^2 \approx \|A_{i*}R\|_2^2$, which is a standard idea [20]. However, computing the matrix $AR\boldsymbol{G}$ takes $\Omega(\mathtt{nnz}(A)\log(n))$ time. We consider using a Gaussian matrix with only $O(1/\gamma)$ columns for an absolute constant $\gamma > 0$, which is also a standard idea in this area. Consider an arbitrary vector v and let $\boldsymbol{g}$ be a vector of i.i.d. normal random variables. Then we have the probability that $|\langle v, \boldsymbol{g}\rangle| \leq \|v\|_2/n^{\gamma}$ is at most $1/n^{\gamma}$. If $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_t$ are independent Gaussian vectors for $t = O(1/\gamma)$, then at least one of the values $|\langle v, \boldsymbol{g}_i\rangle|$ is at least $\|v\|_2/n^{\gamma}$ with probability $\geq 1 - 1/n^2$. If $\boldsymbol{G}$ is a matrix with $\boldsymbol{g}_j$ as its columns, we therefore have that $\|A_{i*}R\boldsymbol{G}\|_2^2 \geq \|A_{i*}R\|_2^2/n^{2\gamma}$ for all i. We also argue that $\|A_{i*}R\boldsymbol{G}\|_2^2 = O(\|A_{i*}R\|_2^2\log(n))$ for all $i \in [n]$. Now the matrix $AR\boldsymbol{G}$ and the approximations $\|A_{i*}R\boldsymbol{G}\|_2^2$ can be computed in time $O(\gamma^{-1}(\mathtt{nnz}(A)+k^2))$. Therefore we can obtain over-estimates to the leverage scores. Using over-estimates to the leverage score sampling probabilities, we first sample rows and then compute accurate leverage scores only for the rows that are sampled. Then we employ a rejection step, in which we reject rows randomly based on the probabilities computed using accurate leverage scores, and finally we show that we obtain a sample from the leverage score sampling distribution. As we compute accurate leverage scores only for the rows that are sampled in the first stage, we do not incur the $O(\mathtt{nnz}(A)\log(n))$ factor. We then compose our leverage score embedding with an OSNAP embedding to obtain a $1+\varepsilon$ embedding with $O(k\log(k)/\varepsilon^2)$ rows, which is faster than previous constructions.

Computing Linearly Independent Rows. We give an algorithm to compute a maximal set of linearly independent rows of a matrix $A \in \mathbb{R}^{n\times d}$ of rank k in time $O(\mathtt{nnz}(A) + k^{\omega}\,\mathrm{poly}(\log\log(n)))$. Using the rank-preserving sketches of [8], we can assume without loss of generality that $d = ck$ for a constant c. The crucial idea here is that a leverage score sample of the matrix A, with high probability, must contain a set of k linearly independent rows. Therefore, directly applying the above leverage score sampling algorithm for constant ε gives, in time $O(\gamma^{-1}\mathtt{nnz}(A) + n^{\gamma}k^{2+o(1)} + k^{\omega}\operatorname{poly}(\log\log(n)))$, for any constant γ, a set of $O(k\exp(\operatorname{poly}(\log\log k)))$ rows of the matrix A that must contain a set of k linearly independent rows. To obtain a running time that does not depend on γ, we show that instead of running leverage score sampling on the matrix A, we can apply reductions as in [8] to reduce the problem to computing linearly independent rows of a sub-matrix A' with $\mathtt{nnz}(A') \le \mathtt{nnz}(A)/\operatorname{poly}(\log(n))$ and with $n/\operatorname{poly}(\log(n))$ rows. This reduction can be performed in time $O(\mathtt{nnz}(A) + k^{\omega}\log\log(n))$. After this reduction, we perform leverage score sampling for the matrix A' as described above with constant ε and $\gamma = O(1/\log(n))$ to obtain a matrix $\boldsymbol{S}_{\text{lev}}$ that selects and scales $O(k\exp(\operatorname{poly}(\log\log k)))$ rows randomly according to the leverage score distribution such that for all x, $\|\boldsymbol{S}_{\text{lev}}A'x\|_2 = (1 \pm 1/2)\|A'x\|_2$. In particular, the guarantee implies that $\operatorname{rowspace}(\boldsymbol{S}_{\text{lev}}A') = \operatorname{rowspace}(A')$. Therefore there are k linearly independent rows among the $O(k\exp(\operatorname{poly}(\log\log k)))$ rows sampled by $\boldsymbol{S}_{\text{lev}}$. Now we can again apply the recursive row reduction procedure mentioned above to the matrix $\boldsymbol{S}_{\text{lev}}A'$, to finally obtain, in time $O(k^{2+o(1)} + k^{\omega}\operatorname{poly}(\log\log k))$, a set of $O(k)$ rows that, with high probability, contain a set of k linearly independent rows. These rows can now be identified in time $O(k^{\omega})$. Thus, we obtain that in time $O(\mathtt{nnz}(A) + k^{\omega}\operatorname{poly}(\log\log n) + k^{2+o(1)})$, we can compute a set of k linearly independent rows of a rank k matrix A. As discussed above, the subspace embedding having $k\operatorname{poly}(\log\log k)$ rows turns out to be crucial to obtain a running time that depends on $k^{\omega}\operatorname{poly}(\log\log n)$ instead of the $k^{\omega}\log(n)$ dependence of earlier algorithms.

Low Rank Approximation. Finally, we give an algorithm to compute a $(1+\varepsilon)$-approximate rank-k approximation to an arbitrary matrix A. We note that we do not need to utilize our subspace embedding construction in this algorithm, though we include it as it is also a fundamental problem in linear algebra for which we remove the last logarithmic factor. We compute a low rank approximation in two stages: (i) we first find a rank k orthonormal matrix V whose columns span a $1+\varepsilon$ approximation. (ii) we then find a right factor $\tilde{X}$ such that $V \cdot \tilde{X}$ is a $(1+\varepsilon)$ rank-k approximation. We obtain the left factor V by using projection-cost preserving sketches and subspace embeddings along with the CUR decomposition algorithm from [6], to first obtain an $O(k)$-dimensional subspace that spans an $O(1)$-approximate rank-k low rank approximation. We then perform the residual sampling algorithm of [19] to obtain a set of $O(k/\varepsilon)$ columns of the matrix A, which along with the $O(k)$ dimensional subspace we already found, span a $(1+\varepsilon)$-approximation. We then use affine embeddings to compute a left factor V that spans a $(1+\varepsilon)$-approximation.

After finding a left factor V, the matrix $V^{\mathsf{T}}A$ is the optimal right factor but it takes $\Omega(\mathtt{nnz}(A)\cdot k)$ time to compute this matrix. We then run the CUR decomposition algorithm of Boutsidis and Woodruff [6] using the matrix V we found to obtain a right factor $\tilde{X}$ such that $\|V\cdot\tilde{X} - A\|_{\mathsf{F}} \le (1+\varepsilon)\|A - [A]_k\|_{\mathsf{F}}$.

5 Flattening the vectors

In this section, we argue that there is a linear mapping $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^{n^{1+o(1)}}$ such that for any unit vector $x \in \mathbb{R}^n$, the set

$$\operatorname{Large}(\mathcal{F}x) := \{i \in [n^{1+o(1)}] \mid |(\mathcal{F}x)_i| \ge \frac{1}{\sqrt{n}\cdot\operatorname{epll}(n)}\}$$

has size $|\operatorname{Large}(\mathcal{F}x)| = \Omega(n)$.

We show that an explicit $\ell_2 \to \ell_1$ linear embedding construction of Indyk [22] can be used to obtain such a mapping $\mathcal{F}$. First we define (ε, l) extractors as follows.

Definition 5.1. ((ε, l) extractors) *A bipartite graph $G = (A, B, E)$, $A = [a]$ and $B = [b]$, with each left node having degree d is an (ε, l) extractor if it has the following property. Let $\mathcal{P}$ be any distribution over the set A such that for all $i \in [a]$, $\mathbf{Pr}_{\mathcal{P}}[i] \le 1/l$. Consider the distribution over B generated by the following process:*

1. *Sample $i \in A$ from distribution $\mathcal{P}$*

2. *Sample $t \in [d]$ uniformly at random and set $j = \Gamma_G(i)_t$. Here $\Gamma_G(i)$ is the ordered set of neighbors of i in the graph G and $\Gamma_G(i)_t$ is the t-th neighbor in the ordered set.*

Let $G(\mathcal{P})$ be the distribution of the element j sampled by the above process and let $\mathcal{I}$ be the uniform distribution over the set B. The graph G is an (ε, l) extractor if $\sum_{j\in B}|\mathbf{Pr}_{G(\mathcal{P})}[j] - 1/b| \le \varepsilon$. We stress that this property must hold for every distribution $\mathcal{P}$ with $\mathbf{Pr}_{\mathcal{P}}[i] \le 1/l$ for all i.

See [22] and references therein for explicit constructions of extractors. Indyk uses the following extractor: Fix a $\delta = \Omega(1/\sqrt{n})$ and let $L = O(1/\delta^2)$ and $s = \sqrt{n}$. Let G be an (ε, l) extractor with $A = [Ln]$, $B = [b]$ for $b = n^{1/2-\kappa}$, $\kappa > 0$, $l = (1-\delta)^2 s/L$, left degree $d = (\log a)^{O(1)} = (\log Ln)^{O(1)}$ and right degree $\Delta = O(nLd/b)$.

In the following it will be helpful to have an abbreviation.

Definition 5.2. *Let* $\mathrm{epll}(n)$ *denote the class of functions in* $\exp(\mathrm{poly}(\log\log(n)))$ *as integer* $n \to \infty$.

Theorem 5.1. (Theorem 1.1 of [22]) *For any* $\zeta, \kappa > 0$*, there is an explicit linear mapping* $F : \mathbb{R}^n \to \mathbb{R}^m$, $m = O(nLd) = n\log^{O(1)}(n)/\zeta^{O(1)}$ *and a partitioning of the coordinate set* $[m]$ *into sets* $B_1, \ldots, B_b$*, for* $b = n^{1/2-\kappa}$*, each of size at most* $\Delta = n^{1/2+\kappa}\mathrm{epll}(n)/\zeta^{O(1)}$*, such that for any* $x \in \mathbb{R}^n$, $\|x\|_2 = 1$,

$$(1 - O(\zeta))\sqrt{Ldb} \le \sum_{j=1}^{b} \|(Fx)_{B_j}\|_2 \le \sqrt{Ldb}.$$

Without loss of generality, we can assume that all the partitions B_j *have the same size* Δ *by appending* 0*-valued coordinates and so we have* $m = n \cdot \mathrm{epll}(n)/\zeta^{O(1)}$.

We now prove the following lemma which essentially shows that an application of Indyk's embedding to a unit vector shrinks the Euclidean norm by a lot, while keeping the ℓ_1 norm $\Omega(1)$.

Lemma 5.1. *Let* n *be an arbitrary integer and* $0 < \zeta, \kappa < c$ *for a small enough constant* c*. There is an explicit linear mapping* $F : \mathbb{R}^n \to \mathbb{R}^m$ *for* $m = n \cdot \mathrm{epll}(n)/\zeta^{O(1)}$ *and a partitioning of* $[m]$ *into equal sized sets* $B_1, \ldots, B_b$ *where* $b = n^{1/2-\kappa}$ *and each set* B_j *satisfies* $|B_j| = \Delta = n^{1/2+\kappa}\mathrm{epll}(n)/\zeta^{O(1)}$*, such that for any* $x \in \mathbb{R}^n$*, we have*

$$(1 - O(\zeta))\|x\|_2 \le \sum_{j=1}^{b} \|(Fx)_{B_j}\|_2 \le \|x\|_2$$

and

$$\frac{1}{b}(1 - O(\zeta))\|x\|_2^2 \le \|Fx\|_2^2 = \sum_{j=1}^{b} \|(Fx)_{B_j}\|_2^2 \le \frac{1}{b}\|x\|_2^2.$$

Proof. In the proof of the above theorem, Indyk uses the (ε, l) construction specified above with $\delta = \zeta$ and $\varepsilon = \zeta^2$. Indyk also defines $(Fx)_{B_j} := (Dx)_{\Gamma_G(j)}$ for $j \in [b]$, where D is a concatenation of certain L orthonormal matrices and $\Gamma_G(j) \subseteq A$ is the set of neighbors of $j \in B$ in the graph G. For any unit vector x, we have $\|Dx\|_2^2 = L$ and as the left degree of G is exactly equal to d, we have $\|Fx\|_2^2 = \sum_j \|(Fx)_{B_j}\|_2^2 = \sum_j \|(Dx)_{\Gamma_G(j)}\|_2^2 = d\|Dx\|_2^2 = Ld$.

Let $y = Dx$ and let S be the set of the s largest magnitude entries of y. Define $z = y_{[a]-S}$ where z is obtained by zeroing out the coordinates of the set S. Indyk [22] showed that

$$\sum_{j=1}^{b} \left| \frac{1}{\rho^2 d}\|z_{\Gamma_G(j)}\|_2^2 - 1/b \right| \le \varepsilon$$

where $\rho \ge \sqrt{L}(1-\delta)$. The inequality implies that $\sum_j \|z_{\Gamma_G(j)}\|_2^2 \ge \rho^2 d(1-\varepsilon) \ge Ld(1-\delta)^2(1-\varepsilon)$. As $\|y_{\Gamma_G(j)}\|_2 \ge \|z_{\Gamma_G(j)}\|_2$, we get $\sum_j \|y_{\Gamma_G(j)}\|_2^2 \ge Ld(1-\delta)^2(1-\varepsilon)$ and plugging in $\delta = \zeta$ and $\varepsilon = \zeta^2$, we obtain $Ld(1 - O(\zeta)) \le \sum_{j=1}^{b} \|(Fx)_{B_j}\|_2^2 \le Ld$. Hence, the matrix $F/\sqrt{Ldb}$ satisfies that for any vector x,

$$\frac{1}{b}(1 - O(\zeta))\|x\|_2^2 \le \sum_{j=1}^{b} \|(\frac{F}{\sqrt{Ldb}}x)_{B_j}\|_2^2 \le \frac{1}{b}\|x\|_2^2.$$

From the above theorem, we already have

$$(1 - O(\zeta))\|x\|_2 \le \sum_{j=1}^{b} \|(\frac{F}{\sqrt{Ldb}}x)_{B_j}\|_2 \le \|x\|_2.$$

Therefore, scaling the matrix F gives the proof. □

We apply the above lemma recursively to each of the partitions B_j for $\Theta(\log\log(n))$ levels to obtain the following theorem.

THEOREM 5.2. *Given any n, there is an explicit map $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^m$ with $m = n \cdot \operatorname{epll}(n)$ such that for all unit vectors $x \in \mathbb{R}^n$, we have*

$$\|\mathcal{F}x\|_1 \geq \frac{\sqrt{n}}{4}$$

and

$$\frac{1}{2} \leq \|\mathcal{F}x\|_2^2 \leq 1.$$

Further, given any vector x, the vector $\mathcal{F}x$ can be computed in $n^{1+o(1)}$ time.

Proof. Let $N = \Theta(\log\log(n))$ and $\zeta = \Theta(1/\log\log(n))$. Let $B_1, \ldots, B_{b_1}$ be the partitions of the coordinates of the range of F from the Lemma 5.1. We recursively apply the lemma for each of the partitions for N levels to obtain $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^m$ for $m = n \cdot \operatorname{epll}(n)$. Define $n_0 = n$ and let n_i be the number of entries in each of the i-th level partitions. Also, let $b_0 = 1$ and b_i be the number of partitions an $(i-1)$-th level partition is mapped into. From Lemma 5.1, we have

$$b_i = n_{i-1}^{1/2-\kappa}$$

and

$$n_i = n_{i-1}^{1/2+\kappa} \operatorname{epll}(n_{i-1})/\zeta^{O(1)}.$$

The following lemma lower bounds the number of partitions in the N-th level.

LEMMA 5.2. *The total number of partitions in the N-th level is given by $B = b_0 \cdot b_1 \cdots b_N$ and*

$$B \geq n/2.$$

Proof. We have $B = b_1 \cdots b_N = (n_0 \cdots n_{N-1})^{1/2-\kappa}$. As $n_i \geq n^{(1/2+\kappa)^i}$, we have that $n_0 \cdots n_{N-1} \geq n^{\sum_{i=0}^{N-1}(1/2+\kappa)^i}$. Now, $\sum_{i=0}^{N-1}(1/2+\kappa)^i = (1-(1/2+\kappa)^N)/(1/2-\kappa)$ which implies $B \geq n^{1-(1/2+\kappa)^N}$. For $N = \Theta(\log\log(n))$, $(1/2+\kappa)^N \leq 1/\operatorname{poly}(\log(n))$ and $B \geq n/2$. □

This lemma implies that the N-th level has the partitions $\mathcal{B}_1, \ldots, \mathcal{B}_B$ of $[m]$ with $B \geq n/2$ and $|\mathcal{B}_j| = \operatorname{epll}(n)$ such that for any unit vector x,

$$\frac{1}{2}\|x\|_2 \leq (1-O(\zeta))^N \|x\|_2 \leq \sum_{j=1}^{B} \|(\mathcal{F}x)_{\mathcal{B}_j}\|_2 \leq \|x\|_2$$

and

$$\frac{1}{2B}\|x\|_2^2 \leq \frac{(1-O(\zeta))^N}{B}\|x\|_2^2 \leq \sum_{j=1}^{B} \|(\mathcal{F}x)_{\mathcal{B}_j}\|_2^2 \leq \frac{1}{B}\|x\|_2^2.$$

Finally, for a unit vector x,

$$\frac{1}{2} = \frac{1}{2}\|x\|_2 \leq \sum_{j=1}^{B} \|(\mathcal{F}x)_{\mathcal{B}_j}\|_2 \leq \sum_{j=1}^{B} \|(\mathcal{F}x)_{\mathcal{B}_j}\|_1 = \|\mathcal{F}x\|_1$$

and

$$\frac{1}{2B} = \frac{1}{2B}\|x\|_2^2 \leq \|\mathcal{F}x\|_2^2 = \sum_{j=1}^{B} \|(\mathcal{F}x)_{\mathcal{B}_j}\|_2^2 \leq \frac{1}{B}\|x\|_2^2 = \frac{1}{B}.$$

By scaling the map $\mathcal{F}$ by $\sqrt{B}$, we complete the proof. □

We now have the following corollary.

Corollary 5.1. *Given any unit vector x, at least $\Theta(n)$ coordinates of the vector $\mathcal{F}x \in \mathbb{R}^m$ have an absolute value of at least $\eta = 1/(\sqrt{n} \cdot \mathrm{epll}(n))$.*

Proof. Let m' be the number of coordinates of $\mathcal{F}x$ with an absolute value of at least η. Let $T \subseteq [m]$ be the set of indices of those coordinates. Then

$$\begin{aligned}\frac{1}{4}\sqrt{n} \le \|\mathcal{F}x\|_1 &= \sum_{i \notin T} |(\mathcal{F}x)_i| + \sum_{i \in T} |(\mathcal{F}x)_i| \\ &\le \frac{m}{\sqrt{n} \cdot \mathrm{epll}(n)} + \sqrt{\sum_{i \in T} (\mathcal{F}x)_i^2} \sqrt{|T|} \\ &\le \frac{n \cdot \mathrm{epll}(n)}{\sqrt{n} \cdot \mathrm{epll}(n)} + \sqrt{m'}.\end{aligned}$$

Here we use the Cauchy-Schwarz inequality and the fact that $\|\mathcal{F}x\|_2^2 \le 1$. For appropriate η chosen based on m, the above inequality implies that

$$\sqrt{m'} \ge \sqrt{n}/8 \implies m' \ge n/64$$

which shows that an $\Omega(n)$ fraction of the coordinates of $\mathcal{F}x$ have an absolute value of at least η. □

Thus, applying Lemma 5.1 for $N = \Theta(\log\log(n))$ levels gives an n dimensional subspace of $\mathbb{R}^m$ for $m = n \cdot \mathrm{epll}(n)$ such that for every unit vector x, the vector $\mathcal{F}x$ has a *large* number of *large* coordinates.

6 Fast Subspace Embeddings

Algorithm 1: FastEmbedding

```
Input: $A \in \mathbb{R}^{n \times k}, \gamma > 0$
Output: A subspace embedding $\boldsymbol{S}A$ with $O(k \cdot \mathrm{epll}(k))$ rows
$\boldsymbol{S}_1 \leftarrow \mathsf{OSNAP}(A, \gamma)$ with $O(k^{1+\gamma+o(1)})$ rows
$\boldsymbol{S}_2 \leftarrow \mathsf{OSNAP}(\boldsymbol{S}_1 A, O(1/\log(n)))$ with $O(k \log(k))$ rows
$\mathcal{F} \leftarrow$ Indyk Embedding for $\mathbb{R}^{O(k \log(k))}$ for $\Theta(\log\log(k))$ levels with $r = k \cdot \mathrm{epll}(k)$ rows
$m \leftarrow k \cdot \mathrm{poly}(\log\log k)$, $p \leftarrow \mathrm{epll}(k)/r$
$\boldsymbol{G} \leftarrow m \times r$ random matrix where each entry is independently 0 with probability $1-p$, and $\pm 1$ with
  probability $p/2$ each
$\boldsymbol{S}A \leftarrow \kappa \cdot \boldsymbol{G} \cdot \mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 A$ where $\kappa$ is an appropriate scaling factor
return $\boldsymbol{S}A$
```

Let A be an arbitrary $n \times k$ matrix with $\mathtt{nnz}(A)$ nonzero entries. We design a random matrix $\boldsymbol{S}$ with $k \cdot \mathrm{poly}(\log\log(k))$ rows such that with probability $\ge 9/10$, for all vectors x,

$$\|x\|_2 \le \|\boldsymbol{S}Ax\|_2 \le \mathrm{epll}(k)\|x\|_2.$$

The matrix $\boldsymbol{S}A$ can be computed in time $\mathtt{nnz}(A) + k^{2.1+o(1)}$. The matrix $\boldsymbol{S}$ is constructed as a composition of various oblivious subspace embeddings.

We first apply OSNAP $\boldsymbol{S}_1$ with $\mu = 0.1$ to obtain an $O(k^{1.1}\log(k)) \times k$ matrix $\boldsymbol{S}_1 A$ in time $O(\mathtt{nnz}(A))$. Now, $\mathtt{nnz}(\boldsymbol{S}_1 A) = O(k^{2.1}\log(k))$. Therefore, we can apply OSNAP $\boldsymbol{S}_2$ with $\mu = 1/\log(k)$, to obtain an $O(k \log k) \times k$ matrix $\boldsymbol{S}_2\boldsymbol{S}_1 A$ in time $O(\mathtt{nnz}(\boldsymbol{S}_1 A) \cdot 1/\mu) = O(k^{2.1}\log^2(k))$. We also have with probability $\ge 98/100$ that

$$\|\boldsymbol{S}_2\boldsymbol{S}_1 Ax\|_2 \in (1 \pm 3/10)\|Ax\|_2$$

for all vectors $x \in \mathbb{R}^k$. We then use the flattening transform $\mathcal{F}$ to obtain a constant subspace embedding for the matrix $\boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot A$ which also has the property that every unit vector in the column space of the matrix $\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot A$ has a large number of large entries.

THEOREM 6.1. (INDYK EMBEDDING, THEOREM 5.2 AND COROLLARY 5.1) *Given any n, there is an explicit linear map/matrix $\mathcal{F} \in \mathbb{R}^{m\times n}$ with $m = n \cdot \text{epll}(n)$ such that for any vector $x \in \mathbb{R}^n$,*

$$\frac{1}{2}\|x\|_2 \le \|\mathcal{F}x\|_2 \le \|x\|_2$$

and for any unit vector x, at least $\Theta(n)$ coordinates of the vector $\mathcal{F}x$ have an absolute value of at least $1/(\sqrt{n} \cdot \text{epll}(n))$. Given a vector $x \in \mathbb{R}^n$, the explicit map $\mathcal{F}x$ can be computed in time $n^{1+o(1)}$.

Combining $\mathcal{F}$, $\boldsymbol{S}_2, \boldsymbol{S}_1$, we obtain that with probability $\ge 98/100$, for all vectors x,

$$\frac{1}{4}\|Ax\|_2 \le \|\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \frac{3}{2}\|Ax\|_2.$$

The matrix $\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot A$ can be computed in time $\mathtt{nnz}(A) + k^{2.1+o(1)}$. As the matrix $\boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot A$ has $O(k \cdot \log(k))$ rows, the matrix $\mathcal{F}$ has $O(k \log(k) \cdot \text{epll}(k)) = k \cdot \text{epll}(k)$ rows and we also obtain that for any unit vector x in the column space of $\mathcal{F} \cdot S_2 \cdot S_1 \cdot A$, at least $\Theta(k \log k)$ coordinates have an absolute value of at least $1/(\sqrt{k \log k}\, \text{epll}(k)) = 1/(\sqrt{k}\, \text{epll}(k))$. The following theorem shows that a sparse sign matrix is a subspace embedding for a subspace with every unit vector in the subspace having a large number of large entries.

THEOREM 6.2. *Let $A \in \mathbb{R}^{m\times k}$, with $m = k \cdot \text{epll}(k)$, be a matrix such that for all unit vectors $x \in$ colspan(A), the set*

$$\textit{Large}(x) := \left\{ i \in [m] \mid |x_i| \ge \eta = \frac{1}{\sqrt{k} \cdot \text{epll}(k)} \right\}$$

satisfies $|\textit{Large}(x)| \ge Ck$ for some constant C. There is a distribution $\mathcal{G}$ over matrices with $M = k \cdot \text{poly}(\log\log(k))$ rows such that for $\boldsymbol{G} \sim \mathcal{G}$, with probability $\ge 9/10$, for all vectors $x \in \mathbb{R}^k$,

$$\|Ax\|_2 \le \|\boldsymbol{G}Ax\|_2 \le \text{epll}(k)\|Ax\|_2.$$

With probability $\ge 9/10$, the matrix $\boldsymbol{G}A$ can be computed in time $k^2 \cdot \text{epll}(k)$.

Proof. Define the $M \times m$ random matrix $\boldsymbol{G}$ as follows:

$$\boldsymbol{G}_{ij} = \begin{cases} +1 & \text{with probability } p/2 \\ -1 & \text{with probability } p/2 \\ 0 & \text{with probability } 1-p \end{cases}$$

for some values of $M \le m$ and p to be chosen later. The random variables $\boldsymbol{G}_{ij}$ are mutually independent. Let $\boldsymbol{X}_i$ be the number of nonzero entries in the i-th row of $\boldsymbol{G}$ and let $\boldsymbol{Y}_j$ be the number of nonzero entries in the i-th column of $\boldsymbol{G}$. By the Chernoff bound, for $\delta > 1$,

$$\mathbf{Pr}[\boldsymbol{X}_i \ge (1+\delta) \cdot mp] \le \exp(-\delta mp/4) \quad \text{and} \quad \mathbf{Pr}[\boldsymbol{Y}_j \ge (1+\delta) \cdot Mp] \le \exp(-\delta Mp/4).$$

Let p be such that $p|\text{Large}(x)| \ge 10$ for all x. As $|\text{Large}(x)| \ge Ck$, there is a value of p for which $pm \le \text{epll}(k)$. By a union bound, we obtain that with probability $\ge 99/100$, for all i and j, $\boldsymbol{X}_i \le \text{epll}(k)$ and $\boldsymbol{Y}_j \le \text{epll}(k)$. Thus, with probability $\ge 99/100$

$$\max_i \sum_j |\boldsymbol{G}_{ij}| = \max_i \boldsymbol{X}_i \le \text{epll}(k) \text{ and } \max_j \sum_i |\boldsymbol{G}_{ij}| = \max_j \boldsymbol{Y}_j \le \text{epll}(k).$$

We now have that $\|\boldsymbol{G}\|_2 \le \sqrt{(\max_i \sum_j |\boldsymbol{G}_{ij}|)(\max_j \sum_i |\boldsymbol{G}_{ij}|)} \le \text{epll}(k)$, which implies that for any vector y,

$$\|\boldsymbol{G} \cdot Ay\|_2 \le \text{epll}(k)\|Ay\|_2.$$

Let the event that $\|\boldsymbol{G}\|_2 \le \text{epll}(k)$ be $\mathcal{E}$.

We now show a contraction lower bound. Let x be an arbitrary unit vector in the column space of the matrix A. We say a row $\boldsymbol{G}_{i*}$ is *good* if $\boldsymbol{G}_{ij}$ is nonzero for some $j \in \text{Large}(x)$. We say $\boldsymbol{G}_{i*}$ is *bad* if it is not *good*. We have

$$\mathbf{Pr}[\boldsymbol{G}_{i*} \text{ is } \textit{bad}] = (1-p)^{|\text{Large}(x)|} \le \exp(-p|\text{Large}(x)|) \le \exp(-10) \le 1/100.$$

Thus, $\mathbf{Pr}[\boldsymbol{G}_{i*} \text{ is } \textit{good}] \ge 99/100$.

We say a row $\boldsymbol{G}_{i*}$ is *large* if $|G_{i*}x| \ge \eta$. Condition on the event that $\boldsymbol{G}_{i*}$ is *good*. Let $j \in \text{Large}(x) \cap \texttt{nnz}(\boldsymbol{G}_{i*}) \ne \emptyset$. Now, $\boldsymbol{G}_{i*}x = \sum_{j' \in \texttt{nnz}(\boldsymbol{G}_{i*})-j} \boldsymbol{G}_{ij'}x_{j'} + \boldsymbol{G}_{ij}x_j$. As entries of the matrix $\boldsymbol{G}$ are mutually independent, with probability $1/2$, $\boldsymbol{G}_{ij}x_j$ has the same sign as $\sum_{j' \in \texttt{nnz}(\boldsymbol{G}_{i*})-j} \boldsymbol{G}_{ij}x_j$, which implies that with probability $\ge 1/2$, $|\boldsymbol{G}_{i*}x| \ge |x_j| \ge \eta$. Thus,

$$\mathbf{Pr}[\boldsymbol{G}_{i*} \text{ is } \textit{large} \,|\, \boldsymbol{G}_{i*} \text{ is } \textit{good}] \ge 1/2$$

which implies that

$$\mathbf{Pr}[|\boldsymbol{G}_{i*}x| \ge \eta] = \mathbf{Pr}[\boldsymbol{G}_{i*} \text{ is } \textit{large}] \ge (1/2) \cdot (99/100) \ge 1/4.$$

Let l denote the number of *large* rows. As rows of the matrix $\boldsymbol{G}_{i*}$ are independent, *largeness* of rows is mutually independent. Thus, by the Chernoff bound,

$$\mathbf{Pr}[l \le (1/2) \cdot M \cdot (1/4)] \le \exp(-M/32).$$

We now condition on the event $\mathcal{E}$. We have

$$\mathbf{Pr}[l \le M/8 \,|\, \mathcal{E}] \le \frac{\mathbf{Pr}[l \le M/8]}{\mathbf{Pr}[\mathcal{E}]} \le 2\exp(-M/32).$$

Therefore, conditioned on the event $\mathcal{E}$, with probability $\ge 1 - 2\exp(-M/32)$, we have $l \ge M/8$ which implies that

$$\|\boldsymbol{G}x\|_2^2 \ge \sum_{\textit{large } i} |\boldsymbol{G}_{i*}x|^2 \ge l\eta^2 \ge \frac{l}{k\,\text{epll}(k)} \ge \frac{M}{8k\,\text{epll}(k)}.$$

In what follows, we condition on the event $\mathcal{E}$. For $M = k \cdot \text{poly}(\log\log(k))$, we obtain that for a unit vector x, with probability $\ge 1 - \exp(-k\,\text{poly}(\log\log(k)))$,

$$\|\boldsymbol{G}x\|_2^2 \ge \frac{\text{poly}(\log\log(k))}{\text{epll}(k)}.$$

By suitably scaling $\boldsymbol{G}$, we obtain that for all vectors x,

$$\|\boldsymbol{G}x\|_2 \le \text{epll}(k)\|x\|_2$$

and for any unit vector x, with probability $\ge 1 - \exp(-k \cdot \text{poly}(\log\log(k)))$,

$$\|\boldsymbol{G}x\|_2 \ge 2.$$

The column space of the matrix A has dimension at most k. Let $\mathcal{N}$ be a net of the unit vectors in the column space of A such that for any $y \in \text{colspace}(A)$, $\|y\|_2 = 1$, there is an $x_y \in \mathcal{N}$, $\|x_y\|_2 = 1$ such that

$$\|x_y - y\|_2 \le \frac{1}{\|\boldsymbol{G}\|_2}.$$

As $\|\boldsymbol{G}\|_2 \le \text{epll}(k)$, there exists a net $\mathcal{N}$ of size $\exp(k \cdot \text{poly}(\log\log(k)))$. We union bound over all the net vectors to obtain that with probability $\ge 99/100$, for all net vectors $x \in \mathcal{N}$,

$$\|\boldsymbol{G}x\|_2 \ge 2.$$

Now conditioning on this event, for an arbitrary $y \in \text{colspan}(A)$, $\|y\|_2 = 1$, we have

$$\begin{aligned}\|\boldsymbol{G}y\|_2 &= \|\boldsymbol{G}(x_y + (y - x_y))\|_2 \\ &\ge \|\boldsymbol{G}x_y\|_2 - \|\boldsymbol{G}(y - x_y)\|_2 \\ &\ge 2 - \|\boldsymbol{G}\|_2\|y - x_y\|_2 \\ &\ge 1\end{aligned}$$

as the net is chosen so that $\|y - x_y\|_2 \cdot \|\boldsymbol{G}\|_2 \le 1$.

Conditioned on the event $\mathcal{E}$, we have that each row of $\boldsymbol{G}$ has at most $\mathrm{epll}(k)$ nonzero entries. Thus, each row of the matrix $\boldsymbol{G}A$ can be computed in $k \cdot \mathrm{epll}(k)$ time and hence the matrix $\boldsymbol{G}A$ can be computed in time $k^2\mathrm{epll}(k)$. As $\mathbf{Pr}[\mathcal{E}] \ge 99/100$, the claim follows. □

Theorem 6.3. (Subspace Embedding) *Given an $n \times k$ matrix A, we can compute an $m \times k$ matrix $\boldsymbol{S}A$ with $m = k \cdot \mathrm{poly}(\log\log(k))$ such that with probability $\ge 9/10$, for all vectors $x \in \mathbb{R}^k$,*

$$\|Ax\|_2 \le \|\boldsymbol{S}Ax\|_2 \le \mathrm{epll}(k)\|Ax\|_2.$$

The matrix $\boldsymbol{S}\cdot A$ can be computed in time $O(\mathtt{nnz}(A)+k^{2.1+o(1)})$ or more generally in time $O(\gamma^{-1}\mathtt{nnz}(A)+k^{2+\gamma+o(1)})$ for any constant $\gamma > 0$. Further, for any matrix M with n rows,

$$\mathbf{E}[\|\boldsymbol{S}M\|_{\mathsf{F}}^2] \le \mathrm{epll}(k)\|M\|_{\mathsf{F}}^2.$$

Proof. The matrix $\boldsymbol{S}$ is defined as follows

$$\boldsymbol{S} = 4 \cdot \boldsymbol{G} \cdot \mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1$$

where $\boldsymbol{S}_1$ is OSNAP for k dimensional subspaces with $\gamma = 0.1$, $\boldsymbol{S}_2$ is OSNAP for k dimensional subspaces with $\gamma = 1/\log(k)$, $\mathcal{F}$ is Indyk's embedding for $O(k\log(k))$ dimensional subspaces as in Theorem 6.1 and $\boldsymbol{G}$ is the sparse embedding matrix with $k \cdot \mathrm{poly}(\log\log(k))$ rows as in Theorem 6.2. We have with probability $\ge 9/10$, for any vector $x \in \mathbb{R}^k$,

$$\frac{1}{2}\|Ax\|_2 \le \|\boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \frac{3}{2}\|Ax\|_2.$$

Condition on the above event. From Theorem 6.1, we have

$$\frac{1}{4}\|Ax\|_2 \le \frac{1}{2}\|\boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \|\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \|\boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \frac{3}{2}\|Ax\|_2.$$

By Theorem 6.1, every unit vector in the span of $\mathcal{F}$ has at least Ck coordinates with an absolute value of at least $1/(\sqrt{k} \cdot \mathrm{epll}(k))$. Thus, the matrix $\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot A$ satisfies the conditions of Theorem 6.2. Therefore with probability $\ge 9/10$, we have for all vectors $x \in \mathbb{R}^k$,

$$\|\boldsymbol{G} \cdot \mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \mathrm{epll}(k)\|\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \le \mathrm{epll}(k)\|Ax\|_2$$

and

$$\|\boldsymbol{G} \cdot \mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \ge \|\mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 \cdot Ax\|_2 \ge \frac{1}{4}\|Ax\|_2.$$

Thus with probability $\ge 8/10$, for all vectors x,

$$\|Ax\|_2 \le \|\boldsymbol{S} \cdot Ax\|_2 \le \mathrm{epll}(k)\|Ax\|_2.$$

The matrix $\boldsymbol{S} \cdot A$ can be computed as $4\boldsymbol{G}(\mathcal{F}(\boldsymbol{S}_2(\boldsymbol{S}_1 A))))$ in time

$$O(\mathtt{nnz}(A) + k^{2.1}\log^2(k) + k^{2+o(1)} + k^2 \cdot \mathrm{epll}(k))$$

where the last term follows from the fact that each of the $k\,\mathrm{poly}(\log\log(k))$ rows of the matrix $\boldsymbol{G}$ has at most $\mathrm{epll}(k)$ nonzero entries.

There is nothing special about $\gamma = 0.1$. We can choose any constant $1 > \gamma > 0$ and use OSNAP with the parameter γ which gives an overall running time of $O(\gamma^{-1}\mathtt{nnz}(A) + k^{2+\gamma+o(1)})$.

We now bound $\mathbf{E}_{\boldsymbol{S}}[\|\boldsymbol{S}M\|_{\mathsf{F}}^2]$ for an arbitrary matrix M. We have

$$\begin{aligned}\mathop{\mathbf{E}}_{\boldsymbol{S}}[\|\boldsymbol{S}M\|_{\mathsf{F}}^2] &= 16 \mathop{\mathbf{E}}_{\boldsymbol{G},\boldsymbol{S}_2,\boldsymbol{S}_1}[\|\boldsymbol{G} \cdot \mathcal{F} \cdot \boldsymbol{S}_1 \cdot \boldsymbol{S}_2 M\|_{\mathsf{F}}^2] \\ &\le 16 \cdot \mathop{\mathbf{E}}_{\boldsymbol{S}_1}[\mathop{\mathbf{E}}_{\boldsymbol{S}_2}[\mathop{\mathbf{E}}_{\boldsymbol{G}}[\|\boldsymbol{G} \cdot \mathcal{F} \cdot \boldsymbol{S}_2 \cdot \boldsymbol{S}_1 M\|_{\mathsf{F}}^2 \,|\, \boldsymbol{S}_1, \boldsymbol{S}_2] \,|\, \boldsymbol{S}_1]].\end{aligned}$$

First, $\mathbf{E}_{\boldsymbol{G}}[\|\boldsymbol{G}\cdot\mathcal{F}\cdot\boldsymbol{S}_2\cdot\boldsymbol{S}_1M\|_{\mathsf{F}}^2\,|\,\boldsymbol{S}_1,\boldsymbol{S}_2]\le Mp\cdot(\text{scale})\cdot\|\mathcal{F}\cdot\boldsymbol{S}_2\cdot\boldsymbol{S}_1M\|_{\mathsf{F}}^2$, where M is the number of rows of $\boldsymbol{G}$, p is the probability of an entry of $\boldsymbol{G}$ being nonzero and scale $=\text{epll}(k)$ is the scaling factor for the random sign matrix. As $M=k\cdot\text{poly}(\log\log(k))$ and $p=\text{epll}(k)/k$, we have $\mathbf{E}_{\boldsymbol{G}}[\|\boldsymbol{G}\cdot\mathcal{F}\cdot\boldsymbol{S}_2\cdot\boldsymbol{S}_1M\|_{\mathsf{F}}^2\,|\,\boldsymbol{S}_1,\boldsymbol{S}_2]\le\text{epll}(k)\cdot\|\mathcal{F}\cdot\boldsymbol{S}_2\cdot\boldsymbol{S}_1M\|_{\mathsf{F}}^2\le\text{epll}(k)\|\boldsymbol{S}_2\cdot\boldsymbol{S}_1M\|_{\mathsf{F}}^2$ as the matrix $\mathcal{F}$ does not increase the Euclidean norm of any vector. Thus,

$$\mathop{\mathbf{E}}_{\boldsymbol{S}}[\|\boldsymbol{S}M\|_{\mathsf{F}}^2]\le\text{epll}(k)\mathop{\mathbf{E}}_{\boldsymbol{S}_1}[\mathop{\mathbf{E}}_{\boldsymbol{S}_2}[\|\boldsymbol{S}_2\cdot\boldsymbol{S}_1M\|_{\mathsf{F}}^2\,|\,\boldsymbol{S}_1]]\le\text{epll}(k)\|M\|_{\mathsf{F}}^2,$$

where the last inequality follows from the fact that $\|\boldsymbol{S}_iM\|_{\mathsf{F}}^2$ is an unbiased estimator to $\|M\|_{\mathsf{F}}^2$ if $\boldsymbol{S}_i$ is an OSNAP. □

7 Applications

Algorithm 2: LeverageScoreSampling

```
Input: $A \in \mathbb{R}^{n\times k}$, $\varepsilon, \gamma > 0$
Output: An $\varepsilon$ subspace embedding $\boldsymbol{S}_{\text{lev}}A$
$\boldsymbol{S}A \leftarrow$ SparseEmbedding($A$)
$[Q, R^{-1}] \leftarrow$ QR-Decomposition($\boldsymbol{S}A$)                    // $QR^{-1} = \boldsymbol{S}A$
$s \leftarrow k\exp(\text{poly}(\log\log k)/\varepsilon^2$
$\boldsymbol{S}_1 \subseteq [n], f_i$ for $i \in [\boldsymbol{S}_1] \leftarrow$ SampleFromProduct($A, R, s, \gamma$)     // Lemma 7.2
For $i \in \boldsymbol{S}_1$, set $(\boldsymbol{S}_{\text{lev}})_{ii}$ to be equal to $1/\sqrt{f_i}$
return $\boldsymbol{S}_{\text{lev}}A$ after removing 0-value rows
```

7.1 Subspace Embeddings We use the fast subspace embedding construction from previous sections to compute approximate leverage scores and then sample rows using the approximate leverage scores to compute $1+\varepsilon$ subspace embeddings in time $O(\gamma^{-1}\mathtt{nnz}(A)+\varepsilon^{-3}n^{\gamma}k^{2+o(1)}+k^{\omega}\,\text{poly}(\log\log(k)))$ for any constant γ. We then compose with an OSNAP to obtain a subspace embedding with $O(\varepsilon^{-2}k\log(k))$ rows.

Theorem 7.1. (Leverage Score Sampling) *Given a full column rank matrix $A\in\mathbb{R}^{n\times k}$, let ℓ_i^2 for $i\in[n]$ be the leverage score of the i-th row. Let $p\in[0,1]^n$ be a vector of probabilities such that for all $i\in[n]$, $\min(1,r\cdot(\ell_i^2/k))\ge p_i\ge\min(1,r\cdot\beta\cdot(\ell_i^2/k))$ for some $\beta<1$, and let the $n\times n$ diagonal random matrix $\boldsymbol{S}_{\text{lev}}$ be defined as follows: for each $i\in[n]$, the entry $(\boldsymbol{S}_{\text{lev}})_{ii}$ is set to be equal to $1/\sqrt{p_i}$ with probability p_i, and is set to be 0 with probability $1-p_i$. If $r\ge Ck\log(k)/\beta\varepsilon^2$ for an absolute constant C, then with probability $\ge 99/100$, for all vectors $x\in\mathbb{R}^d$*

$$\|\boldsymbol{S}_{\text{lev}}Ax\|_2^2\in(1\pm\varepsilon)\|Ax\|_2^2.$$

With probability $\ge 1-\exp(-\Theta(k))$, the matrix $\boldsymbol{S}_{\text{lev}}$ has at most $\Theta(Ck\log(k)/\beta\varepsilon^2)$ nonzero entries.

The following lemma shows that a subspace embedding S for the column space of a matrix A can be used to compute approximate leverage scores which can be used to perform leverage score sampling as described above to obtain a $1+\varepsilon$ subspace embedding.

Lemma 7.1. *If S is a β subspace embedding for the column space of a full rank matrix $A\in\mathbb{R}^{n\times k}$ i.e., for any vector x,*

$$\|Ax\|_2\le\|SAx\|_2\le\beta\|Ax\|_2$$

and if $SA=QR^{-1}$ for an orthonormal matrix Q, then for all $i\in[n]$,

$$\ell_i^2/\beta^2\le\|A_{i*}R\|_2^2\le\ell_i^2,$$

where ℓ_i is the leverage score of the i-th row of A.

The proof of the lemma is in Appendix A.1. Using our fast subspace embedding with $k\,\text{poly}(\log\log(k))$ rows and $\beta=\text{epll}(k)$, the above lemma shows that if we can compute the values $\|A_{i*}R\|_2^2$, then we can obtain a $1+\varepsilon$ subspace embedding with $k\cdot\text{epll}(k)/\varepsilon^2$ rows.

Often, the row norms $\|A_{i*}R\|_2^2$ are approximated with $\|A_{i*}R\boldsymbol{G}\|_2^2$, where $\boldsymbol{G}$ is a Gaussian matrix with $O(\log n)$ columns using the fact that for an arbitrary vector x, $\|x^{\mathsf{T}}\boldsymbol{G}\|_2^2 \in (1/2, 2)\|x\|_2^2$ with probability $1 - 1/\operatorname{poly}(n)$. However, computing the matrix $AR\boldsymbol{G}$ takes $O((\mathtt{nnz}(A) + k^2)\log(n))$ time.

The following simple lemma shows that instead of obtaining constant approximations to $\|A_{i*}R\|_2^2$ for all the rows by using a Gaussian matrix $\boldsymbol{G}$ with $O(\log(n))$ columns, we can use a Gaussian matrix $\boldsymbol{G}'$ with only $O(1/\gamma)$ columns to obtain $O(n^{\gamma}\log(n))$ factor approximations to $\|A_{i*}R\|_2^2$. We sample the rows using these coarse approximations and then compute constant-factor approximations to $\|A_{i*}R\|_2^2$ only for the rows that are sampled in the first stage and then reject each of the sampled rows with appropriate probabilities to obtain a leverage score sample.

LEMMA 7.2. *Let $A \in \mathbb{R}^{n\times d}$ and $R \in \mathbb{R}^{d\times d}$ be such that for any vector $x \in \mathbb{R}^d$, the matrix-vector products ARx, Rx can be computed in time at most T_1 and T_2 respectively. Given parameters γ and s, there is an algorithm conditioned on an event $\mathcal{E}$, $\mathbf{Pr}[\mathcal{E}] \ge 95/100$, that samples indices $i \in [n]$ to obtain a random subset $\boldsymbol{S} \subseteq [n]$, such that each $i \in [n]$ is in the set $\boldsymbol{S}$ independently with probability f_i, where*

$$\min(1, s\frac{\|A_{i*}R\|_2^2}{\|AR\|_{\mathsf{F}}^2}) \ge f_i \ge \min(1, (s/16)\frac{\|A_{i*}R\|_2^2}{\|AR\|_{\mathsf{F}}^2}).$$

The algorithm returns the random subset $\boldsymbol{S}$ along with the probabilities f_i for $i \in \boldsymbol{S}$. The algorithm runs in time $O(\gamma^{-1}T_1 + T_2\log(n) + sdn^{\gamma}\log^2(n))$.

Proof. Let $p_i := \|A_{i*}R\|_2^2/\|AR\|_{\mathsf{F}}^2$ for $i \in [n]$. Let $\boldsymbol{G}_1$ be a Gaussian matrix with $O(1)$ rows and n columns and $\boldsymbol{G}_2$ be a Gaussian matrix with d rows and $O(1)$ columns. We have

$$\frac{1}{2}\|AR\|_{\mathsf{F}}^2 \le \|\boldsymbol{G}_1AR\boldsymbol{G}_2\|_{\mathsf{F}}^2 \le 2\|AR\|_{\mathsf{F}}^2 \quad \text{(Event } \mathcal{E}_1\text{)}$$

with probability $\ge 99/100$. The matrix $\boldsymbol{G}_1AR\boldsymbol{G}_2$ can be computed in $O(T_1 + n)$ time. Let $\boldsymbol{G}_3$ be a Gaussian matrix with $O(\log(n))$ columns. With probability $\ge 99/100$,

$$\text{for all } i \in [n], \quad \frac{1}{2}\|A_{i*}R\|_2^2 \le \|A_{i*}R\boldsymbol{G}_3\|_2^2 \le 2\|A_{i*}R\|_2^2 \quad \text{(Event } \mathcal{E}_2\text{)}.$$

We note that we *do not* compute the matrix $AR\boldsymbol{G}_3$ but we only compute the matrix $R\boldsymbol{G}_3$ which can be done in time $O(T_2\log(n))$.

Now, let $\boldsymbol{G}_4$ be a Gaussian matrix with $t = O(1/\gamma)$ columns. Let $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_t$ be the columns of the matrix $\boldsymbol{G}_4$. For each $i \in [n]$, with probability $\ge 1 - 1/100n^2$, $\max_{j\in[t]} |\langle A_{i*}R, \boldsymbol{g}_j\rangle| \ge \|A_{i*}R\|_2/n^{\gamma/2}$ using the fact that $|\langle A_{i*}R, \boldsymbol{g}_j\rangle|_{j\in[t]}$ are independent half-Gaussians with standard deviation $\|A_{i*}R\|_2$. By a union bound, with probability $\ge 1 - 1/100n$, for all $i \in [n]$, we have $\|A_{i*}R\boldsymbol{G}_4\|_2^2 \ge \max_{j\in[t]}\langle A_{i*}R, \boldsymbol{g}_j\rangle^2 \ge \|A_{i*}R\|_2^2/n^{\gamma}$. By Lemma 1 of [25], we also obtain that with probability $\ge 1 - 1/100n$, for all $i \in [n]$, $\|A_{i*}R\boldsymbol{G}_4\|_2^2 \le O(\log(n))\|A_{i*}R\|_2^2$. Thus, with probability $\ge 1 - 2/100n$, for all $i \in [n]$:

$$\frac{\|A_{i*}R\|_2^2}{n^{\gamma}} \le \|A_{i*}R\boldsymbol{G}_4\|_2^2 \le C\log(n)\|A_{i*}R\|_2^2 \quad \text{(Event } \mathcal{E}_3\text{)}.$$

We compute $AR\boldsymbol{G}_4$ and all squared row norms $\|A_{i*}R\boldsymbol{G}_4\|_2^2$ in time $O(T_1\gamma^{-1})$. Condition on the event $\mathcal{E} := \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$. We have $\mathbf{Pr}[\mathcal{E}] \ge 95/100$.

Define $z_i := 2n^{\gamma}\|A_{i*}R\boldsymbol{G}_4\|_2^2/\|\boldsymbol{G}_1AR\boldsymbol{G}_2\|_{\mathsf{F}}^2$. We have $4Cn^{\gamma}\log(n)p_i \ge z_i \ge p_i$ and define $q_i := \min(1, sz_i)$. Sample $i \in [n]$ independently, each with probability q_i to obtain a random subset $\boldsymbol{S}_1 \subseteq [n]$. If $i \in \boldsymbol{S}_1$, compute the value $\|A_{i*}(R\boldsymbol{G}_3)\|_2^2$ in time $O(d\log(n))$ and reject i with probability $1 - \min(1, (s/4)\|A_{i*}R\boldsymbol{G}_3\|_2^2/\|\boldsymbol{G}_1AR\boldsymbol{G}_2\|_{\mathsf{F}}^2)/q_i$.

We need to show that this procedure is well-defined. We have $(s/4)\|A_{i*}R\boldsymbol{G}_3\|_2^2/\|\boldsymbol{G}_1AR\boldsymbol{G}_2\|_2^2 \le (s/4)(4p_i) = sp_i \le sz_i$ which implies that $\min(1, (s/4)\|A_{i*}R\boldsymbol{G}_3\|_2^2/\|\boldsymbol{G}_1AR\boldsymbol{G}_2\|_{\mathsf{F}}^2) \le q_i$ and therefore the rejection probability as defined is valid. Let $\boldsymbol{S}_2$ be the subset obtained after performing the rejection step on $\boldsymbol{S}_1$. The probability that a row $i \in \boldsymbol{S}_2$ is

$$f_i = q_i \cdot \frac{\min(1, (s/4)\|A_{i*}R\boldsymbol{G}_3\|_2^2/\|\boldsymbol{G}_1AR\boldsymbol{G}_2\|_{\mathsf{F}}^2)}{q_i} \ge \min(1, (s/4)(p_i/4)) = \min(1, (s/16)p_i).$$

We also have that $f_i \le \min(1, sp_i)$. Thus with probability $\exp(-s)$ only $O(s)$ rows survive the rejection.

Now, with probability $\ge 1-\exp(-s)$, $|\boldsymbol{S}_1| = O(\sum_i q_i) = O(sn^\gamma \log(n))$ and therefore the squared row norm $\|A_i \boldsymbol{R}\boldsymbol{G}_3\|_2^2$ has to be computed only for $O(sn^\gamma \log(n))$ rows. Therefore the time complexity of sampling is $O(\gamma^{-1}T_1 + T_2\log(n) + O(sdn^\gamma \log^2(n)))$. Thus, conditioned on the event $\mathcal{E}$, the algorithm returns a subset $\boldsymbol{S} \subseteq [n]$ sampled from the desired probability distribution in time $O(\gamma^{-1}T_1 + T_2\log(n) + sdn^\gamma \log^2(n))$. □

Using these lemmas, the following theorem shows that Algorithm 2 gives a $1+\varepsilon$ subspace embedding by sampling using approximate leverage scores.

THEOREM 7.2. *Given a full rank matrix $A \in \mathbb{R}^{n\times k}$, a constant γ and a parameter $\varepsilon > 0$, we have the following:*

1. *Algorithm 2 computes a matrix $\boldsymbol{S}_{\text{lev}}A$ with $\Theta(\varepsilon^{-2}k \cdot \text{epll}(k))$ rows such that with probability $\ge 9/10$, for all vectors x,*
$$\|\boldsymbol{S}_{\text{lev}}Ax\|_2^2 \in (1\pm\varepsilon)\|Ax\|_2^2.$$
This matrix $\boldsymbol{S}_{\text{lev}}A$ can be computed in time
$$O(\gamma^{-1}\texttt{nnz}(A) + \varepsilon^{-2}n^\gamma k^{2+o(1)} + k^\omega \operatorname{poly}(\log\log(k))).$$
2. *Composing $\boldsymbol{S}_{\text{lev}}$ with the matrix $\boldsymbol{S}_{\mathsf{OSNAP}}$, an OSNAP with $O(\varepsilon^{-2}k\log(k))$ and at most $O(\varepsilon^{-1}\log(k))$ nonzero entries in each column, we obtain that with probability $\ge 9/10$, for all vectors x,*
$$\|\boldsymbol{S}_{\mathsf{OSNAP}} \cdot \boldsymbol{S}_{\text{lev}} \cdot Ax\|_2^2 \in (1\pm O(\varepsilon))\|Ax\|_2^2.$$
The matrix $\boldsymbol{S}_{\mathsf{OSNAP}} \cdot (\boldsymbol{S}_{\text{lev}}A)$ can be computed in time $O(\varepsilon^{-3}k^{2+o(1)})$ and hence, overall, the matrix $\boldsymbol{S}_{\mathsf{OSNAP}} \cdot \boldsymbol{S}_{\text{lev}} \cdot A$ can be computed in time
$$O(\gamma^{-1}\texttt{nnz}(A) + k^\omega \operatorname{poly}(\log\log(k)) + \varepsilon^{-3}k^{2+o(1)} + \varepsilon^{-2}n^{\gamma+o(1)}k^{2+o(1)})$$
for any constant γ.

Proof. From Theorem 6.2, we have a subspace embedding $\boldsymbol{S}_{\text{fast}}$ with $O(k\operatorname{poly}(\log\log k))$ rows and distortion $\text{epll}(k)$ that can be applied to matrix A in time $O(\gamma^{-1}\texttt{nnz}(A) + k^{2+\gamma+o(1)})$ for any constant $\gamma > 0$. Compute the matrices Q, R^{-1} such that Q has orthonormal columns and $\boldsymbol{S}_{\text{fast}}A = QR^{-1}$ which can be done in time $O(k^\omega \operatorname{poly}(\log\log(k)))$. By Lemma 7.1, we have

$$\frac{\ell_i^2}{\text{epll}(k)} \le \|A_{i*}R\|_2^2 \le \ell_i^2$$

which implies, using the fact $\sum_i \ell_i^2 = k$, that

$$\frac{\ell_i^2}{k\cdot\text{epll}(k)} \le \frac{\|A_{i*}R\|_2^2}{\|AR\|_{\mathsf{F}}^2}.$$

Using Lemma 7.2, conditioned on the event $\mathcal{E}$, we can sample a random subset $\boldsymbol{S}$ along with probabilities f_i for $i \in \boldsymbol{S}$ such that each $i\in[n]$ is independently in the subset $\boldsymbol{S}$ with probability f_i,

$$f_i \ge \min(1, (s/4)\cdot\frac{\|A_{i*}R\|_2^2}{\|AR\|_{\mathsf{F}}^2}) \ge \min(1, (s/4)\cdot\frac{\ell_i^2}{k\cdot\text{epll}(k)}).$$

For $s = \Theta(k\log(k)\exp(\operatorname{poly}(\log\log k))/\varepsilon^2)$, we have $f_i \ge \min(1, C\ell_i^2\log(k)/\varepsilon^2)$ which implies that the matrix $\boldsymbol{S}_{\text{lev}}$ constructed by Algorithm 2 is a $1+\varepsilon$ subspace embedding, with probability $\ge 9/10$, for the column space of A by Theorem 7.1. In the notation of Lemma 7.2, for the matrices A and R, $T_1 = \texttt{nnz}(A) + k^2$ and $T_2 = k^2$. Thus, the sampling process runs in time

$$O(\gamma^{-1}\texttt{nnz}(A) + k^2\log(n) + \varepsilon^{-2}n^\gamma k^2 \exp(\operatorname{poly}(\log\log k))) = O(\gamma^{-1}\texttt{nnz}(A) + \varepsilon^{-2}n^{\gamma+o(1)}k^{2+o(1)}).$$

Thus, overall, in time $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-2}n^{\gamma+o(1)}k^{2+o(1)} + k^{\omega}\operatorname{poly}(\log\log k))$, we can compute a leverage score sampling matrix $\boldsymbol{S}_{\text{lev}}$ with $O(\varepsilon^{-2}k\exp(\log\log k))$ rows such that for all $x \in \mathbb{R}^k$,

$$\|\boldsymbol{S}_{\text{lev}}Ax\|_2^2 \in (1 \pm \varepsilon)\|Ax\|_2^2.$$

As $\mathtt{nnz}(\boldsymbol{S}_{\text{lev}}A) \le (\varepsilon^{-1}k)^2\text{epll}(k)$, the OSNAP embedding $\boldsymbol{S}_{\mathsf{OSNAP}}$ can be applied to $\boldsymbol{S}_{\text{lev}}A$ in $O(\varepsilon^{-3}k^2\text{epll}(k))$ time and the fact that $\boldsymbol{S}_{\mathsf{OSNAP}}\cdot\boldsymbol{S}_{\text{lev}}$ is a subspace embedding follows from the composability. Thus, we can compute $\boldsymbol{S}_{\mathsf{OSNAP}}\cdot\boldsymbol{S}_{\text{lev}}\cdot A$ which has $O(\varepsilon^{-2}k\log k)$ rows in $O(\gamma^{-1}\mathtt{nnz}(A)+k^{\omega}\operatorname{poly}(\log\log k)+\varepsilon^{-3}k^{2+o(1)}+\varepsilon^{-2}n^{\gamma+o(1)}k^{2+o(1)})$ time.

□

7.2 Linear Regression Let $A \in \mathbb{R}^{n\times k}$ and $b \in \mathbb{R}^n$. By the linear regression problem (A, b), we mean $\min_x \|Ax - b\|_2$ and $\text{OPT}(A, b)$ denotes the optimum value of this problem. We prove the following theorem.

THEOREM 7.3. *Given a full-rank matrix $A \in \mathbb{R}^{n\times k}$ and $b \in \mathbb{R}^n$, we obtain a solution x^* such that*

$$\|Ax^* - b\|_2 \le (1+\varepsilon)\text{OPT}(A, b)$$

in time $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-3}n^{\gamma+o(1)}k^{2+o(1)} + k^{\omega}\operatorname{poly}(\log\log k))$ for any constant γ.

Proof. We first find a $1+\varepsilon$ subspace embedding $\boldsymbol{S}$ for the column space of $[A, b]$. From Theorem 7.2, $\boldsymbol{S}A$ and $\boldsymbol{S}b$ can be computed in at most $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-3}n^{\gamma+o(1)}k^{2+o(1)} + k^{\omega}\operatorname{poly}(\log\log k))$ time. We can also compute a preconditioner R using the fast subspace embedding from Theorem 6.2 such that

$$\kappa(AR) = \text{epll}(k)$$

by first computing $\boldsymbol{S}_{\text{fast}}A = QR^{-1}$ and then inverting R^{-1} to obtain R. The matrix R can be computed in time $O(\gamma^{-1}\mathtt{nnz}(A) + k^{2+\gamma+o(1)} + k^{\omega}\operatorname{poly}(\log\log(k)))$ for any constant γ. We also have that

$$\kappa(\boldsymbol{S}AR) = \text{epll}(k).$$

Let x^* be a solution such that $\|\boldsymbol{S}ARx^* - \boldsymbol{S}b\|_2 \le (1+\varepsilon)\min_x\|\boldsymbol{S}ARx - \boldsymbol{S}b\|_2$. Then, we have

$$\|ARx^* - b\|_2 \le \frac{1}{1-\varepsilon}\|\boldsymbol{S}ARx^* - \boldsymbol{S}b\|_2 \le \frac{1+\varepsilon}{1-\varepsilon}\|\boldsymbol{S}Ax_{\text{opt}} - \boldsymbol{S}b\|_2 \le \frac{(1+\varepsilon)^2}{1-\varepsilon}\|Ax_{\text{opt}} - b\|_2.$$

Thus, Rx^* is a $1+O(\varepsilon)$ approximate solution for the linear regression problem (A, b). Now, we focus on obtaining a $1+\varepsilon$ approximate solution for the regression problem $(\boldsymbol{S}AR, \boldsymbol{S}b)$.

We first compute an approximate solution for the regression problem as follows: let $\boldsymbol{S}_{\text{fast}}$ be the subspace embedding with $k\operatorname{poly}(\log\log(k))$ rows for the column space of $[A, b]$. Let $x^{(0)} = (\boldsymbol{S}_{\text{fast}}A)^+(\boldsymbol{S}_{\text{fast}}b)$. This solution can be computed in time $O(\mathtt{nnz}(A) + k^{2+\gamma+o(1)} + k^{\omega}\operatorname{poly}(\log\log(k)))$. Let $x_{\text{start}} = R^{-1}x^{(0)}$ which can also be computed in time $O(k^2)$. Now, we have

$$\|\boldsymbol{S}ARx_{\text{start}} - \boldsymbol{S}b\|_2 \le (1+\varepsilon)\|ARx_{\text{start}} - b\|_2 = (1+\varepsilon)\|Ax^{(0)} - b\|_2 \le (1+\varepsilon)\|\boldsymbol{S}_{\text{fast}}Ax^{(0)} - \boldsymbol{S}_{\text{fast}}b\|_2.$$

Let $x_{\boldsymbol{S}}$ be the optimal solution for the regression problem $(\boldsymbol{S}A, \boldsymbol{S}b)$. By optimality of $x^{(0)}$ for the regression problem $(\boldsymbol{S}_{\text{fast}}A, \boldsymbol{S}_{\text{fast}}b)$, we have

$$\begin{aligned}\|\boldsymbol{S}ARx_{\text{start}} - \boldsymbol{S}b\|_2 &\le (1+\varepsilon)\|\boldsymbol{S}_{\text{fast}}Ax^{(0)} - \boldsymbol{S}_{\text{fast}}b\|_2\\ &\le (1+\varepsilon)\|\boldsymbol{S}_{\text{fast}}Ax_{\boldsymbol{S}} - \boldsymbol{S}_{\text{fast}}b\|_2\\ &\le (1+\varepsilon)\cdot\text{epll}(k)\cdot\|Ax_{\boldsymbol{S}} - b\|_2\\ &\le \text{epll}(k)\cdot\text{OPT}((\boldsymbol{S}A, \boldsymbol{S}b)).\end{aligned}$$

Thus, x_{start} is an $\text{epll}(k)$ approximate solution for the linear regression problem $(\boldsymbol{S}AR, \boldsymbol{S}b)$. Using the solution x_{start}, we can obtain a $1+\varepsilon$ approximate solution in $O(\text{epll}(k)/\varepsilon)$ iterations of gradient descent where each iteration can be performed in time $O(k^2\log(k)/\varepsilon^2)$. Thus, overall, in time

$$O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-3}n^{\gamma+o(1)}k^{2+o(1)} + k^{\omega}\operatorname{poly}(\log\log k)),$$

we can compute a $1+O(\varepsilon)$ approximate solution for the linear regression problem (A, b). □

7.3 Rank Computation and Independent Row Selection We give an algorithm to compute a maximal set of independent rows of an $n \times n$ matrix A of rank $k = n^{\Omega(1)}$ in time $O(\gamma^{-1}\texttt{nnz}(A) + k^{2+\gamma+o(1)} + k^{\omega}\,\mathrm{poly}(\log\log(k)))$ for any constant $\gamma > 0$, improving upon the earlier running time of $O((\texttt{nnz}(A) + k^{\omega})\log(k))$ from Cheung et al. [8] for any constant $\omega > 2$.

Definition 7.1. (Rank Preserving Sketches) *A distribution $\mathcal{S}$ over $z_S \times n$ matrices is a rank preserving sketch if there exists a constant c such that for $\boldsymbol{S} \sim \mathcal{S}$, with high probability, for a given matrix $A \in \mathbb{R}^{n \times d}$,* $\min(\mathrm{rank}(\boldsymbol{S}A), z_S/c) = \min(\mathrm{rank}(A), z_S/c)$ *i.e., multiplying A with the matrix $\boldsymbol{S}$ preserves the rank if* $\mathrm{rank}(A) \le z_S/c$.

Theorem 7.4. ([8]) *There are rank-preserving sketching distributions as above with $c = 11$ such that*

- *$\boldsymbol{S}A$ can be computed in $O(\texttt{nnz}(A))$ time*
- *$\boldsymbol{S}$ has at most 2 nonzero entries in a column*
- *$\boldsymbol{S}$ has at most $2n/z_S$ nonzero entries in a row*

They use rank preserving sketches to give an algorithm to compute the rank of an arbitrary matrix and an algorithm to compute a maximal set of linearly independent rows of the matrix.

Theorem 7.5. (Theorem 2.6 of [8]) *Let $A \in \mathbb{R}^{n \times d}$ be an arbitrary matrix with $n \ge d$. There is a randomized algorithm to compute $k = \mathrm{rank}(A)$ in time $O(\texttt{nnz}(A)\log(k) + \min(k^{\omega}, k \cdot \texttt{nnz}(A)))$ with failure probability at most $O(1/n^{1/3})$. There is also an algorithm to find k linearly independent rows of the matrix A in time $O((\texttt{nnz}(A) + k^{\omega})\log(n))$ with failure probability at most $O(\log(n)/n^{1/3})$.*

We show that the $\log(k)$ factor can be removed from the time required to compute the rank of the matrix.

Theorem 7.6. (Rank computation) *Given $A \in \mathbb{R}^{n \times d}$, let $k = \mathrm{rank}(A)$. Let ω be the matrix multiplication constant and assume $\omega > 2$. Consider two cases:*

1. *If $k \le \log(n)^{2/(\omega-2)}$, k can be computed in time $O(\texttt{nnz}(A) + \log(n)^{6/(\omega-2)}) = O(\texttt{nnz}(A))$.*
2. *If $k \ge \log(n)^{2/(\omega-2)}$, k can be computed using Algorithm 3 (*Rank*) in time $O(\texttt{nnz}(A) + \min(k^{\omega}, k \cdot \texttt{nnz}(A)))$.*

Proof. If $k \le \log(n)^{2/(\omega-2)}$, then we have rank preserving sketches S, R such that SAR can be computed in time $\texttt{nnz}(A)$, SAR is an $O(\log(n)^{2/(\omega-2)}) \times O(\log(n)^{2/(\omega-2)})$ matrix and $\mathrm{rank}(SAR) = \mathrm{rank}(A)$. Now the rank of SAR can be computed in time $O(\log(n)^{6/(\omega-2)})$. Thus, $\mathrm{rank}(A)$ can be computed in time $O(\texttt{nnz}(A) + \log(n)^{6/(\omega-2)})$.

In the case $k \ge \log(n)^{2/(\omega-2)}$, consider Algorithm 3. As $z \ge \Theta(\sqrt{n/\log(n)})$, with failure probability $\le \Theta(\sqrt{\log(n)/n})$, the sketch SAR is rank preserving. As SAR is a $z \times z$ matrix, we have $\texttt{nnz}(SAR) \le z^2 \le O(\texttt{nnz}(A)/\log(n))$. So, the rank k_1 of SAR can be computed in time $O(\texttt{nnz}(SAR)\log(k_1) + \min(k_1^{\omega}, k_1 \cdot \texttt{nnz}(SAR))$ by Theorem 7.5. As $k_1 \le k$, we have that the rank k_1 can be computed in time $O(\texttt{nnz}(A) + \min(k^{\omega}, k \cdot \texttt{nnz}(A)))$.

We now have two cases. In the case that $k_1 < (\texttt{nnz}(A)/\log(n))^{1/2}$, as we have

$$\min(\mathrm{rank}(A), (\texttt{nnz}(A)/\log(n))^{1/2}) = \min(\mathrm{rank}(S_1AR_1), (\texttt{nnz}(A)/\log(n))^{1/2}),$$

we obtain that $\mathrm{rank}(A) = \mathrm{rank}(SAR) = k_1$.

If $(\texttt{nnz}(A)/\log(n))^{1/2} \le k_1$, we have $k = \mathrm{rank}(A) \ge k_1 \ge (\texttt{nnz}(A)/\log n)^{1/2}$ which shows that $\texttt{nnz}(A)\log(n) \le k^2\log^2(n) \le k^{\omega}$ for any $\omega > 2$ and $k \ge \log(n)^{2/(\omega-2)}$. We can now compute $\mathrm{rank}(A)$ in time $O(\texttt{nnz}(A)\log(k) + \min(k^{\omega}, k \cdot \texttt{nnz}(A)))$ by Theorem 7.5. As $\texttt{nnz}(A)\log(k) = O(\min(\texttt{nnz}(A) \cdot k, k^{\omega}))$, we obtain that the running time is $O(\texttt{nnz}(A) + \min(k^{\omega}, k \cdot \texttt{nnz}(A)))$. □

Algorithm 3: RANK(A)

```
Input: $A \in \mathbb{R}^{n\times d}$, $\mathrm{rank}(A) \geq (\log(n))^{6/(\omega-2)}$
Output: $k := \mathrm{rank}(A)$
// CKL-RE, the algorithm of Theorem 2.6 of [8]
$z \leftarrow c \cdot (\mathtt{nnz}(A)/\log n)^{1/2}$                    // $c \geq 1$ is a constant
Generate rank-preserving sketches $S \in \mathbb{R}^{z\times n}$ and $R^{\mathsf{T}} \in \mathbb{R}^{z\times d}$
Compute $SAR$                                                           // using Theorem 7.4
$k_1 \leftarrow \mathrm{rank}(SAR)$                                        // using CKL-RE
if $k_1 < z/c$ then
    return $k_1$
end
$k_2 \leftarrow \mathrm{rank}(A)$                                          // using CKL-RE
return $k_2$
```

We now describe an algorithm to compute k linearly independent rows of a matrix $A \in \mathbb{R}^{n\times d}$ of rank k in time $O(\mathtt{nnz}(A) + k^{\omega}\,\mathrm{poly}(\log\log(n)))$, replacing the $\log(n)$ factor in the running time of [8] with $\mathrm{poly}(\log\log(n))$. Thus for matrices A with $k^{\omega-1} \leq \mathtt{nnz}(A) \leq k^{\omega}/\log(n)$, we can now compute the rank k and a set of k linearly independent rows in time $O(k^{\omega}\,\mathrm{poly}(\log\log(k)))$ instead of $O(k^{\omega}\log(k))$ time.

Without loss of generality, using the rank-preserving sketch, we can assume that $d = ck$ for a constant c. The following lemma describes a reduction to a sparse sub-matrix of A which also has rank equal to $\mathrm{rank}(A)$.

Algorithm 4: ROWREDUCTION(A, k)

```
Input: $A \in \mathbb{R}^{n\times ck}$, $\mathrm{rank}(A) = k$
Output: $A_Q \in \mathbb{R}^{m\times ck}, m \leq (3n/11)k, \mathtt{nnz}(A_Q) \leq \max((2/5)\mathtt{nnz}(A), \Theta(k^2)), \mathrm{rank}(A_Q) = k$
$\boldsymbol{S} \leftarrow \mathbb{R}^{ck\times n}$ be a rank-preserving sketch
Compute $\boldsymbol{S}A$
Compute $P \subseteq [ck]$, $|P| = k$ such that $(\boldsymbol{S}A)_P$ has $k$ linearly independent rows
Let $Q \leftarrow \{i \in [m] \,|\, \boldsymbol{S}_{ji} \neq 0 \text{ for some } j \in P\}$
return $A_Q$
```

Algorithm 5: INDEPENDENTROWS(A, k)

```
Input: $A \in \mathbb{R}^{n\times d}, \mathrm{rank}(A) = k$
Output: $A_Q \in \mathbb{R}^{k\times d}, \mathrm{rank}(A_Q) = k$
$\boldsymbol{S} \leftarrow \mathbb{R}^{ck\times d}$ be a rank preserving sketch
$B \leftarrow A\boldsymbol{S}^{\mathsf{T}}$
Compute $B'$ by applying ROWREDUCTION $\Theta(\log\log(n))$ times
Compute $\boldsymbol{S}_{\mathrm{lev}}$, a leverage score subspace embedding for $B'$ using Theorem 7.2 with $\gamma = 1/\log(n)$ and
  $\varepsilon = 0.1$
Compute $B''$ with $O(k)$ rows by applying ROWREDUCTION to the matrix $\boldsymbol{S}_{\mathrm{lev}}A$, $\Theta(\log\log(k))$ times
Compute $k$ linearly independent rows of $B''$ and return $A_Q$ corresponding to these $k$ rows
```

LEMMA 7.3. *Let $A \in \mathbb{R}^{n\times ck}$ be an arbitrary matrix of rank k. There is a submatrix $A_Q \in \mathbb{R}^{m\times ck}$ that can be computed in time $O(\mathtt{nnz}(A) + k^{\omega})$ such that*

- $m = |Q| \leq (3n/11)$,
- $\mathtt{nnz}(A_Q) \leq \max((2/5) \cdot \mathtt{nnz}(A), \Theta(k^2))$, *and*
- $\mathrm{rank}(A_Q) = k$.

Proof. Let $\boldsymbol{S} \in \mathbb{R}^{ck\times n}$ be a rank-preserving sketch for $c = 11$. We have $\operatorname{rank}(\boldsymbol{S}A) = \operatorname{rank}(A) = k$ with probability $\geq 1 - O(1/k)$. Consider a set L of k linearly independent rows of the matrix $\boldsymbol{S}A$ which can be determined in $O(k^\omega)$ time. Let $Q \subseteq [n]$ be the set of rows of A that contribute to the construction of the submatrix $(\boldsymbol{S}A)_L$ which implies that $k \geq \operatorname{rank}(A_Q) \geq \operatorname{rank}((\boldsymbol{S}A)_L) = k$ and hence $\operatorname{rank}(A_Q) = k$. We therefore have that the sub-matrix A_Q consists of k linearly independent rows. The reduction $A \to A_Q$ can be performed in $O(\mathtt{nnz}(A) + k^\omega)$ time. As each row of the matrix $\boldsymbol{S}$ has at most $2n/11k$ nonzero entries, we have $|Q| \leq (2n/11k)\cdot k \leq 2n/11$. We now bound $\mathtt{nnz}(A_Q)$.

Let $P \subseteq [ck]$ be an arbitrary subset of size k. We show that if $Q_P \subseteq [n]$ is the subset of rows of A that contribute to the construction of the sub-matrix $(\boldsymbol{S}A)_P$, then $\mathtt{nnz}(A_{Q_P}) \leq (2/5)\cdot \mathtt{nnz}(A)$ with high probability.

Let $\boldsymbol{X}_i$ be the random variable that indicates if A_{i*} contributes to the construction of $(\boldsymbol{S}A)_P$ i.e., if $i \in Q_P$. By inspecting the proof of Theorem 7.4, we obtain that $\mathbf{Pr}[\boldsymbol{X}_i = 0] = (1 - 1/c)^2$. Thus, for $c = 11$, we obtain that $\mathbf{Pr}[\boldsymbol{X}_i = 1] = 1 - (1 - 1/11)^2 = 21/121$. We also note that the random variables $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are negatively associated [36]. Let a_i denote the number of nonzero entries of the row A_{i*} which implies that $\sum_i a_i = \mathtt{nnz}(A)$. Now, we have $\mathtt{nnz}(A_{Q_P}) = \sum_i a_i \boldsymbol{X}_i$. Using the Chernoff-Hoeffding bound for negatively associated random variables [21],

$$\mathbf{Pr}[\mathtt{nnz}(A_{Q_P}) = \sum_i a_i \boldsymbol{X}_i \geq \mathtt{nnz}(A)\cdot 21/121 + t] \leq 2\exp\left(-\frac{2t^2}{\sum_i a_i^2}\right).$$

By a union bound over all $\binom{11k}{k} \leq (11e)^k$ subsets P, we obtain that for a constant C,

$$\mathbf{Pr}[\text{There is a subset } P \subseteq [11k], |P| = k \text{ with } \mathtt{nnz}(A_{Q_P}) \geq \mathtt{nnz}(A)/5 + t] \leq 2\exp\left(Ck - \frac{2t^2}{\sum_i a_i^2}\right).$$

Now, we have $\sum_i a_i^2 \leq \max_i a_i \cdot \sum_i a_i \leq 11k\cdot(\mathtt{nnz}(A))$ since the matrix A is assumed to have only $ck = 11k$ columns. For $t \geq \Theta(k\sqrt{\mathtt{nnz}(A)})$, we obtain that with probability $\geq 1 - \exp(-\Theta(k))$, for all $P \subseteq [11k], |P| = k$, we have that $\mathtt{nnz}(A_{Q_P}) \leq \mathtt{nnz}(A)/5 + t$. For $\mathtt{nnz}(A) \geq \Theta(k^2)$, we have $\mathtt{nnz}(A)/5 \geq \Theta(k\sqrt{\mathtt{nnz}(A)})$ which implies that for all P, $\mathtt{nnz}(A_{Q_P}) \leq (2/5)\mathtt{nnz}(A)$. This, in particular, implies that for $M = Q_L$, that corresponds to the set of rows contributing to a linearly independent set of rows of $(\boldsymbol{S}A)$, we have $\mathtt{nnz}(A_M) \leq (2/5)\cdot\mathtt{nnz}(A)$ if $\mathtt{nnz}(A) \geq \Theta(k^2)$. □

Recursively applying the above lemma, we obtain the following.

Corollary 7.1. *Let $A \in \mathbb{R}^{n\times d}$ be an arbitrary matrix of rank k. There is a matrix $A' \in \mathbb{R}^{m\times ck}$ with either $\mathtt{nnz}(A') \leq \mathtt{nnz}(A)/\log(n)$ or $\mathtt{nnz}(A') \leq \Theta(k^2)$ such that*

- $\operatorname{rank}(A') = \operatorname{rank}(A) = k$, *and*
- $m \leq n/\operatorname{poly}(\log(n))$
- *linearly independent rows of A' correspond to linearly independent rows of A.*

The reduction $A \to A'$ can be performed in $O(\mathtt{nnz}(A) + k^\omega \log\log(n))$ time.

Proof. Let $N = \Theta(\log\log(n))$ and $A^{(0)} = A$. Starting with $i = 0$, we apply the above reduction $A^{(i)} \to A^{(i+1)}$ to obtain a matrix with $\mathtt{nnz}(A^{(i+1)}) \leq (2/5)\cdot\mathtt{nnz}(A^{(i)})$. Then

$$\mathtt{nnz}(A^{(N)}) \leq \max((2/5)^N \mathtt{nnz}(A), \Theta(k^2)) \leq \max(\mathtt{nnz}(A)/\log(n), \Theta(k^2)).$$

The time complexity is $O(\sum_{i=1}^N (\mathtt{nnz}(A^{(i)}) + k^\omega)) = O(\mathtt{nnz}(A) + k^\omega \log\log(n))$. □

We have now reduced the general problem of computing k linearly independent rows of a rank-k $n \times d$ matrix A to computing k linearly independent rows of a rank-k $m \times ck$ matrix A' with $m \leq n/\operatorname{poly}(\log(n))$ and $\mathtt{nnz}(A') \leq O(\max(k^2, \mathtt{nnz}(A)/\log(n)))$. Using these reductions, we have the following theorem.

Theorem 7.7. *Given an arbitrary matrix $A \in \mathbb{R}^{n\times d}$ of rank k, Algorithm 5 computes a set of k linearly independent rows of the matrix A in time $O(\mathtt{nnz}(A) + k^\omega \operatorname{poly}(\log\log(n)) + k^{2+o(1)})$.*

Proof. Let $\boldsymbol{S} \in \mathbb{R}^{ck \times d}$ be a rank preserving sketch which implies $\operatorname{rank}(A\boldsymbol{S}^{\mathsf{T}}) = \operatorname{rank}(A) = k$ with probability $1 - O(1/k)$. Condition on this event. Let $M \subseteq [n]$, $|M| = k$ be such that rows of the sub-matrix $(A\boldsymbol{S}^{\mathsf{T}})_M = A_M\boldsymbol{S}^{\mathsf{T}}$ are linearly independent. Then, $k \ge \operatorname{rank}(A_M) \ge \operatorname{rank}(A_M\boldsymbol{S}^{\mathsf{T}}) = k$ which implies $\operatorname{rank}(A_M) = k$. Thus, we only have to find k linearly independent rows of the $n \times ck$ matrix $B = A\boldsymbol{S}^{\mathsf{T}}$. We also have $\mathtt{nnz}(B) = O(\mathtt{nnz}(A))$. Using the above corollary, we can find an $m \times ck$ sub-matrix B' such that $\operatorname{rank}(B') = k$, $\mathtt{nnz}(B') \le O(\max(\mathtt{nnz}(B)/\operatorname{poly}(\log(n)), \Theta(k^2))$ and $m = n/\operatorname{poly}(\log(n))$.

From Theorem 7.2, using $\gamma = 1/\log(n)$, in time $O(\mathtt{nnz}(B')\log(n) + k^{\omega}\operatorname{poly}(\log\log(n)) + k^{2+o(1)} + m\gamma^{-1}) = O(\mathtt{nnz}(A) + k^{\omega}\operatorname{poly}(\log\log(n)) + k^{2+o(1)})$, we can compute a row sampling matrix $\boldsymbol{S}_{\text{lev}}$ that samples $O(k \cdot \operatorname{epll}(k))$ rows such that

$$\|\boldsymbol{S}_{\text{lev}}B'x\|_2^2 \in (1 \pm 1/10)\|B'x\|_2^2$$

for all vectors x. This, implies that the matrix $\boldsymbol{S}_{\text{lev}}B'$ has rank k and hence has k linearly independent rows.

As $\boldsymbol{S}_{\text{lev}}$ is a leverage score sampling matrix, the rows of $\boldsymbol{S}_{\text{lev}}B'$ are multiples of rows of the matrix B'. Thus, a set of k linearly independent rows of the matrix $\boldsymbol{S}_{\text{lev}}B'$ directly corresponds to a set of k linearly independent rows of B which corresponds to a set of k linearly independent rows of the matrix A.

Applying the row reduction $\operatorname{poly}(\log\log(k))$ times to the matrix $\boldsymbol{S}_{\text{lev}}B'$, we obtain a matrix B'' of dimension $O(k) \times k$ from which we can determine a set of k linearly independent rows in time $O(k^{\omega})$. This concludes the proof. □

7.4 Low-Rank Approximation

Let $A \in \mathbb{R}^{n \times d}$ be an arbitrary matrix. We want to compute a matrix B of rank at most k such that

$$\|A - B\|_{\mathsf{F}}^2 \le (1 + \varepsilon)\|A - [A]_k\|_{\mathsf{F}}^2.$$

Let OPT_A denote $\|A - [A]_k\|_{\mathsf{F}}^2$. Our main theorem for Low-Rank Approximation (LRA) is as follows.

THEOREM 7.8. *Let $A \in \mathbb{R}^{n \times d}$, $k < \min(n, d)$ be a rank parameter and $\varepsilon > 0$ be an accuracy parameter. There is an algorithm that outputs matrices $V \in \mathbb{R}^{n \times k}$ and $X \in \mathbb{R}^{k \times d}$, $V^{\mathsf{T}}V = I_k$, such that with $\Omega(1)$ probability,*

$$\|A - VX\|_{\mathsf{F}}^2 \le (1 + \varepsilon)\|A - [A]_k\|_{\mathsf{F}}^2.$$

The algorithm runs in time $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-1}(n + d)k^{\omega - 1} + \varepsilon^{-1}k(nd^{\gamma + o(1)} + dn^{\gamma + o(1)}) + \operatorname{poly}(\varepsilon^{-1}k))$ for any constant $\gamma > 0$.

In the following sections, we will describe how to compute the left factor V and the right factor X. We are not very careful with probabilities, as we only have to condition over the success of $O(1)$ events, and all these events can be chosen to have a success probability $1 - c$ for any absolute constant $c > 0$ without affecting the time complexity.

We start with a residual sampling algorithm that lets us obtain a subspace containing a $1 + \varepsilon$ approximation given a subspace that is only $O(1)$ approximate.

7.4.1 Residual Sampling

Suppose we have a subspace $V \in \mathbb{R}^d$ such that

$$\|A - A\mathbb{P}_V\|_{\mathsf{F}}^2 \le K\|A - [A]_k\|_{\mathsf{F}}^2.$$

The following theorem of [19] shows that sampling $O(K \cdot k/\varepsilon)$ rows of the matrix A with probabilities proportional to the squared distances of the rows to the subspace V gives a subspace that along with V contains a $1 + \varepsilon$ rank-k approximation to the matrix A.

THEOREM 7.9. (THEOREM 2.1 OF [19]) *Let $A \in \mathbb{R}^{n \times d}$ and $V \in \mathbb{R}^d$ be a subspace. Let $E = A - A\mathbb{P}_V$, the matrix formed by projecting each row of A away from the subspace V. Let $\boldsymbol{S}$ be a random sample of s rows of A from a distribution $\mathcal{D}$ such that row i is chosen with probability $p_i \ge \alpha\|E_{i*}\|_2^2/\|E\|_{\mathsf{F}}^2$. Then for any non-negative integer k,*

$$\mathop{\mathbf{E}}_{\boldsymbol{S}}[\min_{\substack{\text{rank-}k\ B \\ rowspan(B) \subseteq V + rowspan(A_{\boldsymbol{S}})}} \|A - B\|_{\mathsf{F}}^2] \le \|A - A_k\|_{\mathsf{F}}^2 + \frac{k}{s\alpha}\|E\|_{\mathsf{F}}^2.$$

Instead of sampling s rows independently from the distribution p, we can also sample each $i \in [n]$ with probability $q_i := \min(1, sp_i)$ and obtain the same result for the resulting random subset of rows. Sampling each $i \in [n]$ independently with probability q_i lets us use the sampling framework from Lemma 7.2.

LEMMA 7.4. (SAMPLING EACH ROW INDEPENDENTLY) *Let $A \in \mathbb{R}^{n\times d}$ and V be a subspace in $\mathbb{R}^d$ and let $E = A - A\mathbb{P}_V$. Sample each $i \in [n]$ independently with a probability $q_i := \min(1, sp_i)$, with $p_i \ge \alpha\|E_{i*}\|_2^2/\|E\|_{\mathsf{F}}^2$ to obtain a random subset $\boldsymbol{S} \subseteq [n]$. For any nonnegative integer k,*

$$\mathop{\mathbf{E}}_{\boldsymbol{S}}[\min_{\substack{\text{rank-}k\ B\\ \textit{rowspan}(B)\subseteq V+\textit{rowspan}(A_{\boldsymbol{S}})}} \|A-B\|_{\mathsf{F}}^2] \le \|A-A_k\|_{\mathsf{F}}^2 + \frac{k}{s\alpha}\|E\|_{\mathsf{F}}^2.$$

The proof of this lemma is in Appendix A.2

7.4.2 Computing the left factor of an approximation Let $\boldsymbol{T}$ be a CountSketch matrix with $\Theta(k^2)$ columns. In [13], the authors show that $\boldsymbol{T}$ is a projection cost preserving sketch, i.e., with probability $9/10$, for all projection matrices P of rank at most $O(k)$,

$$\|(I-P)A\boldsymbol{T}\|_{\mathsf{F}}^2 = (1 \pm 1/10)\|(I-P)A\|_{\mathsf{F}}^2.$$

Let $\boldsymbol{S}$ be a CountSketch matrix with $\Theta(k^4)$ rows. Then, with probability $\ge 99/100$, $\boldsymbol{S}$ is a subspace embedding for the matrix $A\boldsymbol{T}$ and therefore for any matrix X,

$$\|\boldsymbol{SAT}X - \boldsymbol{SAT}\|_{\mathsf{F}}^2 = (1 \pm 1/10)\|A\boldsymbol{T}X - A\boldsymbol{T}\|_{\mathsf{F}}^2.$$

We can relate OPT_A and $\mathrm{OPT}_{\boldsymbol{SAT}}$ as follows:

$$\mathrm{OPT}_{\boldsymbol{SAT}} = \|\boldsymbol{SAT} - [\boldsymbol{SAT}]_k\|_{\mathsf{F}}^2 = \min_{\text{rank-}k\ X}\|\boldsymbol{SAT} - \boldsymbol{SAT}X\|_{\mathsf{F}}^2 \le \frac{11}{10}\min_{\text{rank-}k\ X}\|A\boldsymbol{T} - A\boldsymbol{T}X\|_{\mathsf{F}}^2 = \frac{11}{10}\mathrm{OPT}_{A\boldsymbol{T}}$$

where the inequality follows from the subspace embedding property of $\boldsymbol{S}$ for the column space of $A\boldsymbol{T}$. Now,

$$\mathrm{OPT}_{A\boldsymbol{T}} = \min_{\text{rank-}k\ \text{projections}\ P}\|(I-P)A\boldsymbol{T}\|_{\mathsf{F}}^2 \le \frac{10}{9}\min_{\text{rank-}k\ \text{projections}\ P}\|(I-P)A\|_{\mathsf{F}}^2 = \frac{10}{9}\mathrm{OPT}_A.$$

Here, the inequality follows as $\boldsymbol{T}$ is a projection cost preserving sketch for k dimensional projections. Thus, $\mathrm{OPT}_{\boldsymbol{SAT}} \le (11/9)\mathrm{OPT}_A$.

Boutsidis and Woodruff [6] show that for any matrix M, there exists a sub-matrix M' of M, with $O(k/\varepsilon)$ columns such that there is a rank k matrix B, $\mathrm{colspan}(B) \subseteq \mathrm{colspan}(M')$, and $\|M-B\|_{\mathsf{F}}^2 \le (1+\varepsilon)\|M-[M]_k\|_{\mathsf{F}}^2$. They also give an algorithm to find such a subset of columns. As $\boldsymbol{SAT}$ is a $O(k^4)\times O(k^2)$ matrix, using their algorithm, we can compute in time $\mathrm{poly}(k)$, a column selection matrix Ω that selects $O(k)$ columns of $\boldsymbol{SAT}$ such that

$$\min_{\text{rank-}k\ X}\|\boldsymbol{SAT} - \boldsymbol{SAT}\Omega X\|_{\mathsf{F}}^2 \le \frac{3}{2}\mathrm{OPT}_{\boldsymbol{SAT}} \le 2\mathrm{OPT}_A.$$

We now have $\|(\boldsymbol{SAT}\Omega)(\boldsymbol{SAT}\Omega)^+\boldsymbol{SAT} - \boldsymbol{SAT}\|_{\mathsf{F}}^2 \le \min_{\text{rank-}k\ X}\|\boldsymbol{SAT} - \boldsymbol{SAT}\Omega X\|_{\mathsf{F}}^2 \le 2\mathrm{OPT}_A$. Using the property that $\boldsymbol{S}$ is a subspace embedding for the column space of $A\boldsymbol{T}$, we have

$$\|A\boldsymbol{T}\Omega(\boldsymbol{SAT})^+\boldsymbol{SAT} - A\boldsymbol{T}\|_{\mathsf{F}}^2 \le \frac{20}{11}\mathrm{OPT}_A.$$

Let U be a matrix with orthonormal columns such that $\mathrm{colspan}(A\boldsymbol{T}\Omega) = \mathrm{colspan}(U)$. Therefore,

$$\|UU^{\mathsf{T}}A\boldsymbol{T} - A\boldsymbol{T}\|_{\mathsf{F}}^2 \le \|(A\boldsymbol{T}\Omega)(\boldsymbol{SAT}\Omega)^+\boldsymbol{SAT} - A\boldsymbol{T}\|_{\mathsf{F}}^2 \le \frac{20}{11}\mathrm{OPT}_A$$

which finally implies, as $\boldsymbol{T}$ is a projection cost preserving sketch for $O(k)$ dimensional projections, that $\|UU^{\mathsf{T}}A - A\|_{\mathsf{F}}^2 \le (10/9)(20/11)\mathrm{OPT}_A \le 3\mathrm{OPT}_A$. Thus, $\mathrm{colspan}(U)$ is an $O(k)$ dimensional subspace with $\|(I - UU^{\mathsf{T}})A\|_{\mathsf{F}}^2 \le 3\mathrm{OPT}_A$. As, $\boldsymbol{T}$ and $\boldsymbol{S}$ are CountSketch matrices, the matrices $A\boldsymbol{T}$ and $\boldsymbol{SAT}$ can be computed in time $\mathtt{nnz}(A)$. The matrix Ω can be computed in time $\mathrm{poly}(k)$ and the matrix $A\boldsymbol{T}\Omega$ is obtained by selecting the appropriate columns of matrix $A\boldsymbol{T}$. The orthonormal matrix U can be computed in time $O(nk^{\omega-1})$. Using U, we now obtain a larger subspace of dimension $O(k/\varepsilon)$ that spans a $1+\varepsilon$ approximation.

Using Lemma 7.4, we have that if *columns* of the matrix A are sampled independently to obtain a subset $\boldsymbol{S}_{\text{res}} \subseteq [d]$ such that $\mathbf{Pr}[j \in \boldsymbol{S}_{\text{res}}] \ge \min(1, sp_j)$ for $s = O(k/\varepsilon)$, $p_j = \|(I - UU^{\mathsf{T}})A_{*j}\|_2^2 / \|(I - UU^{\mathsf{T}})A\|_{\mathsf{F}}^2$, then with probability $\ge 99/100$, the subspace $\text{colspan}(U) + \text{colspan}(A^{\boldsymbol{S}_{\text{res}}})$ spans columns of a k dimensional matrix that is a $(1+\varepsilon)$ rank-k approximation for A.

Lemma 7.2 shows how to sample $\boldsymbol{S}_{\text{res}}$ from such a distribution. In the notation of Lemma 7.2, we have $T_1 = O(\mathtt{nnz}(A) + nk)$ and $T_2 = nk$. Therefore, with probability $\ge 95/100$, we can obtain a sample $\boldsymbol{S}_{\text{res}}$ from a distribution over subsets of $[d]$ such that independently, $\mathbf{Pr}[j \in \boldsymbol{S}_{\text{res}}] \ge \min(1, O(k/\varepsilon)p_j)$ in time $O(\gamma^{-1}(\mathtt{nnz}(A) + nk) + nk\log(d) + \varepsilon^{-1}d^{\gamma}nk\log^2(d)) = O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-1}nkd^{\gamma+o(1)})$ for any small constant γ. Let $M = [U\, A^{\boldsymbol{S}_{\text{res}}}]$. We have with probability $\ge 9/10$, that

$$\min_{\text{rank-}k\, X} \|MX - A\|_{\mathsf{F}}^2 \le (1+\varepsilon)\text{OPT}_A.$$

To obtain a good k-dimensional subspace within the column space of M, we can sketch and solve the above problem. Let $\boldsymbol{T}_1$ be a CountSketch matrix with $O((k/\varepsilon)^2/\varepsilon^2)$ rows. Then with probability $\ge 99/100$, $\boldsymbol{T}_1$ is an affine embedding for (M, A) and therefore for any matrix X, $\|\boldsymbol{T}_1MX - \boldsymbol{T}_1A\|_{\mathsf{F}}^2 \in (1\pm\varepsilon)\|MX - A\|_{\mathsf{F}}^2$. Let $X_{\boldsymbol{T}_1}$ be the optimal solution for $\min_{\text{rank-}k\, X} \|\boldsymbol{T}_1MX - \boldsymbol{T}_1A\|_{\mathsf{F}}$. As $X_{\boldsymbol{T}_1}$ is optimal, the rows of the matrix $X_{\boldsymbol{T}_1}$ must be spanned by the rows of the matrix $\boldsymbol{T}_1A$, which implies that $\min_{\text{rank-}k\, X} \|MX\boldsymbol{T}_1A - A\|_{\mathsf{F}}^2 \le (1 + O(\varepsilon))\text{OPT}_A$. This problem can now be solved by sketching on the left and the right with $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$, where $\boldsymbol{T}_2$ is a CountSketch matrix with $\text{poly}(k/\varepsilon)$ rows, and then solving the sketched problem optimally. The time complexity of sketching is $O(\mathtt{nnz}(M) + \mathtt{nnz}(A)) = O(\mathtt{nnz}(A) + nk/\varepsilon)$, and the sketched problem can be solved in time $\text{poly}(k/\varepsilon)$. Thus in time $O(\mathtt{nnz}(A) + nk/\varepsilon + \text{poly}(k/\varepsilon))$, we can compute a rank k matrix X such that

$$\|MX\boldsymbol{T}_1A - A\|_{\mathsf{F}}^2 \le (1 + O(\varepsilon))\text{OPT}_A.$$

We can also compute a decomposition of $X = X_1 \cdot X_2$ where X_1 has k columns in time $\text{poly}(k/\varepsilon)$, which implies that the k dimensional column span of MX_1 is a $1 + O(\varepsilon)$ approximate rank k singular subspace i.e., $\|(MX_1)(MX_1)^+A - A\|_{\mathsf{F}}^2 \le (1 + O(\varepsilon))\text{OPT}_A$. The matrix MX_1 can be computed in time $O(nk^{\omega-1}/\varepsilon)$ and a matrix V which is an orthonormal basis for the column space of the $n \times k$ matrix MX_1 can be computed in time $O(nk^{\omega-1})$. Thus, in time $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-1}nkd^{\gamma+o(1)} + \varepsilon^{-1}nk^{\omega-1} + \text{poly}(\varepsilon^{-1}k))$, we can compute a left factor for a $1+\varepsilon$ rank-k approximation of A. Thus, we have the following lemma.

LEMMA 7.5. *Given a matrix $A \in \mathbb{R}^{n\times d}$, a rank parameter k and accuracy parameter ε, we can compute a matrix V with k orthonormal columns in time $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-1}nkd^{\gamma+o(1)} + \varepsilon^{-(\omega-1)}nk^{\omega-1} + \text{poly}(\varepsilon^{-1}k))$ such that*

$$\|A - VV^{\mathsf{T}}A\|_{\mathsf{F}}^2 \le (1+\varepsilon)\|A - [A]_k\|_{\mathsf{F}}^2.$$

7.4.3 Computing a right factor given a left factor Given a matrix V with k orthonormal columns such that

$$\min_X \|VX - A\|_{\mathsf{F}}^2 \le (1 + O(\varepsilon))\|A - [A]_k\|_{\mathsf{F}}^2,$$

we want to compute a rank k matrix $\tilde{X}$ that satisfies $\|V\tilde{X} - A\|_{\mathsf{F}}^2 \le (1 + O(\varepsilon))\|A - [A]_k\|_{\mathsf{F}}^2$.

For $i \in [n]$, let $p_i = \|V_{*i}\|_2^2/k$. Suppose $\boldsymbol{S}_{\text{lev}}$ is a sampling matrix with $s = O(k\log(k))$ rows such that each row of $\boldsymbol{S}_{\text{lev}}$ is independently equal to $e_i^{\mathsf{T}}/\sqrt{sp_i}$ with a probability p_i. Then we have

$$\text{for all vectors } x, \|\boldsymbol{S}_{\text{lev}}Vx\|_2^2 \in (1 \pm 1/2)\|Vx\|_2^2.$$

Let $M_2 = V^{\mathsf{T}}\boldsymbol{S}_{\text{lev}}^{\mathsf{T}}$ and let V_{M_2} be a matrix with k orthonormal columns such that $\text{colspan}(V_{M_2}) = \text{rowspan}(M_2)$. Let S_2 be the BSS-Sampling matrix returned by the dual set spectral sparsification algorithm of [6] on the inputs $V_{M_2}, \boldsymbol{S}_{\text{lev}}(I - VV^{\mathsf{T}})A\boldsymbol{T}$ with a parameter $4k$, where $\boldsymbol{T}$ is a CountSketch matrix with $O(k^2)$ columns. The matrix S_2 selects $4k$ rows of the matrix $\boldsymbol{S}_{\text{lev}}A$. Let $R_1 = S_2\boldsymbol{S}_{\text{lev}}A$. Lemma 6.7 of [6] shows that

$$\|A - AR_1^+R_1\|_{\mathsf{F}}^2 \le O(1)\|A - [A]_k\|_{\mathsf{F}}^2.$$

As the matrix R_1 has $4k$ rows, an orthonormal basis U for the rowspace of R_1, with $4k$ orthonormal columns, can be computed in time $dk^{\omega-1}$. We can then perform residual sampling of rows of A with respect to the subspace U using the Lemma 7.2. Here $T_1 = \mathtt{nnz}(A) + dk$ and $T_2 = dk$. Thus, we can sample rows from a distribution defined by the probabilities $\min(1, (s/16)\|A_{i*}(I - UU^{\mathsf{T}})\|_2^2/\|A(I - UU^{\mathsf{T}})\|_{\mathsf{F}}^2)$, for $s = O(k/\varepsilon)$ in time $O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-1}dkn^{\gamma+o(1)})$. Let $\boldsymbol{S}'_{\mathrm{res}} \subseteq [n]$ be the rows sampled. Let $R = \begin{bmatrix} U^{\mathsf{T}} \\ A_{\boldsymbol{S}'_{\mathrm{res}}} \end{bmatrix}$. The matrix R has $O(k/\varepsilon)$ rows.

Now, as in proof of the Theorem 5.1 of [6], we have with proabability $\geq 9/10$,

$$\|A - VV^{\mathsf{T}}AR^{+}R\|_{\mathsf{F}}^2 \leq (1 + O(\varepsilon))\|A - [A]_k\|_{\mathsf{F}}^2,$$

which implies $\min_X \|A - VXR\|_{\mathsf{F}}^2 \leq (1 + O(\varepsilon))\|A - [A]_k\|_{\mathsf{F}}^2$. By sketching the problem on the left and the right with CountSketch matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ with $\mathrm{poly}(k/\varepsilon)$ rows and columns respectively, the optimal solution $X_{\boldsymbol{T}}$ for the sketched problem satisfies

$$\|A - VX_{\boldsymbol{T}}R\|_{\mathsf{F}}^2 \leq (1 + O(\varepsilon))\|A - [A]_k\|_{\mathsf{F}}^2.$$

Finally, the product $X_{\boldsymbol{T}} \cdot R$ can be computed in time $O(dk^{\omega-1}/\varepsilon)$ to obtain a matrix $\tilde{X}$ such that

$$\|A - V\tilde{X}\|_{\mathsf{F}}^2 \leq (1 + O(\varepsilon))\|A - [A]_k\|_{\mathsf{F}}^2.$$

Thus, we can compute two matrices $V, \tilde{X}$ with k columns and k rows respectively, such that the product $V \cdot \tilde{X}$ is a $1 + \varepsilon$ approximate rank-k Frobenius norm approximation to the matrix A, in time

$$O(\gamma^{-1}\mathtt{nnz}(A) + \varepsilon^{-1}(n + d)k^{\omega-1} + \varepsilon^{-1}k(nd^{\gamma+o(1)} + dn^{\gamma+o(1)}) + \mathrm{poly}(\varepsilon^{-1}k)).$$

Acknowledgments: P. Kacham and D. Woodruff research were supported in part by National Institute of Health grant 5R01 HG 10798-2, NSF award CCF-1815840, and a Simons Investigator Award.

References

[1] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. SIAM, 2021.

[2] Ainesh Bakshi, Rajesh Jayaram, and David P Woodruff. Learning two layer rectified neural networks in polynomial time. In *Conference on Learning Theory*, pages 195–268. PMLR, 2019.

[3] Frank Ban, Vijay Bhattiprolu, Karl Bringmann, Pavel Kolev, Euiwoong Lee, and David P Woodruff. A PTAS for ℓ_p-low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 747–766. SIAM, 2019.

[4] Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. *Geometric and Functional Analysis*, 25(4):1009–1088, 2015.

[5] Christos Boutsidis and Alex Gittens. Improved matrix algorithms via the subsampled randomized Hadamard transform. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1301–1340, 2013.

[6] Christos Boutsidis and David P Woodruff. Optimal CUR matrix decompositions. *SIAM Journal on Computing*, 46(2):543–589, 2017.

[7] Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. Training (overparametrized) neural networks in near-linear time. *arXiv preprint arXiv:2006.11648*, 2020.

[8] Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. *Journal of the ACM (JACM)*, 60(5):31, 2013.

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

[10] Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017.

[11] Michael B Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 278–287. SIAM, 2016.

[12] Michael B Cohen and Richard Peng. L_p row sampling by Lewis weights. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 183–192. ACM, 2015.

[13] Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172, 2015.

[14] Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190. ACM, 2015.

[15] Michael B Cohen, Jelani Nelson, and David P Woodruff. Optimal approximate matrix product in terms of stable rank. *arXiv preprint arXiv:1507.02268*, 2015.

[16] Michael B Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1758–1777. SIAM, 2017.

[17] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 938–942, 2019.

[18] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108 (1):59–91, 2007.

[19] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2(1):225–247, 2006.

[20] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.*, 13:3475–3506, 2012.

[21] Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

[22] Piotr Indyk. Uncertainty principles, extractors, and explicit embeddings of l2 into l1. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 615–620, 2007.

[23] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 910–918. IEEE, 2020.

[24] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. Faster dynamic matrix inverse for faster LPs. *arXiv preprint arXiv:2004.07470*, 2020.

[25] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[26] Yi Li and David Woodruff. Input-sparsity low rank approximation in schatten norm. In *International Conference on Machine Learning*, pages 6001–6009. PMLR, 2020.

[27] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 91–100, 2013.

[28] Cameron Musco and Christopher Musco. Projection-cost-preserving sketches: Proof strategies and constructions. *CoRR*, abs/2004.08434, 2020.

[29] Jelani Nelson and Huy L. Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 117–126, 2013. doi: 10.1109/FOCS.2013.21.

[30] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec): 3413–3430, 2011.

[31] Vladimir Rokhlin and Mark Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36):13212–13217, 2008.

[32] Mark Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60–72, 1999.

[33] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006.

[34] Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 688–701, 2017.

[35] Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2772–2789. Society for Industrial and Applied Mathematics, 2019.

[36] David Wajc. Negative association - definition, properties, and applications, 2017. URL `https://web.stanford.edu/~wajc/notes/NegativeAssociation.pdf`.

[37] Ruosong Wang and David P Woodruff. Tight bounds for ℓ_p oblivious subspace embeddings. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1825–1843. SIAM, 2019.

[38] 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 Missing proofs from Section 7

A.1 Proof of Lemma 7.1

Proof. [Proof of Lemma 7.1] Let $AR = UT$ where U is an orthonormal matrix. As colspan(AR) = colspan(A), we have that $\ell_i^2 = \|U_{i*}\|_2^2$. We first have for any vector x,

$$\|Tx\|_2 = \|UTx\|_2 = \|ARx\|_2 \le \|SARx\|_2 = \|Qx\|_2 = \|x\|_2$$

and

$$\|Tx\|_2 = \|UTx\|_2 = \|ARx\|_2 \ge (1/\beta)\|SARx\|_2 = (1/\beta)\|Qx\|_2 = (1/\beta)\|x\|_2.$$

Here we repeatedly used the facts that Q and U are orthonormal matrices. Thus, we obtain $\|T\|_2 \le 1$ and $\sigma_{\min}(T) \ge 1/\beta$. As $A_{i*}R = U_{i*}T$, we obtain that

$$\|A_{i*}R\|_2 = \|U_{i*}T\|_2 \le \|U_{i*}\|_2\|T\|_2 \le \|U_{i*}\|_2$$

and

$$\|A_{i*}R\|_2 = \|U_{i*}T\|_2 \ge \|U_{i*}\|_2\sigma_{\min}(T) \ge (1/\beta)\|U_{i*}\|_2.$$

Thus, ${\ell_i}^2/\beta^2 \le \|A_{i*}R\|_2^2 \le \ell_i^2$. □

A.2 Proof of Lemma 7.4

Proof. [Proof of Lemma 7.4] Let $u^{(1)}, \dots, u^{(d)}$ be the left singular vectors and $v^{(1)}, \dots, v^{(d)}$ be the right singular vectors. For $j = 1, \dots, k$, let

$$\boldsymbol{X}^{(j)} = \sum_{i:q_i<1} \frac{u_i^{(j)}}{q_i} (E_{i*})^{\mathsf{T}} \boldsymbol{I}[i \text{ is sampled}]$$

and $\boldsymbol{w}^{(j)} = \boldsymbol{X}^{(j)} + \sum_{i:q_i=1} u_i^{(j)} (E_{i*})^{\mathsf{T}} + \mathbb{P}_V A^{\mathsf{T}} u^{(j)}$. We have $\mathbf{E}[\boldsymbol{w}^{(j)}] = A^{\mathsf{T}} u^{(j)} = \sigma_j v^{(j)}$. Now,

$$\mathbf{E}[\|\boldsymbol{w}^{(j)} - \sigma_j v^{(j)}\|_2^2] = \mathbf{E}[\|\boldsymbol{X}^{(j)} - \sum_{i:q_i<1} u_i^{(j)} (E_{i*})^{\mathsf{T}}\|_2^2] = \mathbf{E}[\|\boldsymbol{X}^{(j)}\|_2^2] - \|\sum_{i:q_i<1} u_i^{(j)} (E_{i*})^{\mathsf{T}}\|_2^2.$$

Now,

$$\begin{aligned}\mathbf{E}[\|\boldsymbol{X}^{(j)}\|_2^2] &= \mathbf{E}[\|\sum_{i:q_i<1} \frac{u_i^{(j)}}{q_i} (E_{i*})^{\mathsf{T}} \boldsymbol{I}[i \text{ is sampled}]\|_2^2] \\ &= \sum_{i:q_i<1} \frac{(u_i^{(j)})^2}{q_i^2} \|E_{i*}\|_2^2 q_i + \sum_{i \neq i' : q_i, q_{i'}<1} u_i^{(j)} u_{i'}^{(j)} \langle E_{i*}, E_{i'*} \rangle\end{aligned}$$

As the values p_i used to define probabilities q_i are such that $p_i \geq \alpha \|E_{i*}\|_2^2 / \|E\|_{\mathsf{F}}^2$, then we have

$$\mathbf{E}[\|\boldsymbol{X}^{(j)}\|_2^2] \leq \frac{1}{s\alpha} \|E\|_{\mathsf{F}}^2 + \|\sum_{i:q_i<1} u_i^{(j)} (E_{i*})^{\mathsf{T}}\|_2^2 - \sum_{i:q_i<1} \|u_i^{(j)} (E_{i*})^{\mathsf{T}}\|_2^2.$$

Thus, $\mathbf{E}[\|\boldsymbol{w}^{(j)} - \sigma_j v^{(j)}\|_2^2] \leq (1/s\alpha)\|E\|_{\mathsf{F}}^2 - \sum_{i:q_i<1} \|u_i^{(j)} (E_{i*})^{\mathsf{T}}\|_2^2$. From here, using the same proof as [19], we obtain that the subspace $V + \text{span}(A_S)$ spans rows of a rank k matrix B such that

$$\|A - B\|_{\mathsf{F}}^2 \leq \|A - A_k\|_{\mathsf{F}}^2 + \frac{k}{s\alpha} \|E\|_{\mathsf{F}}^2.$$

□