Numerical Linear Algebra in the Streaming Model

Kenneth L. Clarkson* David P. Woodruff *

April 9, 2009

Abstract

We give near-optimal space bounds in the streaming model for linear algebra problems that include estimation of matrix products, linear regression, low-rank approximation, and approximation of matrix rank. In the streaming model, sketches of input matrices are maintained under updates of matrix entries; we prove results for turnstile updates, given in an arbitrary order. We give the first lower bounds known for the space needed by the sketches, for a given estimation error ϵ. We sharpen prior upper bounds, with respect to combinations of space, failure probability, and number of passes. The sketch we use for matrix A is simply $S^T A$, where S is a sign matrix.

Our results include the following upper and lower bounds on the bits of space needed for 1-pass algorithms. Here A is an $n \times d$ matrix, B is an $n \times d'$ matrix, and $c := d + d'$. These results are given for fixed failure probability; for failure probability $\delta > 0$, the upper bounds require a factor of $\log(1/\delta)$ more space. We assume the inputs have integer entries specified by $O(\log(nc))$ bits, or $O(\log(nd))$ bits.

1. (Matrix Product) Output matrix C with

$$\|A^T B - C\| \leq \epsilon \|A\| \|B\|.$$

We show that $\Theta(c\epsilon^{-2} \log(nc))$ space is needed.

2. (Linear Regression) For $d' = 1$, so that B is a vector b, find x so that

$$\|Ax - b\| \leq (1 + \epsilon) \min_{x' \in \mathbb{R}^d} \|Ax' - b\|.$$

We show that $\Theta(d^2 \epsilon^{-1} \log(nd))$ space is needed.

3. (Rank-k Approximation) Find matrix $\tilde{A}_k$ of rank no more than k, so that

$$\|A - \tilde{A}_k\| \leq (1 + \epsilon) \|A - A_k\|,$$

where A_k is the best rank-k approximation to A. Our lower bound is $\Omega(k\epsilon^{-1}(n + d) \log(nd))$ space, and we give a one-pass algorithm

*IBM Almaden Research Center, San Jose, CA.

matching this when A is given row-wise or column-wise. For general updates, we give a one-pass algorithm needing

$$O(k\epsilon^{-2}(n + d/\epsilon^2)\log(nd))$$

space. We also give upper and lower bounds for algorithms using multiple passes, and a bicriteria low-rank approximation.

1 Introduction

In recent years, starting with [FKV04], many algorithms for numerical linear algebra have been proposed, with substantial gains in performance over older algorithms, at the cost of producing results satisfying Monte Carlo performance guarantees: with high probability, the error is small. These algorithms pick random samples of the rows, columns, or individual entries of the matrices, or else compute a small number of random linear combinations of the rows or columns of the matrices. These compressed versions of the matrices are then used for further computation. The two general approaches are *sampling* or *sketching*, respectively.

Algorithms of this kind are generally also *pass-efficient*, requiring only a constant number of passes over the matrix data for creating samples or sketches, and other work. Most such algorithms require at least two passes for their sharpest performance guarantees, with respect to error or failure probability. However, in general the question has remained of what was attainable in one pass. Such a one-pass algorithm is close to the *streaming* model of computation, where there is one pass over the data, and resource bounds are sublinear in the data size.

Muthukrishnan [Mut05] posed the question of determining the complexity in the streaming model of computing or approximating various linear algebraic functions, such as the best rank-k approximation, matrix product, eigenvalues, determinants, and inverses. This problem was posed again by Sarlós [Sar06], who asked what space and time lower bounds can be proven for any pass-efficient approximate matrix product, ℓ_2 regression, or SVD algorithm.

In this paper, we answer some of these questions. We also study a few related problems, such as rank computation. In many cases we give algorithms together with matching lower bounds. Our algorithms are generally sketching-based, building on and sometimes simplifying prior work on such problems. Our lower bounds are the first for these problems. Sarlós [Sar06] also gives upper bounds for these problems, and our upper bounds are inspired by his and are similar, though a major difference here is our one-pass algorithm for low-rank approximation, improving on his algorithm needing two passes, and our space-optimal one-pass algorithms for matrix product and linear regression, that improve slightly on the space needed for his one-pass algorithms.

We generally consider algorithms and lower bounds in the most general *turnstile model* of computation [Mut05]. In this model an algorithm receives arbitrary updates to entries of a matrix in the form “add x to entry (i, j)”. An entry (i,j) may be updated multiple times and the updates to the different (i,j) can appear in an arbitrary order. Here x is an arbitrary real number of some bounded precision.

The relevant properties of algorithms in this setting are the space required for the sketches; the update time, for changing a sketch as updates are received; the number of passes; and the time needed to process the sketches to produce the final output. Our sketches are matrices, and the final processing is done with standard matrix algorithms. Although sometimes we give upper or lower bounds involving more than one pass, we reserve the descriptor "streaming" for algorithms that need only one pass.

1.1 Results and Related Work

The matrix norm used here will be the Frobenius norm $\|A\|$, where $\|A\| := \left[\sum_{i,j} a_{ij}^2\right]^{1/2}$, and the vector norm will be Euclidean. unless otherwise indicated. The spectral norm $\|A\|_2 := \sup_x \|Ax\|/\|x\|$.

We consider first the Matrix Product problem:

Problem 1.1. Matrix Product. *Matrices A and B are given, with n rows and a total of c columns. The entries of A and B are specified by $O(\log nc)$-bit numbers. Output a matrix C so that*

$$\|A^T B - C\| \leq \epsilon \|A\| \|B\|.$$

Theorem 2.4 on page 10 states that there is a streaming algorithm that solves an instance of this problem with correctness probability at least $1-\delta$, for any $\delta > 0$, and using

$$O(c\epsilon^{-2} \log(nc) \log(1/\delta))$$

bits of space. The update time is $O(\epsilon^{-2} \log(1/\delta))$. This sharpens the previous bounds [Sar06] with respect to the space and update time (for one prior algorithm) and update, final processing time, number of passes (which previously was two), and an $O(\log(1/\delta))$ factor in the space (for another prior algorithm) [Sar06]. We note that it is also seems possible to obtain a one-pass $O(c\epsilon^{-2} \log(nc) \log(c/\delta))$-space algorithm via techniques in [AGMS02, CCFC02, CM05], but the space is suboptimal.

Moreover, Theorem 2.8 on page 13 implies that this algorithm is optimal with respect to space, including for randomized algorithms. The theorem is shown using a careful reduction from an augmented version of the indexing problem, which has communication complexity restated in Theorem 1.6 on page 7.

The sketches in the given algorithms for matrix product, and for other algorithms in this paper, are generally of the form $S^T A$, where A is an input matrix and S is a *sign* matrix, also called a *Rademacher* matrix. Such a sketch satisfies the properties of the Johnson-Lindenstrauss Lemma for random projections, and the upper bounds given here follow readily using that lemma, except that the stronger conditions implied by the JL Lemma require resource bounds that are larger by a $\log n$ factor.

One of the algorithms mentioned above relies on a bound for the higher moments of the error of the product estimate, which is Lemma 2.3 on page 9. The techniques used for that lemma also yield a more general bound for some other matrix norms, given in §A.3 on page 47. The techniques of these bounds are not far from the *trace method* [Vu07], which has been applied to analyzing the eigenvalues of a sign matrix. However, we analyze the use of sign matrices for matrix products, and in a setting of bounded independence, so that trace method analyses don't seem to immediately apply.

The time needed for computing the product $A^T S S^T B$ can be reduced from the immediate $O(dd'm)$, where $m = O(\epsilon^{-2} \log(1/\delta))$, as discussed in §2.3 on page 12, to close to $O(dd')$. When the update regime is somewhat more restrictive than the general turnstile model, the lower bound is reduced by a $\log(nc)$ factor, in Theorem 2.9 on page 14, but the upper bound can be lowered by nearly the same factor, as shown in Theorem 2.5 on page 11, for column-wise updates, which are a special case of this more restrictive model.

Second, we consider the following linear regression problem.

Problem 1.2. Linear Regression. *Given an $n \times d$ matrix A and an $n \times 1$ column vector b, each with entries specified by $O(\log nd)$-bit numbers, output a vector x so that*

$$\|Ax - b\| \leq (1 + \varepsilon) \min_{x' \in \mathbb{R}^d} \|Ax' - b\|.$$

Theorem 3.7 on page 17 gives a lower bound of $\Omega(d^2 \epsilon^{-1} \log(nd))$ space for randomized algorithms for the regression problem (This is under a mild assumption regarding the number of bits per entry, and the relation of n to d.) Our upper bound algorithm requires a sketch with $O(d^2 \epsilon^{-1} \log(1/\delta)))$ entries, with success probability $1 - \delta$, each entry of size $O(\log(nd))$, thus matching the lower bound, and improving on prior upper bounds by a factor of $\log d$ [Sar06].

In Section 4 on page 24, we give upper and lower bounds for low rank approximation:

Problem 1.3. Rank-k Approximation. *Given integer k, value $\epsilon > 0$, and $n \times d$ matrix A, find a matrix $\tilde{A}_k$ of rank at most k so that*

$$\|A - \tilde{A}_k\| \leq (1 + \epsilon)\|A - A_k\|,$$

where A_k is the best rank-k approximation to A.

There have been several proposed algorithms for this problem, but all so far have needed more than 1 pass. A 1-pass algorithm was proposed by Achlioptas and McSherry [AM07], whose error estimate includes an additive term of $\|A\|$; that is, their results are not low relative error. Other work on this problem in the streaming model includes work by Desphande and Vempala [DV06], and by Har-Peled [HP06], but these algorithms require a logarithmic number of passes. Recent work on coresets [FMSW09] solves this problem for measures other than the Frobenius norm, but requires two passes.

We give a one-pass algorithm needing

$$O(k\epsilon^{-2}(n + d/\epsilon^2)\log(nd)\log(1/\delta))$$

space. While this does not match the lower bound (given below), it is the first one-pass rank-k approximation with low relative error; only the trivial $O(nd\log(nd))$-space algorithm was known before in this setting, even for $k = 1$. In particular, this algorithm solves Problem 28 of [Mut05].

The update time is $O(k\epsilon^{-4})$; the total work for updates is thus $O(Nk\epsilon^{-4})$, where N is the number of nonzero entries in A.

We also give a related construction, which may be useful in its own right: a low-relative-error bicriteria low-rank approximation. We show that for appropriate sign matrices S and $\hat{S}$, the matrix $\tilde{A} := A\hat{S}(S^T A\hat{S})^- S^T A$ (where X^- denotes the pseudo-inverse of matrix X) satisfies $\|A - \tilde{A}\| \le (1+\epsilon)\|A - A_k\|$, with probability $1-\delta$. The space needed by these three matrices is $O(k\epsilon^{-1}(n + d/\epsilon)\log(nd)\log(1/\delta))$. The rank of this approximation is at most $k\epsilon^{-1}\log(1/\delta)$. The ideas for this construction are in the spirit of those for the "CUR" decomposition of Drineas *et al.* [DMM08].

When the entries of A are given a column or a row at a time, a streaming algorithm for low-rank approximation with the space bound

$$O(k\epsilon^{-1}(n + d)\log(nd)\log(1/\delta))$$

is achievable, as shown in Theorem 4.5 on page 28. (It should be remarked that under such conditions, it may be possible to adapt earlier algorithms to use one pass.) Our lower bound Theorem 4.10 on page 33 shows that at least $\Omega(k\epsilon^{-1}n)$ bits are needed, for row-wise updates, thus when $n \ge d$, this matches our upper bound up to a factor of $\log(nd)$ for constant δ.

Our lower bound Theorem 4.13 on page 37, for general turnstile updates, is $\Omega(k\epsilon^{-1}(n + d)\log(nd))$, matching the row-wise upper bound. We give an algorithm for turnstile updates, also with space bounds matching this lower bound, but requiring two passes. (An assumption regarding the computation of intermediate matrices is needed for the multi-pass algorithms given here, as discussed in §1.4 on page 7.)

Our lower bound Theorem 4.14 on page 39 shows that even with multiple passes and randomization, $\Omega((n + d)k\log(nd))$ bits are needed for low-rank approximation, and we give an algorithm needing three passes, and $O(nk\log(nd))$ space, for n larger than a constant times $\max\{d/\epsilon, k/\epsilon^2\}\log(1/\delta)$.

In Section 5 on page 39, we give bounds for the following.

Problem 1.4. Rank Decision Problem. *Given an integer k, and a matrix A, output* 1 *iff the rank of A is at least k.*

The lower bound Theorem 5.4 on page 40 states that $\Omega(k^2)$ bits of space are needed by a streaming algorithm to solve this problem with constant probability; the upper bound Theorem 5.1 on page 39 states that $O(k^2\log(n/\delta))$ bits are needed for failure probability at most δ by a streaming algorithm. The lower

Space	Model	Theorem
	Product	
$\Theta(c\epsilon^{-2}\log(nc))$	turnstile	2.1, 2.4, 2.8
$\Omega(c\epsilon^{-2})$	A before B	2.9
$O(c\epsilon^{-2})(\lg\lg(nc)+\lg(1/\epsilon))$	col-wise	2.5
	Regression	
$\Theta(d^2\epsilon^{-1}\log(nd))$	turnstile	3.2, 3.7
$\Omega(d^2(\epsilon^{-1}+\log(nd)))$	insert-once	3.14
	Rank-k Approximation	
$O(k\epsilon^{-2}(n+d\epsilon^{-2})\log(nd))$	turnstile	4.9
$\Omega(k\epsilon^{-1}(n+d)\log(nd))$	turnstile	4.13
$O(k\epsilon^{-1}(n+d)\log(nd))$	row-wise	4.5
$\Omega(k\epsilon^{-1}n)$	row-wise	4.10
$O(k\epsilon^{-1}(n+d)\log(nd))$	2, turnstile	4.4
$O(k(n+d\epsilon^{-1}+k\epsilon^{-2})\log(nd))$	3, row-wise	4.6
$\Omega(k(n+d)\log(nd))$	$O(1)$, turnstile	4.14
	Rank Decision	
$O(k^2\log n)$	turnstile	5.1
$\Omega(k^2)$	turnstile	5.4

Figure 1: Algorithmic upper and lower bounds given here; results are for one pass, unless indicated otherwise under "Model.". All space upper bounds are multiplied by $\log(1/\delta)$ for failure probability δ.

bound is extended to the problem of checking the invertibility of A, and to eigenvalue or determinant estimation with small relative error, by reduction from Rank Decision.

Lower bounds for related problems have been studied in the two-party communication model [CS91, CS95], but the results there only yield bounds for deterministic algorithms in the streaming model. Bar-Yossef [BY03] gives lower bounds for the *sampling complexity* of low rank matrix approximation and matrix reconstruction. We note that it is much more difficult to lower bound the space complexity. Indeed, for estimating the Euclidean norm of a length-n data stream, the sampling complexity is $\Omega(\sqrt{n})$ [BY02], while there is a sketching algorithm achieving $O((\log n)/\varepsilon^2)$ bits of space [AMS99].

1.2 Techniques for the Lower Bounds

Our lower bounds come from reductions from the two-party communication complexity of augmented indexing. Alice is given $x \in \{0,1\}^n$, and Bob is given $i \in [n]$ together with $x_{i+1},\ldots,x_n$. Alice sends a single message to Bob, who must output x_i with probability at least 2/3. Alice and Bob create matrices M_x and M_y, respectively, and use a streaming algorithm to solve augmented indexing.

For regression even obtaining an $\Omega(d^2\log(nd))$ bound is non-trivial. It is tempting for Alice to interpret x as a $d\times d$ matrix M_x with entries drawn randomly from $[nd]$. She sets $A = M_x^{-1}$, which she gives the streaming algorithm. Bob sets b to a standard unit vector, so that the solution is a column of $A^{-1} = M_x$, which can solve augmented indexing.

This argument is flawed because the entries of A may be exponentially small, so A is not a valid input. We instead design b in conjunction with A. We reduce from augmented indexing, rather than indexing (as is often done in streaming), since Bob must use his knowledge of certain entries of A to guarantee that A and b are valid inputs.

To achieve an extra factor of $1/\varepsilon$, we copy this construction $1/\varepsilon$ times. Bob can set b to force a large error on $1/\varepsilon - 1$ of the copies, forcing the regression coefficients to "approximately solve" the remaining copy. This approach loses a $\log(nd)$ factor, and to gain it back we let Bob delete entries that Alice places in A. The $\log(nd)$ factor comes from creating $\log(nd)$ groups, each group containing the $1/\varepsilon$ copies described above. The entries across the $\log(nd)$ groups grow geometrically in size. This idea is inspired by a lower bound for L_p-estimation in [KNW09], though there the authors studied the Gap-Hamming Problem. Of the groups that are not deleted, only one contributes to the error, since the entries in other groups are too small.

1.3 Notation and Terminology

For integer n, let $[n]$ denote $\{1, 2, \dots, n\}$.

A *Rademacher variable* is a random variable that is $+1$ or -1 with probability $1/2$. A *sign* (or *Rademacher*) matrix has entries that are independent Rademacher variables. A *p-wise independent sign matrix* has entries that are Rademacher variables, every subset of p or more entries being independent.

For a matrix A, let $a_{:j}$ denote the jth column of A, and a_{ij} denote the entry at row i and column j. More generally, use an upper case letter for a matrix, and the corresponding lower case for its columns and entries. We may write $a_{:j}^2$ for $\|a_{:j}\|^2$.

We say that matrices C and D are *conforming for multiplication*, or just *conforming*, if the number of columns of C equals the number of rows of D. If the appropriate number of rows and columns of a matrix can be inferred from context, we may omit it.

For a matrix A, let A^- denote the Moore-Penrose pseudo-inverse of A, so that $A^- = V\Sigma^- U^T$, where $A = U\Sigma V^T$ is the singular value decomposition of A.

The following is a simple generalization of the Pythagorean Theorem, and we will cite it that way.

Theorem 1.5. (Pythagorean Theorem) *If C and D matrices with the same number of rows and columns, then $C^T D = 0$ implies $\|C + D\|^2 = \|C\|^2 + \|D\|^2$.*

Proof. By the vector version, we have $\|C + D\|^2 = \sum_i \|c_{:i} + d_{:i}\|^2 = \sum_i \|c_{:i}\|^2 + \|d_{:i}\|^2 = \|C\|^2 + \|D\|^2$. □

For background on communication complexity, see Section 1.5.

1.4 Bit Complexity

We will assume that the entries of an $n \times d$ input matrix are $O(\log(nd))$-bit integers. The sign matrices used for sketches can be assumed to have dimensions bounded by the maximum of n and d. Thus all matrices we maintain have entries of bit size bounded by $O(\log(nd))$. For low-rank approximation, some of the matrices we use in our two and three pass algorithms are rounded versions of matrices that are assumed to be computed exactly in between passes (though not while processing the stream). This "exact intermediate" assumption is not unreasonable in practice, and still allows our upper bounds to imply that the lower bounds cannot be improved.

1.5 Communication Complexity

For lower bounds, we will use a variety of definitions and basic results from two-party communication complexity, as discussed in [KN97]. We will call the two parties Alice and Bob.

For a function $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, we use $R_{\delta}^{1-way}(f)$ to denote the randomized communication complexity with two-sided error at most δ in which only a single message is sent from Alice to Bob. We also use $R_{\mu,\delta}^{1-way}(f)$ to denote the minimum communication of a protocol, in which a single message from Alice to Bob is sent, for solving f with probability at least $1-\delta$, where the probability is taken over both the coin tosses of the protocol and an input distribution μ.

In the augmented indexing problem, which we call $AIND$, Alice is given $x \in \{0, 1\}^n$, while Bob is given both an $i \in [n]$ together with $x_{i+1}, x_{i+2}, \ldots, x_n$. Bob should output x_i.

Theorem 1.6. *([MNSW98])* $R_{1/3}^{1-way}(AIND) = \Omega(n)$ *and also* $R_{\mu,1/3}^{1-way}(AIND) = \Omega(n)$*, where* μ *is uniform on* $\{0, 1\}^n \times [n]$.

Corollary 1.7. *Let A be a randomized algorithm, which given a random $x \in \{0, 1\}^n$, outputs a string $A(x)$ of length m. Let B be an algorithm which given a random $i \in [n]$, outputs a bit $B(A(x))$ so that with probability at least $2/3$, over the choice of x, i, and the coin tosses of A and B, we have $B(A(x))_i = x_i$. Then $m = \Omega(n)$.*

Proof. Given an instance of $AIND$ under distribution μ, Alice sends $A(x)$ to Bob, who computes $B(A(x))_i$, which equals x_i with probability at least $2/3$. Hence, $m = \Omega(n)$. □

Suppose x, y are Alice and Bob's input, respectively. We derive lower bounds for computing $f(x, y)$ on data stream $x \circ y$ as follows. Any r-pass streaming algorithm A yields a $(2r-1)$-round communication protocol for f in the following way. Alice computes $A(x)$ and sends the state of A to Bob, who computes $A(x \circ y)$. Bob sends the state of A back to Alice, who continues the execution of A (the second pass) on $x \circ y$. In the last pass, Bob outputs the answer. The communication is $2r - 1$ times the space of the algorithm.

2 Matrix Products

2.1 Upper Bounds

Given matrices A and B with the same number of rows, suppose S is a sign matrix also with the same number of rows, and with m columns. It is known that

$$\mathbf{E}[A^T SS^T B]/m = A^T\,\mathbf{E}[SS^T]B/m = A^T[mI]B/m = A^T B$$

and

$$\mathbf{E}[\|A^T SS^T B/m - A^T B\|^2] \leq 2\|A\|^2\|B\|^2/m. \tag{1}$$

Indeed, the variance bound (1) holds even when the entries of S are not fully independent, but only 4-wise independent [Sar06]. Such limited independence implies that the storage needed for S is only a constant number of random entries (a logarithmic number of bits), not the nm bits needed for explicit representation of the entries. The variance bound implies, via the Chebyshev inequality, that for given $\epsilon > 0$, there is an $m = O(1/\epsilon^2)$ such that $\|A^T SS^T B/m - A^T B\| \leq \epsilon\|A\|\|B\|$, with probability at least $3/4$. For given $\delta > 0$, we will give two streaming algorithms that have probability of failure at most δ, and whose dependence on δ, in their space and update times is $O(\log(1/\delta))$.

One algorithm relies only on the variance bound; another is slightly simpler, but relies on bounds on higher moments of the error norm that need some additional proof.

The first algorithm is as follows: for a value $p = O(\log(1/\delta))$, but to be determined, maintain p pairs of sketches $S^T A$ and $S^T B$, and use standard streaming algorithms [AMS99] to maintain information sufficient to estimate $\|A\|$ and $\|B\|$ accurately. Compute all product estimates $P_1, P_2, \ldots, P_p$ of $(S^T A)^T S^T B = A^T SS^T B/m$ for each pair of sketches, and consider the Frobenius norm of the difference between P_1 and the remaining estimates. If that Frobenius distance is less than (the estimate of) $\epsilon\|A\|\|B\|/2$, for more than half the remaining product estimates, then return P_1 as the estimate of $A^T B$. Otherwise, do the same test for P_i, for $i = 2 \ldots p$, until some P_i is found that satisfies the test.

Theorem 2.1. *Given $\delta > 0$ and $\epsilon \in (0, 1/3)$, for suitable $m = O(1/\epsilon^2)$ and $p = O(\log(1/\delta))$, the above algorithm returns a product estimate whose error is no more $\epsilon\|A\|\|B\|$, with probability at least $1-\delta$. Using 4-wise independent entries for S, the space required is $O(cmp) = O(c\epsilon^{-2}\log(1/\delta))$.*

Proof. Choose m so that the probability of the event E_i, that $\|P_i - A^T B\| \leq X/4$, is at least $3/4$, where $X := \epsilon\|A\|\|B\|$. Pick p so that the probability that at least $5/8$ of the E_i events occur is at least $1-\delta$. The existence of $m = O(1/\epsilon^2)$ with this property follows from (1), and the existence of $p = O(\log(1/\delta))$ with this property follows from the Chernoff bound. Now assume that at least $5/8$ of the E_i events have indeed occurred. Then for more than half the P_i's, the condition F_i must hold, that for more than half the remaining product estimates P_j,

$$\|P_i - P_j\| \leq \|P_i - A^T B\| + \|A^T B - P_j\| \leq X/2.$$

Suppose P_i satisfies this condition, or since only an estimate of $\|A\|\|B\|$ is available, satisfies $\|P_i - P_j\| \le (1+\epsilon)X/2$ for more than half the remaining product estimates P_j. Then there must be some $P_{j'}$ with both $\|P_i - P_{j'}\| \le (1+\epsilon)X/2$, and $\|A^T B - P_{j'}\| \le X/4$, since the number of P_j not satisfying one or both of the conditions is less than the total. Therefore $\|P_i - A^T B\| \le 3(1+\epsilon)X/4 < X$, for $\epsilon < 1/3$, as desired. Testing for F_i for $i = 1, 2 \ldots n$ thus succeeds in 2 expected steps, with probability at least $1-\delta$, as desired. The space required for storing the random data needed to generate S is $O(p)$, by standard methods, so the space needed is that for storing p sketches $S^T A$ and $S^T B$, each with m rows and a total of c columns. □

While this algorithm is not terribly complicated or expensive, it will also be useful to have an even simpler algorithm, that simply uses a sign matrix S with $m = O(\log(1/\delta)/\epsilon^2)$ columns. As shown below, for such m the estimate $A^T S S^T B$ satisfies the same Frobenius error bound proven above.

This claim is formalized as follows.

Theorem 2.2. *For A and B matrices with n rows, and given $\delta, \epsilon > 0$, there is $m = \Theta(\log(1/\delta)/\epsilon^2)$, as $\epsilon \to 0$, so that for an $n \times m$ sign matrix S,*

$$\mathbf{P}\{\|A^T S S^T B/m - A^T B\| < \epsilon\|A\|\|B\|\} \ge 1 - \delta.$$

This bound holds also when S is a $4\lceil\log(\sqrt{2}/\delta)\rceil$-wise independent sign matrix.

This result can be extended to certain other matrix norms, as discussed in §A.3 on page 47.

Theorem 2.2 is proven using Markov's inequality and the following lemma, which generalizes (1), up to a constant. Here for a random variable X, $\mathbf{E}_p[X]$ denotes $[\mathbf{E}[|X|^p]]^{1/p}$.

Lemma 2.3. *Given matrices A and B, suppose S is a sign matrix with $m > 15$ columns, and A, B, and S have the same number of rows. Then there is an absolute constant C so that for integer $p > 1$ with $m > Cp$,*

$$\mathbf{E}_p\left[\|A^T S S^T B/m - A^T B\|^2\right] \le 4((2p-1)!!)^{1/p}\|A\|^2\|B\|^2/m.$$

This bound holds also when S is $4p$-wise independent.

The proof is given in §A.1 on page 43.

For integer $p \ge 1$, the double factorial $(2p-1)!!$ denotes $(2p-1)(2p-3)\cdots 5 \cdot 3 \cdot 1$, or $(2p)!/2^p p!$. This is the number of ways to partition $[2p]$ into blocks all of size two. From Stirling's approximation,

$$(2p-1)!! = (2p)!/2^p p! \le \frac{\sqrt{2\pi 2p}(2p/e)^{2p} e^{1/24p}}{2^p\sqrt{2\pi p}(p/e)^p e^{1/(12p+1)}} \le \sqrt{2}(2p/e)^p. \tag{2}$$

Thus, the bound of Lemma 2.3 is $O(p)$ as $p \to \infty$, implying that

$$\mathbf{E}_p\left[\|A^T S S^T B/m - A^T B\|\right] = O(\sqrt{p})$$

as $p \to \infty$. It is well known that a random variable X with $\mathbf{E}_p[X] = O(\sqrt{p})$ is subgaussian, that is, its tail probabilities are dominated by those of a normal distribution.

When $B = A$ has one column, so that both are a column vector a, and $m = 1$, so that S is a single vector s, then this bound becomes $\mathbf{E}_p[(1 - (a \cdot s)^2)^2] \leq 4[(2p-1)!!]^{1/p}\|a\|^4$. The Khintchine inequalities give a related bound $\mathbf{E}_p[(a \cdot s)^4] \leq [(4p-1)!!]^{1/p}\|a\|^4$, and an argument similar that the proof of Lemma 2.3 on the previous page can be applied.

The proof of Theorem 2.2 is standard, but included for completeness.

Proof. Let

$$p := \lceil \log(\sqrt{2}/\delta) \rceil$$

and

$$m := \lceil 8p/\epsilon^2 \rceil = \Theta(\log(1/\delta)/\epsilon^2).$$

(Also $p \leq m/2$ for ϵ not too large.)

Applying Markov's inequality and (2) on the preceding page,

$$\begin{aligned}
&\mathbf{P}\{\|A^T SS^T B/m - A^T B\| > \epsilon\|A\|\|B\|\} \\
&\qquad = \mathbf{P}\{\|A^T SS^T B/m - A^T B\|^{2p} > (\epsilon\|A\|\|B\|)^{2p}\} \\
\text{(Markov)}\quad &\qquad \leq (\epsilon\|A\|\|B\|)^{-2p}\,\mathbf{E}\left[\|A^T SS^T B/m - A^T B\|^{2p}\right] \\
\text{(Lemma 2.3)}\quad &\qquad \leq \epsilon^{-2p}(4/m)^p(2p-1)!! \\
&\qquad \leq \sqrt{2}(8p/e\epsilon^2 m)^p \\
\text{(choice of } m)\quad &\qquad \leq e^{-p}\sqrt{2} \\
\text{(choice of } p)\quad &\qquad \leq e^{\log(\delta/\sqrt{2})}\sqrt{2} \\
&\qquad = \delta.
\end{aligned}$$

$\square$

The following algorithmic result is an immediate consequence of Theorem 2.2 on the previous page, maintaining sketches $S^T A$ and $S^T B$, and (roughly) standard methods to generate the entries of S with the independence specified by that theorem.

Theorem 2.4. *Given $\delta, \epsilon > 0$, suppose A and B are matrices with n rows and a total of c columns. The matrices A and B are presented as turnstile updates, using at most $O(\log nc)$ bits per entry. There is a data structure that requires $m = O(\log(1/\delta)/\epsilon^2)$ time per update, and $O(cm\log(nc))$ bits of space, so that at a given time, $A^T B$ may be estimated such that with probability at least $1-\delta$, the Frobenius norm of the error is at most $\epsilon\|A\|\|B\|$.*

2.2 Column-wise Updates

When the entries to A and B are received in column-wise order, a procedure using less space is possible. The sketches are not $S^T A$ and $S^T B$, but instead rounded versions of those matrices. If we receive entries of A (or B) one-by-one in a given column, we can maintain the inner product with each of the rows of S^T exactly using $m \log(cn)$ space. After all the entries of a column of A are known, the corresponding column of $S^T A$ is known, and its m entries can be rounded to the nearest power of $1+\epsilon$. After all updates have been received, we have $\hat{A}$ and $\hat{B}$, where $\hat{A}$ is $S^T A$ where each entry has been rounded, and similarly for $\hat{B}$. We return $\hat{A}^T \hat{B}$ as our output.

The following theorem is an analysis of this algorithm. By Theorem 2.9 on page 14 below, the space bound given here is within a factor of $\lg\lg(nc) + \lg(1/\epsilon)$ of best possible.

Theorem 2.5. *Given $\delta, \epsilon > 0$, suppose A and B are matrices with n rows and a total of c columns. Suppose A and B are presented in column-wise updates, with integer entries having $O(\log(nc))$ bits. There is a data structure so that, at a given time, $A^T B$ may be estimated, so that with probability at least $1-\delta$ the Frobenius norm of the error at most $\epsilon\|A\|\|B\|$. There is $m = O(1/\epsilon^2)$ so that for c large enough, the data structure needs $O(cm \log 1/\delta)(\lg\lg(nc) + \lg(1/\epsilon))$ bits of space.*

Proof. The space required by the above algorithm, including that needed for the exact inner products for a given column, is

$$\lg(cn))/\epsilon^2 + c(\lg\lg(cn) + \lg(1/\epsilon))/\epsilon^2 = c(\lg\lg(cn) + \lg(1/\epsilon))/\epsilon^2$$

for $c > \lg(cn)$.

By Theorem 2.2 on page 9, with probability at least $1-\delta$,

$$\|A^T S S^T B/m - A^T B\| \leq \epsilon\|A\|\|B\|.$$

By expanding terms, one can show

$$\|\hat{A}^T \hat{B}/m - A^T S S^T B/m\| \leq 3(\epsilon/m)\|A^T S\|\|S^T B\|.$$

By the triangle inequality,

$$\|\hat{A}^T \hat{B}/m - A^T B\| \leq \epsilon\|A\|\|B\| + 3(\epsilon/m)\|S^T A\|\|S^T B\|.$$

By Lemma 2.6 on the following page, with probability at least $1-\delta$, $\|S^T A\|/\sqrt{m} \leq (1+\epsilon)\|A\|$, and similarly for $S^T B$. Thus with probability at least $1-3\delta$, $\|\hat{A}^T \hat{B}/m - A^T B\| \leq \epsilon(4+3\epsilon)\|A\|\|B\|$, and the result follows by adjusting constant factors. ∎

The proof depends on the following, where the $\log n$ penalty of JL is avoided, since only a weaker condition is needed.

Lemma 2.6. *For matrix A with n rows, and given $\delta, \epsilon > 0$, there is $m = \Theta(\epsilon^{-2}\log(1/\delta))$, as $\epsilon \to 0$, such that for an $n \times m$ sign matrix S,*

$$\mathbf{P}\{|\|S^T A\|/\sqrt{m} - \|A\|| \leq \epsilon\|A\|\} \geq 1 - \delta.$$

The bound holds when the entries of S are p-wise independent, for large enough p in $O(\log(1/\delta))$.

This tail estimate follows from the moment bound below, which is proven in §A.2 on page 46.

Lemma 2.7. *Given matrix A and sign matrix S with the same number of rows, there is an absolute constant C so that for integer $p > 1$ with $m > Cp$,*

$$\mathbf{E}_p\left[[\|S^T A\|^2/m - \|A\|^2]^2\right] \leq 4((2p-1)!!)^{1/p}\|A\|^4/m.$$

This bound holds also when S is $4p$-wise independent.

2.3 Faster Products of Sketches

The last step of finding our product estimator is computing the product $A^T SS^T B$ of the sketches. We can use fast rectangular matrix multiplication for this purpose. It is known [Cop97, HP98] that for a constant $\gamma > 0$, multiplying an $r \times r^\gamma$ matrix by an $r^\gamma \times r$ matrix can be done in r^2polylog r time. An explicit value of $\gamma = .294$ is given in [Cop97]. Thus, if $m \leq \min(d, d')^{.294}$, then $A^T SS^T B$ can be computed in $O(dd'\text{polylog}\min(d, d'))$ time using block multiplication.

When m is very small, smaller than polylog$(\min(d, d'))$, we can take the above rounding approach even further: that under some conditions it is possible to estimate the sketch product $A^T SS^T B$ more quickly than $O(dd'm)$, even as fast as $O(dd')$, where the constant in the $O(\cdot)$ notation is absolute. As $dd' \to \infty$, if δ and ϵ are fixed (as so m is fixed), the necessary computation is to estimate the dot products of a large number of fixed-dimensional vectors (the columns of $S^T A$ with those of $S^T B$).

Suppose we build an ϵ-cover E for the unit sphere in $\mathbb{R}^m$, and map each column $\hat{a}_{:i}$ of $S^T A$ to $x \in E$ nearest to $\hat{a}_{:i}/\|\hat{a}_{:i}\|$, and similarly map each $\hat{b}_{:j}$ to some $y \in E$. Then the error in estimating $\hat{a}_{:i}^T \hat{b}_{:j}$ by $x^T y\|\hat{a}_{:i}\|\|\hat{b}_{:j}\|$ is at most $3\epsilon\|\hat{a}_{:i}\|\|\hat{b}_{:j}\|$, for ϵ small enough, and the sum of squares of all such errors is at most $9\epsilon^2\|S^T A\|^2\|S^T B\|^2$. By Lemma 2.6, this results in an overall additive error that is within a constant factor of $\epsilon\|A\|\|B\|$, and so is acceptable.

Moreover, if the word size is large enough that a table of dot products $x^T y$ for $x, y \in E$ can be accessed in constant time, then the time needed to estimate $A^T SS^T B$ is dominated by the time needed for at most dd' table lookups, yielding $O(dd')$ work overall.

Thus, under these word-size conditions, our algorithm is optimal with respect to number of passes, space, and the computation of the output from the sketches, perhaps leaving only the update time for possible improvement.

2.4 Lower Bounds for Matrix Product

Theorem 2.8. *Suppose $n \geq \beta(\log_{10} cn)/\epsilon^2$ for an absolute constant $\beta > 0$, and that the entries of A and B are represented by $O(\log(nc))$-bit numbers. Then any randomized 1-pass algorithm which solves Problem 1.1, Matrix Product, with probability at least $4/5$ uses $\Omega(c\epsilon^{-2}\log(nc))$ bits of space.*

Proof. Throughout we shall assume that $1/\varepsilon$ is an integer, and that c is an even integer. These conditions can be removed with minor modifications. Let Alg be a 1-pass algorithm which solves Matrix Product with probability at least 4/5. Let $r = \log_{10}(cn)/(8\epsilon^2)$. We use Alg to solve instances of $AIND$ on strings of size $cr/2$. It will follow by Theorem 1.6 that the space complexity of Alg must be $\Omega(cr) = \Omega(c\log(cn))/\varepsilon^2$.

Suppose Alice has $x \in \{0,1\}^{cr/2}$. She creates a $c/2 \times n$ matrix U as follows. We will have that $U = (U^0, U^1, \ldots, U^{\log_{10}(cn)-1}, Z)$, where for each $k \in \{0, 1, \ldots, \log_{10}(cn) - 1\}$, U^k is a $c/2 \times r/(\log_{10}(cn))$ matrix with entries in the set $\{-10^k, 10^k\}$. Also, Z is a $c/2 \times (n-r)$ matrix consisting of all zeros.

Each entry of x is associated with a unique entry in a unique U^k. If the entry in x is 1, the associated entry in U^k is 10^k, otherwise it is -10^k. Recall that $n \geq \beta(\log_{10}(cn))/\varepsilon^2$, so we can assume that $n \geq r$ provided that $\beta > 0$ is a sufficiently large constant.

Bob is given an index in $[cr/2]$, and suppose this index of x is associated with the (i^*, j^*)-th entry of U^{k^*}. By the definition of the AIND problem, we can assume that Bob is given all entries of U^k for all $k > k^*$. Bob creates a $c/2 \times n$ matrix V as follows. In V, all entries in the first $k^* r/(\log_{10}(cn))$ columns are set to 0. The entries in the remaining columns are set to the negation of their corresponding entry in U. This is possible because Bob has U^k for all $k > k^*$. This is why we chose to reduce from the AIND problem rather than the IND problem. The remaining $n - r$ columns of V are set to 0. We define $A^T = U + V$. Bob also creates the $n \times c/2$ matrix B which is 0 in all but the $((k^*-1)r/(\log_{10}(cn)) + j^*, 1)$-st entry, which is 1. Then,

$$\|A\|^2 = \|A^T\|^2 = \left(\frac{c}{2}\right)\left(\frac{r}{\log_{10}(cn)}\right)\sum_{k=1}^{k^*} 100^k \leq \left(\frac{c}{16\varepsilon^2}\right)\frac{100^{k^*+1}}{99}.$$

Using that $\|B\|^2 = 1$,

$$\varepsilon^2\|A\|^2\|B\|^2 \leq \varepsilon^2\left(\frac{c}{16\varepsilon^2}\right)\frac{100^{k^*+1}}{99} = \frac{c}{2}\cdot 100^{k^*}\cdot\frac{25}{198}.$$

$A^T B$ has first column equal to the j^*-th column of U^{k^*}, and remaining columns equal to zero. Let C be the $c/2 \times c/2$ approximation to the matrix $A^T B$. We say an entry $C_{\ell,1}$ is *bad* if its sign disagrees with the sign of $(A^T B)_{\ell,1}$. If an entry $C_{\ell,1}$ is bad, then $((A^T B)_{\ell,1} - C_{\ell,1})^2 \geq 100^{k^*}$. Thus, the fraction of bad entries is at most $\frac{25}{198}$. Since we may assume that i^*, j^*, and k^* are chosen independently of x, with probability at least 173/198, $\mathrm{sign}(C_{i^*,1}) = \mathrm{sign}(U^{k^*}_{i^*,j^*})$.

Alice runs Alg on U in an arbitrary order, transmitting the state to Bob, who continues the computation on V and then on B, again feeding the entries into Alg in an arbitrary order. Then with probability at least $4/5$, over Alg's internal coin tosses, Alg outputs a matrix C for which $\|A^T B - C\|^2 \leq \varepsilon^2 \|A\|^2 \|B\|^2$.

It follows that the parties can solve the AIND problem with probability at least $4/5 - 25/198 > 2/3$. The theorem now follows by Theorem 1.6.

□

For a less demanding computational model, we have:

Theorem 2.9. *Suppose $n \geq \beta/\epsilon^2$ for an absolute constant $\beta > 0$, and that the entries of A and B are represented by $O(\log(nc))$-bit numbers. Then even if each entry of A and B appears exactly once in the stream, for every ordering of the entries of A and B for which every entry of A appears before every entry of B, any randomized 1-pass algorithm which solves Problem 1.1, Matrix Product, with probability at least $4/5$ uses $\Omega(c\epsilon^{-2})$ bits of space.*

Proof. The proof is very similar to the proof of Theorem 2.8, so we only highlight the differences. Now we set $r = 1/(8\varepsilon^2)$. Instead of reducing from the AIND problem, we reduce from IND on instances of size $cr/2$, which is the same problem as AIND, except Bob does not receive $x_{i+1}, \ldots, x_{cr/2}$. It is well-known that $R^{1-way}_{\mu,1/3}(IND) = \Omega(cr)$, where μ is the uniform distribution on $\{0,1\}^{cr/2} \times [cr/2]$. We use Alg to solve IND.

This time Alice simply sets U to equal (U^0, Z). Bob is given an (i^*, j^*) and his task is to recover $U^0_{i^*,j^*}$. This time $A^T = U$ and B contains a single non-zero entry in position $(j^*, 1)$, which contains a 1. It follows that the first column of $A^T B$ equals the j^*-th column of U^0, and the remaining columns are zero. We now have $\|A\|^2 = \frac{c}{16\varepsilon^2}$, $\|B\|^2 = 1$, and so $\varepsilon^2 \|A\|^2 \|B\|^2 = \frac{c}{16}$. Defining a bad entry as before, we see that the fraction of bad entries is at most $1/8$, and so the parties can solve the IND problem with probability at least $4/5 - 1/8 > 2/3$. □

3 Regression

3.1 Upper Bounds

Our algorithm for regression is a consequence of the following theorem. For convenience of application of this result to algorithms for low-rank approximation, it is stated with multiple right-hand sides: that is, the usual vector b is replaced by a matrix B. Moreover, while the theorem applies to a matrix A of rank at most k, we will apply it to regression with the assumption that A has $d \leq n$ columns implying an immediate upper bound of d on the rank. This also is for convenience of application to low-rank approximation.

Theorem 3.1. *Given $\delta, \epsilon > 0$, suppose A and B are matrices with n rows, and A has rank at most k. There is an $m = O(k \log(1/\delta)/\epsilon)$ such that, if S is an $n \times m$ sign matrix, then with probability at least $1 - \delta$, if $\tilde{X}$ is the solution to*

$$\min_X \|S^T(AX - B)\|^2, \tag{3}$$

and X^* is the solution to

$$\min_X \|AX - B\|^2, \tag{4}$$

then

$$\|A\tilde{X} - B\| \leq (1+\epsilon)\|AX^* - B\|.$$

The entries of S need be at most $\eta(k+\log(1/\delta))$-wise independent, for a constant η.

This theorem has the following immediate algorithmic implication.

Theorem 3.2. *Given $\delta, \epsilon > 0$, and $n \times d$ matrix A, and n-vector b, sketches of A and b of total size*

$$O(d^2\epsilon^{-1}\log(1/\delta)\log(nd))$$

can be maintained under turnstile updates, so that a vector $\tilde{x}$ can be found using the sketches, so that with probability at least $1-\delta$,

$$\|A\tilde{x} - b\| \leq (1+\epsilon)\|Ax^* - b\|,$$

where x^* minimizes $\|Ax - b\|$. The update time is

$$O(d\epsilon^{-2}\log(1/\delta)).$$

The proof of Theorem 3.1 on the previous page is not far that of from Theorem 12 of [Sar06], or that in [DMMS07]. The following lemma is crucial.

Lemma 3.3. *For A, B, X^*, and $\tilde{X}$ as in Theorem 3.1 on the preceding page,*

$$\|A(\tilde{X} - X^*)\| \leq 2\sqrt{\epsilon}\|B - AX^*\|$$

Proof. Before giving a proof of Theorem 3.1 on the previous page, we state a lemma limiting the independence needed for S to satisfy a particular spectral bound, also some standard lemmas, and the proof of Lemma 3.3.

Lemma 3.4. *Given integer k and $\epsilon, \delta > 0$, there is $m = O(k\log(1/\delta)/\epsilon)$ and an absolute constant η such that if S is an $n \times m$ sign matrix with $\eta(k+\log(1/\delta))$-wise independent entries, then for $n \times k$ matrix U with orthonormal columns, with probability at least $1-\delta$, the spectral norm $\|U^TSS^TU - I\|_2 \leq \epsilon$.*

Proof. As in the proof of Corollary 11 of [Sar06], by Lemma 10 of that reference, it is enough to show that with failure probability $\eta^{-k}\delta$, for given k-vectors x, y with no more than unit norm, $|x^TU^TSS^TUy/m - x^Ty| \leq \alpha\epsilon$, for absolute constant $\alpha > 0$ and $\eta > 1$. This bound follows from Theorem 2.2 on page 9, for the given m and $\eta(k+\log(1/\delta))$-wise independence. $\square$

Lemma 3.5. *If matrix U has columns that are unit vectors and orthogonal to each other, then $UU^TU = U$. If C then $\|U^TUC\| = \|UC\|$.*

Proof. The proof of the first fact is omitted. For the second,

$$\|U^T UC\|^2 = \operatorname{trace} C^T U^T U U^T U C = \operatorname{trace} C^T U^T U C = \|UC\|^2.$$

$\square$

Lemma 3.6. *Given $n \times d$ matrix C, and $n \times d'$ matrix D consider the problem*

$$\min_{X \in \mathbb{R}^{d \times d'}} \|CX - D\|^2.$$

The solution to this problem is $X^* = C^- D$, where C^- is the Moore-Penrose inverse of C. Moreover, $C^T(CX^* - D) = 0$, and so if c is any vector in the column space of C, then $c^T(CX^* - D) = 0$.

Proof. Omitted. $\square$

The system $C^T CX = C^T D$ is called the *normal equations* for the regression problem. While the regression problem is commonly stated with $d' = 1$, the generalization to $d' > 1$ is immediate.

Proof. (of Lemma 3.3 on the preceding page) Let $A = U\Sigma V^T$ denote the singular value decomposition of A. Since A has rank at most k, we can consider U and V to have at most k columns.

By Lemma 3.5 on the previous page, it is enough to bound $\|\beta\|$, where $\beta := U^T A(\tilde{X} - X^*)$.

We use the normal equations for (3) on page 14,

$$U^T SS^T (A\tilde{X} - B) = A^T SS^T (A\tilde{X} - B) = 0. \tag{5}$$

To bound $\|\beta\|$, we bound $\|U^T SS^T U\beta\|$, and then show that this implies that $\|\beta\|$ is small. Using Lemma 3.5 and (5) we have

$$\begin{aligned} U^T SS^T U\beta &= U^T SS^T U U^T A(\tilde{X} - X^*) \\ &= U^T SS^T A(\tilde{X} - X^*) + U^T SS^T (B - A\tilde{X}) \\ &= U^T SS^T (B - AX^*). \end{aligned}$$

Using the normal equations (6) and Theorem 2.2, and appropriate $m = O(k \log(1/\delta)/\epsilon)$, with probability at least $1 - \delta$,

$$\begin{aligned} \|U^T SS^T U\beta/m\| &= \|U^T SS^T (B - AX^*)/m\| \\ &\leq \sqrt{\epsilon/k}\|U\|\|B - AX^*\| \\ &\leq \sqrt{\epsilon}\|B - AX^*\|. \end{aligned}$$

To show that this bound implies that $\|\beta\|$ is small, we use the property of any conforming matrices C and D, that $\|CD\| \leq \|C\|_2 \|D\|$, obtaining

$$\begin{aligned} \|\beta\| &\leq \|U^T SS^T U\beta/m\| + \|U^T SS^T U\beta/m - \beta\| \\ &\leq \sqrt{\epsilon}\|B - AX^*\| + \|U^T SS^T U/m - I\|_2 \|\beta\|. \end{aligned}$$

By Lemma 3.4 on page 15, with probability at least $1-\delta$, $\|U^T SS^T U/m - I\|_2 \leq \epsilon_0$, for $m \geq kC\log(1/\delta)/\epsilon_0^2$ and an absolute constant C. Thus $\|\beta\| \leq \sqrt{\epsilon}\|B - AX^*\| + \epsilon_0\|\beta\|$, or

$$\|\beta\| \leq \sqrt{\epsilon}\|B - AX^*\|/(1-\epsilon_0) \leq 2\sqrt{\epsilon}\|B - AX^*\|,$$

for $\epsilon_0 \leq 1/2$. This bounds $\|\beta\|$, and so proves the claim. ☐

Now for the proof of Theorem 3.1. Again we let $U\Sigma V^T$ denote the SVD of A.

From the normal equations for (4) on page 15, and since U and A have the same columnspace,

$$U^T(AX^* - B) = A^T(AX^* - B) = 0. \tag{6}$$

This and Theorem 1.5, the Pythagorean Theorem, imply

$$\|A\tilde{X} - B\|^2 = \|AX^* - B\|^2 + \|A(\tilde{X} - X^*)\|^2, \tag{7}$$

which with Lemma 3.3 on page 15, implies that with probability at least $1-2\delta$,

$$\|A\tilde{X} - B\| \leq (1+4\epsilon)\|AX^* - B\|.$$

Adjusting and renaming δ and ϵ, and folding the changes into m, the result follows. ☐

3.2 Lower Bounds for Regression

Theorem 3.7. *Suppose $n \geq d(\log_{10}(nd))/(36\epsilon)$ and d is sufficiently large. Then any randomized 1-pass algorithm which solves the Linear Regression problem with probability at least $7/9$ needs $\Omega(d^2\epsilon^{-1}\log(nd))$ bits of space.*

Proof. Throughout we shall assume that $\varepsilon < 1/72$ and that $u := 1/(36\varepsilon)$ is an integer. Put $L := nd$.

We reduce from the AIND problem on strings of length $d(d-1)(\log_{10} L)/(72\epsilon)$. Alice interprets her input string as a $d(\log_{10} L)u \times d$ matrix A, which is constructed as follows.

For each $z \in \{0, \ldots, \log_{10} L - 1\}$ and each $k \in [u]$, we define an upper-triangular $d \times d$ matrix $A^{z,k}$. We say that $A^{z,k}$ is in *level* z and *band* k. The matrix $A^{z,k}$ consists of random $\{-10^z, +10^z\}$ entries inserted above the diagonal from Alice's input string. The diagonal entries of the matrices will be set by Bob. A is then the $d(\log_{10} L)u \times d$ matrix obtained by stacking the matrices $A^{0,1}, A^{0,2}, \ldots, A^{0,u}, A^{1,1}, \ldots, A^{1,u}, \ldots, A^{\log_{10} L-1,u}$ on top of each other.

Bob has an index in the AIND problem, which corresponds to an entry $A^{z^*,k^*}_{i^*,j^*}$ for a $z^* \in \{0, 1, \ldots, \log_{10} L - 1\}$, a $k^* \in [u]$ and an $i^* < j^*$. Put $Q := 100^{z^*}(j^* - 1)$. Bob's input index is random and independent of Alice's input, and therefore, conditioned on the value of j^*, the value of i^* is random subject to the constraint $i^* < j^*$. Notice, in particular, that $j^* > 1$.

By definition of the AIND problem, we can assume Bob is given the entries in $A^{z,k}$ for all $z > z^*$ and each $k \in [u]$.

Let P be a large positive integer to be determined. Bob sets the diagonal entries of A as follows. Only matrices in level z^* have non-zero diagonal entries. Matrix A^{z^*,k^*} has all of its diagonal entries equal to P. The remaining matrices $A^{z^*,k}$ in level z^* in bands $k \neq k^*$ have $A^{z^*,k}_{j,j} = P$ whenever $j \geq j^*$, and $A^{z^*,k}_{j,j} = 0$ whenever $j < j^*$.

Alice feeds her entries of A into an algorithm Alg which solves the linear regression problem with probability at least 7/9, and transmits the state to Bob. Bob then feeds his entries of A into Alg. Next, using the entries that Bob is given in the AIND problem, Bob sets all entries of matrices $A^{z,k}$ in levels $z > z^*$ to 0, for every band k.

Bob creates the $d(\log_{10} L)u \times 1$ column vector b as follows. We think of b as being composed of $(\log_{10} L)u$ vectors $b^{z,k}$, $z \in \{0, \ldots, \log_{10} L - 1\}$, $k \in [u]$, so that b is the vector obtained by stacking $b^{0,1}, b^{0,2}, \ldots, b^{0,u}, \ldots, b^{\log_{10} L-1,u}$ on top of each other. We say $b^{z,k}$ is in level z and band k.

For any $x \in \mathbb{R}^d$, the squared error of the linear regression problem is $\|Ax - b\|^2 = \sum_{z=0}^{\log_{10} L-1} \sum_{k=1}^{u} \|A^{z,k}x - b^{z,k}\|^2$. For all vectors $b^{z^*,k}$ in level z^*, Bob sets $b^{z^*,k}_{j^*} = P$. He sets all other entries of b to 0, and feeds the entries of b to Alg.

We will show in Lemma 3.8 below that there exists a vector $x \in \mathbb{R}^d$ for which $\|Ax - b\|^2 \leq Q\left(u - \frac{97}{99}\right)$. It will follow by Lemma 3.12 that the vector x^* output by Alg satisfies various properties useful for recovering individual entries of A^{z^*,k^*}. By Lemma 3.13, it will follow that for most (j, j^*) pairs that Bob could have, the entry A^{z^*,k^*}_{j,j^*} can be recovered from x^*_j, and so this is also likely to hold of the actual input pair (i^*, j^*). Hence, Alice and Bob can solve the AIND problem with reasonable probability, thereby giving the space lower bound.

Consider the vector $x \in \mathbb{R}^d$ defined as follows. Let $x_j = 0$ for all $j > j^*$. Let $x_{j^*} = 1$. Finally, for all $j < j^*$, let $x_j = -A^{z^*,k^*}_{j,j^*}/P$.

Lemma 3.8. $\|Ax - b\|^2 \leq Q\left(u - \frac{97}{99}\right)$.

Proof. We start with three claims.

Claim 3.9. $(A^{z,k}x - b^{z,k})_j = 0$ *whenever* $z > z^*$.

Proof. For $z > z^*$ and any k, $A^{z,k}$ is the zero matrix and $b^{z,k}$ is the zero vector. □

Claim 3.10. *For all* $j \geq j^*$, $(A^{z,k}x - b^{z,k})_j = 0$.

Proof. By Claim 3.9, if $z > z^*$, for all k, for all j, $(A^{z,k}x - b^{z,k})_j = 0$. So suppose $z \leq z^*$. For $j > j^*$, $b^{z,k}_j = 0$. Since $j > j^*$, $(A^{z,k}x)_j = \sum_{j'=1}^{d} A^{z,k}_{j,j'}x_{j'} = 0$ since $A^{z,k}_{j,1} = \cdots = A^{z,k}_{j,j^*} = 0$, while $x_{j^*+1} = \cdots = x_d = 0$.

For $j = j^*$, $b^{z,k}_{j^*} = P$ if $z = z^*$, and is otherwise equal to 0. We also have $(A^{z,k}x)_{j^*} = A^{z,k}_{j^*,j^*}x_{j^*}$. If $z \neq z^*$, this is 0 since $A^{z,k}_{j^*,j^*} = 0$, and so in this case

$(A^{z,k}x - b^{z,k})_j = 0$. If $z = z^*$, this quantity is P, but then $b_{j^*}^{z^*,k} = P$, and so again $(A^{z,k}x - b^{z,k})_j = 0$. $\square$

Set $P = d^2L^4$. The number of bits needed to describe P is $O(\log L)$.

Claim 3.11. *For all $j < j^*$,*

- *For $(z,k) = (z^*,k^*)$, $(A^{z^*,k^*}x - b^{z^*,k^*})_j^2 \leq \frac{1}{d^2L^4}$.*
- *For $(z,k) \neq (z^*,k^*)$, $(A^{z,k}x - b^{z,k})_j^2 \leq 100^z + \frac{3}{dL}$.*

Proof. Notice that for $j < j^*$, $(A^{z,k^*}x - b^{z,k^*})_j$ equals

$$\begin{aligned}
& (A^{z,k}x)_j \\
= \quad & \sum_{j'=0}^{d} A_{j,j'}^{z,k} x_{j'}. \\
= \quad & \sum_{j'=0}^{j-1} A_{j,j'}^{z,k} x_{j'} + A_{j,j}^{z,k} x_j + \sum_{j'=j+1}^{j^*-1} A_{j,j'}^{z,k} x_{j'} + A_{j,j^*}^{z,k} x_{j^*} + \sum_{j'>j^*} A_{j,j'}^{z,k} x_{j'} \\
= \quad & 0 + A_{j,j}^{z,k}\left(\frac{-A_{j,j^*}^{z^*,k^*}}{P}\right) - \frac{1}{P}\cdot\sum_{j'=j+1}^{j^*-1} A_{j,j'}^{z^*,k^*} A_{j',j^*}^{z^*,k^*} + A_{j,j^*}^{z^*,k^*} + 0,
\end{aligned}$$

which by our choice of P, lies in the interval

$$\left[A_{j,j}^{z,k}\left(\frac{-A_{j,j^*}^{z^*,k^*}}{P}\right) + A_{j,j^*}^{z,k} - \frac{1}{dL^2},\quad A_{j,j}^{z,k}\left(\frac{-A_{j,j^*}^{z^*,k^*}}{P}\right) + \ A_{j,j^*}^{z,k} + \frac{1}{dL^2}\right].$$

Now, if $(z,k) = (z^*,k^*)$, $A_{j,j}^{z,k} = P$, in which case the interval becomes $\left[-\frac{1}{dL^2},\quad +\frac{1}{dL^2}\right]$. Hence, $(A^{z^*,k^*}x - b^{z^*,k^*})_j^2 \leq \frac{1}{d^2L^4}$. On the other hand, if $(z,k) \neq (z^*,k^*)$, then since $j < j^*$, $A_{j,j}^{z,k} = 0$, in which case the interval becomes $\left[A_{j,j^*}^{z,k} - \frac{1}{dL^2},\quad A_{j,j^*}^{z,k} + \frac{1}{dL^2}\right]$. Since, $A_{j,j^*}^{z,k} \in \{-10^z, 0, +10^z\}$, $|(A^{z,k}x - b^{z,k})_j| \leq 10^z + \frac{1}{dL^2}$. Using that $z \leq \log_{10} L - 1$, we have, $(A^{z,k}x - b^{z,k})_j^2 \leq 100^z + \frac{2}{dL} + \frac{1}{d^2L^4} \leq 100^z + \frac{3}{dL}$. $\square$

From Claim 3.9, Claim 3.10, and Claim 3.11, we deduce:

- For any $z > z^*$ and any k, $\|A^{z,k}x - b^{z,k}\|^2 = 0$.
- $\|A^{z^*,k^*}x - b^{z^*,k^*}\|^2 = \sum_j (A^{z^*,k^*}x - b^{z^*,k^*})_j^2 \leq \frac{1}{dL^4}$.
- For any $z \leq z^*$ and any $k \neq k^*$, $\|A^{z,k}x - b^{z,k}\|^2 = \sum_{j<j^*}(A^{z,k}x - b^{z,k})_j^2 \leq 100^z(j^*-1) + \frac{3}{L}$, where the inequality follows from the fact that we sum over at most d indices.

For $z = z^*$, using that $u - 1 = 1/(36\varepsilon) - 1 \geq 2$,

$$\begin{aligned}\sum_{k=1}^{u} \|A^{z^*,k}x - b^{z^*,k}\|^2 &\leq (u-1)\left[Q + \frac{3}{L}\right] + \frac{1}{dL^4} \\ &\leq (u-1)\left[Q + \frac{4}{L}\right],\end{aligned}$$

for sufficiently large d. Moreover,

$$\begin{aligned}\sum_{z<z^*}\sum_{k=1}^{u} \|A^{z,k}x - b^{z,k}\|^2 &\leq \sum_{z<z^*} u\left[100^z(j^*-1) + \frac{4}{L}\right] \\ &\leq \left[\frac{Qu}{99}\right] + \frac{\log_{10} L}{9\varepsilon L}.\end{aligned}$$

We can bound the total error of x by adding these quantities,

$$\sum_{z=0}^{\log_{10} L-1}\sum_{k=1}^{u} \|A^{z,k}x - b^{z,k}\|^2 \leq Q\left(u - \frac{98}{99}\right) + Err$$

where $Err = \frac{1}{9\varepsilon L} + \frac{\log_{10} L}{9\varepsilon L}$. Now, using the bound in the theorem statement, $\frac{1}{9\varepsilon} \leq \frac{4n}{d \log_{10} L}$, where the latter is upper-bounded by $\frac{n}{2d}$ for sufficiently large d. Hence, $\frac{1}{9\varepsilon L} \leq \frac{1}{2d^2}$. Moreover, $\frac{\log_{10} L}{9\varepsilon L}$ is at most $\frac{4n \log_{10} L}{nd^2 \log_{10} L} \leq \frac{4}{d^2}$. It follows that $Err < \frac{5}{d^2}$. For sufficiently large d, $\frac{5}{d^2} \leq \frac{Q}{99}$, and so

$$\sum_{z=0}^{\log_{10} L-1}\sum_{k=1}^{u} \|A^{z,k}x - b^{z,k}\|^2 \leq Q\left(u - \frac{97}{99}\right),$$

and the lemma follows. □

Let x^* be the output of Alg. Then, using Lemma 3.8, and the fact that $u = 1/(36\varepsilon)$, with probability at least $7/9$

$$\begin{aligned}\|Ax^* - b\|^2 \leq (1+\varepsilon)^2\|Ax - b\|^2 &\leq (1+3\varepsilon)\left(u - \frac{97}{99}\right)Q \\ &\leq \left(u - \frac{97}{99} + \frac{1}{12}\right)Q \leq \left(u - \frac{43}{48}\right)Q. \qquad (8)\end{aligned}$$

Call this event $\mathcal{E}$. We condition on $\mathcal{E}$ occurring in the remainder of the proof.

Lemma 3.12. *The following conditions hold simultaneously:*

1. *For all $j > j^*$, $x_j^* \in [-L^2/P, L^2/P]$.*
2. *For $j = j^*$, $x_j^* \in [1 - L^2/P, 1 + L^2/P]$.*
3. *For $j < j^*$, $x_j^* \in [-L^2/P, L^2/P]$.*

Proof. Notice that the occurrence of event $\mathcal{E}$ in (8) implies that

$$\sum_{k=1}^{u} \|A^{z^*,k}x^* - b^{z^*,k}\|^2 \leq \|Ax^* - b\|^2 \leq udL^2,$$

since we have both $100^{z^*} \leq L^2$ and $j^* - 1 \leq d$. Notice, also, that $u \leq \frac{n}{d}$, and so we have that

$$\sum_{k=1}^{u} \|A^{z^*,k}x^* - b^{z^*,k}\|^2 \leq nL^2. \tag{9}$$

To prove condition 1, suppose for some $j > j^*$ the condition were false. Let j be the largest index greater than j^* for which the condition is false. Then for each $k \in [u]$,

$$(A^{z^*,k}x^* - b^{z^*,k})_j^2 \quad = \quad (Px_j^* + \sum_{j'=j+1}^{d} A_{j,j'}^{z^*,k} x_{j'}^*)^2,$$

using that $A_{j,j'}^{z^*,k} = 0$ for $j' < j$. To lower bound the RHS, we can assume that the sign of x_j^* differs from the sign of each of $A_{j,j'}^{z^*,k} x_{j'}^*$. Moreover, since j is the largest index for which the condition is false, the RHS is lower bounded by

$$\begin{aligned} (PL^2/P - d \cdot 10^{z^*} L^2/P)^2 &\geq (L^2 - dL^3/P)^2 \\ &= (L^2 - 1/(dL))^2 \\ &\geq L^4/4, \end{aligned}$$

where the first inequality follows from the fact that $|A_{j,j'}^{z^*,k}| = 10^{z^*}$ and $10^{z^*} \leq L$, while the final inequality follows for sufficiently large d. Thus, $\|Ax^* - b\|^2 \geq L^4/4$. But by inequality (9), $\|Ax^* - b\|^2 \leq nL^2$, which is a contradiction.

We now prove condition 2. Suppose that $x_{j^*}^*$ did not lie in the interval $[1 - L^2/P, 1 + L^2/P]$. Now, $b_{j^*}^{z^*,k^*} = P$. Hence,

$$\begin{aligned} \|Ax^* - b\|^2 &\geq (A^{z^*,k^*}x^* - b^{z^*,k^*})_{j^*}^2 \\ &= \left(Px_{j^*}^* + \sum_{j'=j^*+1}^{d} A_{j^*,j'}^{z^*,k^*} x_{j'}^* - P\right)^2 \\ &\geq \left(L^2 - d \cdot 10^{z^*} L^2/P\right)^2 \\ &\geq \left(L^2 - dL^3/P\right)^2 \\ &\geq L^4/4, \end{aligned}$$

where the second inequality uses condition 1, the third inequality uses that $10^{z^*} \leq L$, while the final inequality holds for sufficiently large d. This contradicts inequality (9).

To prove condition 3, let $j < j^*$ be the largest value of j for which $x_j^* \notin [-L^2/P, L^2/P]$. Then using conditions 1 and 2,

$$\begin{aligned}
\|Ax^* - b\|^2 &\geq \left(Px_j^* + \sum_{j'=j+1}^{j^*-1} A_{j,j'}^{z^*,k^*} x_{j'}^* + A_{j,j^*}^{z^*,k^*} x_{j^*}^* + \sum_{j'=j^*+1}^{d} A_{j,j'}^{z^*,k^*} x_{j'}^* \right)^2 \\
&\geq (L^2 - d10^{z^*} L^2/P - 1)^2 \\
&\geq (L^2 - dL^3/P - 1)^2 \\
&\geq L^4/4,
\end{aligned}$$

where the last inequality holds for sufficiently large d. This again contradicts inequality (9). $\square$

Lemma 3.13. *With probability at least* $1 - 49/d$*, for at least a* $41/46$ *fraction of the indices* $j < j^*$*, we have*

$$\operatorname{sign}(x_j^*) = -\operatorname{sign}(A_{j,j^*}^{z^*,k^*}).$$

Notice that $A_{j,j^*}^{z^*,k^*} \in \{-10^{z^*}, 10^{z^*}\}$*, so its sign is well-defined.*

Proof. To prove this, we first bound $\sum_{k \neq k^*} \|A^{z^*,k} x^* - b^{z^*,k}\|^2$. Fix an arbitrary $k \neq k^*$. We may lower bound $\|A^{z^*,k} x^* - b^{z^*,k}\|^2$ by $\sum_{j<j^*} (A^{z^*,k} x^* - b^{z^*,k})_j^2$. For any $j < j^*$, we have

$$(A^{z^*,k}x^* - b^{z^*,k})_j^2 = (A^{z^*,k}x^*)_j^2 = \left(\sum_{j'=j}^{j^*-1} A_{j,j'}^{z^*,k} x_{j'}^* + A_{j,j^*}^{z^*,k} x_{j^*}^* + \sum_{j'=j^*+1}^{d} A_{j,j'}^{z^*,k} x_{j'}^* \right)^2.$$

By Conditions 1, 2, and 3 of Lemma 3.12, this is at least

$$100^{z^*} \left(1 - \frac{dL^2}{P}\right)^2 \geq 100^{z^*} \left(1 - \frac{2dL^2}{P}\right).$$

It follows that

$$\|A^{z^*,k}x^* - b^{z^*,k}\|^2 \geq Q \left(1 - \frac{2dL^2}{P}\right),$$

and hence

$$\begin{aligned}
\sum_{k \neq k^*} \|A^{z^*,k}x^* - b^{z^*,k}\|^2 &\geq (u-1)\, Q \left(1 - \frac{2dL^2}{P}\right) \\
&\geq (u-1)\, 100^{z^*} \left[j^* - 1 - \frac{2d^2L^2}{P} \right].
\end{aligned}$$

On the other hand, since event $\mathcal{E}$ occurs,

$$\sum_k \|A^{z^*,k}x^* - b^{z^*,k}\|^2 \leq \|Ax^* - b\|^2 \leq \left(u - \frac{43}{48}\right) Q,$$

and so

$$\begin{aligned}\|A^{z^*,k^*}x^* - b^{z^*,k^*}\|^2 &\leq \frac{5\cdot Q}{48} + (u-1)\,\frac{100^{z^*}\cdot 2d^2L^2}{P}\\ &\leq \frac{5\cdot Q}{48} + \frac{100^{z^*}2d^2L^2u}{d^2L^4},\end{aligned}$$

and using that $u = \frac{1}{36\varepsilon} \leq \frac{n}{d\log_{10}L} \leq \frac{n}{2d}$ (for d sufficiently large), we have

$$\|A^{z^*,k^*}x^* - b^{z^*,k^*}\|^2 \leq \frac{5}{48}\cdot Q + \frac{100^{z^*}n}{d(nd)^2} \leq \frac{5}{48}\cdot 100^{z^*}j^*.$$

Now suppose that the sign of x^*_j agrees with the sign of A^{z^*,k^*}_{j,j^*} for some $j < j^*$. Then consider $(A^{z^*,k^*}x^* - b^{z^*,k^*})^2_j = (A^{z^*,k^*}x^*)^2_j = \left(\sum_{j'=1}^d A^{z^*,k^*}_{j,j'}x^*_{j'}\right)^2$, which, since $A^{z^*,k^*}_{j,j'} = 0$ for $j' < j$ and $A^{z^*,k^*}_{j,j} = P$, in turn equals

$$\left(Px^*_j + \sum_{j'=j+1}^{j^*-1} A^{z^*,k^*}_{j,j'}x^*_{j'} + A^{z^*,k^*}_{j,j^*}x^*_{j^*} + \sum_{j'=j^*+1}^{d} A^{z^*,k^*}_{j,j'}x^*_{j'}\right)^2.$$

Using conditions 1, 2, and 3 of Lemma 3.12, this is at least

$$\left(P|x^*_j| + 10^{z^*}\left(1 - \frac{dL^2}{P}\right)\right)^2 \geq 100^{z^*}\left(1 - \frac{2dL^2}{P}\right).$$

It follows that if for more than a $5/46$ fraction of the indices $j < j^*$ we had that the sign of x^*_j agreed with the sign of A^{z^*,k^*}_{j,j^*}, then for large enough d,

$$\|A^{z^*,k^*}x^* - b^{z^*,k^*}\|^2 \geq \frac{5Q}{46}\left(1 - \frac{2dL^2}{P}\right) \geq \frac{5Q}{47}.$$

Now, with probability at least $1 - 49/d$, we have $j^* > 48$, and in this case

$$\|A^{z^*,k^*}x^* - b^{z^*,k^*}\|^2 \geq \frac{5Q}{47} > 100^{z^*}\frac{5j^*}{48},$$

which is a contradiction. The lemma now follows. □

Bob lets x^* be the output of Alg and outputs $-\operatorname{sign}(x^*_{i^*})$. Since i^* is random subject to $i^* < j^*$, the previous lemma ensures that for sufficiently large d, Bob's correctness probability is at least $\frac{41}{46} - \frac{49}{d} \geq \frac{8}{9}$, given $\mathcal{E}$. By a union bound, Alice and Bob solve the AIND problem with probability at least $\frac{8}{9} - \frac{2}{9} \geq \frac{2}{3}$, and so the space complexity of Alg must be $\Omega(d^2(\log(nd))/\varepsilon)$. □

Theorem 3.14. *Suppose $n \geq d/(36\epsilon)$. Consider the Linear Regression problem in which the entries of A and b are inserted exactly once in the data stream. Then any randomized 1-pass algorithm which solves this problem with probability at least 7/9 needs $\Omega(d^2(1/\epsilon + \log(nd)))$ bits of space.*

Proof. We first show that any randomized 1-pass algorithm which solves this problem with probability at least 7/9 needs $\Omega(d^2/\epsilon)$ bits of space.

The proof is implicit in the proof of Theorem 3.7, which now just requires a reduction from IND, i.e., in the proof there we only consider matrices $A^{z,k}$ for which $z = 0$, so Bob does not need to delete any entries of A. The squared error of the linear regression problem is now simply $\sum_{k=1}^{u} \|A^{0,k}x - b^{0,k}\|^2$. Lemma 3.8 continues to provide an upper bound on $\|Ax - b\|^2$ (here, $z^* = 0$). We define $\mathcal{E}$ the same as before, and observe that Lemma 3.12 continues to hold, since the proof only considers properties of $A^{z^*,k}$, $k \in [u]$. Moreover, Lemma 3.13 continues to hold, since only properties of $A^{z^*,k}$, $k \in [u]$, were used in the proof. Thus, Bob simply outputs $-\operatorname{sign}(x^*_{i^*})$ as before, and the proof follows. We omit further details for this part.

We now show that any randomized 1-pass algorithm which solves this problem with probability at least 2/3 needs $\Omega(d^2 \log(nd))$ bits of space.

We reduce from $AIND$ on strings of length $d(d-1)(\log(nd))/2$. Alice interprets her input string as a $d \times d$ upper-triangular matrix A with integer entries above the diagonal in the set $[nd]$. She sets the diagonal entries of A to be 1. Notice that A has full rank, and so for any $d \times 1$ vector b, $\min_{x \in \mathbb{R}^d} \|Ax - b\| = 0$, since $x = A^{-1}b$.

Bob is given an index in the AIND problem, which corresponds to a bit in an entry $A_{i,j}$ with $i < j$. By the definition of the AIND problem, we can assume that he is also given $A_{i',j}$ for all $i' > i$ and all j. Bob creates the $d \times 1$ column vector b as follows. Bob sets $b_{j'} = 0$ for all $j' > j$. He sets $b_j = 1$. For each index $i' \in \{i+1, \ldots, j-1\}$, he sets $b_{i'} = A_{i',j}$. Finally he sets $b_{i''} = 0$ for all $i'' \in \{1, \ldots, i\}$.

Consider $x = A^{-1}b$. The constraint $b_{j'} = 0$ for $j' > j$ ensures that $x_{j'} = 0$ for $j' > j$. Since $A_{j,j} = 1$, the constraint $b_j = 1$ ensures that $x_j = 1$. Finally, a simple induction shows that the constraint $b_{i'} = A_{i',j}$ forces $x_{i'} = 0$ for all $i' \in \{i+1, \ldots, j-1\}$. It follows that $(Ax)_i = x_i + A_{i,j}$, and since $b_i = 0$, we have that $x_i = -A_{i,j}$.

Alice feeds the matrix A into an algorithm Alg which solves the linear regression problem with probability at least 2/3, and transmits the state to Bob. Bob feeds b in to Alg. Then $x = A^{-1}b$ is output with probability at least 2/3, and Bob lets $-x_i$ be his guess for $A_{i,j}$. By the arguments above, Bob correctly guesses $A_{i,j}$ with probability at least 2/3. He then outputs the appropriate bit of $-x_i$, thereby solving the AIND problem with probability at least 2/3. It follows that the space complexity of Alg is $\Omega(d^2 \log(nd))$. ∎

4 Low-Rank Approximation

4.1 Upper Bounds

We give several algorithms, trading specificity and passes for space. As mentioned in §1.4 on page 7, we will assume that we can compute some matrices exactly, and then round them for use. We will show that all matrices (up to those exact computations) can be used in rounded form during the streaming phase.

Throughout this section, A is an $n \times d$ input matrix of rank ρ, with entries of size $\gamma = O(\log(nd))$ as $nd \to \infty$. The value k is a given integer, A_k is the best rank-k approximation to A, $\Delta_k := \|A - A_k\|$ is the error of A_k, $\delta > 0$ is the probability of failure, and $\epsilon > 0$ is the given error parameter.

Bit Complexity To prove space and error bounds for these algorithms, we will show that the numerical error is small enough, using $O(\log(nd))$-bit entries, assuming the input matrix also has entries of $O(\log(nd))$ bits. We will assume that, between passes, we may do exact computations, and then round the results to use during and after the next pass. A key property here is that the singular values of an integer matrix with bounded entries cannot be too small or too large. We note that this lemma is also proven and used in [FMSW09], though there it is used for measures other than the Frobenius norm.

Lemma 4.1. *If $n \times d$ matrix A has integer entries bounded in magnitude by γ, and has rank $\rho \geq 2k$, then the k'th singular value σ_k of A has $|\log \sigma_k| = O(\log(nd\gamma))$ as $nd \to \infty$. This implies that $\|A\|/\Delta_k \leq (nd\gamma)^{O(1)}$ as $nd \to \infty$.*

Proof. The characteristic polynomial of $A^T A$ is $p(\lambda) := \det(\lambda I - A^T A)$, and since $p(\lambda) = \lambda^{d-\rho} \prod_{1 \leq i \leq \rho}(\lambda - \lambda_i)$, the coefficient of $\lambda_{d-\rho}$ in $p(\lambda)$ is $\prod_{1 \leq i \leq \rho} \lambda_i$. Since $A^T A$ has integer entries, the coefficients of $p(\lambda)$ are also integers, and since the eigenvalues of $A^T A$ are nonnegative, $\prod_{1 \leq i \leq \rho} \lambda_i \geq 1$. Since the eigenvalues λ_k of $A^T A$ have $\lambda_k = \sigma_k^2$, we have $\prod_{1 \leq i \leq \rho} \sigma_i \geq 1$.

We have also

$$\sum_{1 \leq i \leq \rho} \lambda_i = \sum_{1 \leq i \leq \rho} \sigma_i^2 = \|A\|^2 \leq nd\gamma^2,$$

and so $\sigma_i \leq nd\gamma^2$ for all i. Thus

$$\lambda_k^{\rho-k} \geq \prod_{k < i \leq \rho} \lambda_i \geq \prod_{1 \leq i \leq \rho} \lambda_i/(nd\gamma^2)^k \geq (nd\gamma^2)^{-k},$$

and so

$$\lambda_k \geq (nd\gamma^2)^{-k/(\rho-k)} \geq (nd\gamma^2)^{-1}. \tag{10}$$

The main statement of the lemma follows upon taking logarithms of the upper and lower bounds for the λ_i, which imply the same asymptotic bounds for σ_i. The last statement follows using the bound for $\|A\|$, and (10), which implies a lower bound for Δ_k. □

The analysis of the low-rank approximation algorithms is based on the application of Theorem 3.1 on page 14, as in the following theorem; again, the proof technique is similar to that of [Sar06, DMMS07].

Theorem 4.2. *There is an $m = O(k \log(1/\delta)/\epsilon)$ such that, if S is an $n \times m$ sign matrix, then with probability at least $1-\delta$, there is an $n \times m$ matrix Y of rank at most k, so that*

$$\|YS^TA - A\| \leq (1+\epsilon)\Delta_k.$$

Similarly, for a $d \times m$ sign matrix R, with probability at least $1-\delta$ there is an $m \times d$ matrix Z so that

$$\|ARZ - A\| \leq (1+\epsilon)\Delta_k.$$

The entries of S and R need be at most $\eta(k + \log(1/\delta))$-wise independent, for a constant η.

The theorem says that the rowspace of S^TA contains a very good rank-k approximation to A, and similarly for the columnspace of AR.

Proof. For the claims about Y, apply Theorem 3.1 on page 14, with A and B equal to the A_k and A of Theorem 4.2. Then $A_k\tilde{X} = A_k(S^TA_k)^-S^TA$ satisfies $\|A_k\tilde{X} - A\| \leq (1+\epsilon)\|A_kX^* - A\|$, and since A_k is the best rank-k approximation to A, $\|A_kX^* - A\| = \|A_k - A\| = \Delta_k$. Thus if $Y := A_k(S^TA_k)^-$, $YS^TA = A_kX$ satisfies the claimed inequality. Since A_k has rank at most k, this is true for Y also, and so the claims for Y follow.

The claims involving R follow by applying the result to A^T. ∎

The following lemma will be helpful.

Lemma 4.3. *Given a matrix A and matrix U with orthonormal columns, both with the same number of rows, the best rank-k approximation to A in the columnspace of U is given by $U[U^TA]_k$, where $[U^TA]_k$ is the best rank-k approximation to U^TA. A similar claim applies for G a matrix with orthonormal rows, and the best rank-k approximation to A in the rowspace of G.*

Proof. The matrix UU^TA is the projection of A onto the columnspace of U, and since, for any conforming Y,

$$(A - UU^TA)^T(UU^TA - UY) = A^T(I - UU^T)U(U^TA - Y) = 0,$$

by the Pythagorean Theorem, we have

$$\|A - UY\|^2 = \|A - UU^TA\|^2 + \|UU^TA - UY\|^2.$$

If Z has rank no more than k, then using $\|Ux\| = \|x\|$ for any conforming x,

$$\|UU^TA - U[U^TA]_k\| = \|U^TA - [U^TA]_k\| \leq \|U^TA - Z\| = \|UU^TA - UZ\|.$$

Hence

$$\begin{aligned} \|A - U[U^TA]_k\|^2 &= \|A - UU^TA\|^2 + \|UU^TA - U[U^TA]_k\|^2 \\ &\leq \|A - UU^TA\|^2 + \|UU^TA - UZ\|^2 \\ &= \|A - UZ\|^2. \end{aligned}$$

The main claim of the lemma follows upon taking square roots. For the last claim, apply the columnspace result to A^T and U^T. ∎

4.1.1 Two passes

The most direct application of the above theorem and lemma yields a two pass algorithm, as follows. In the first pass, accumulate $S^T A$. Before the second pass, compute a matrix G whose rows are an orthonormal basis for the rowspace of $S^T A$. In the second pass, accumulate the coefficients AG^T of the projection $AG^T G = A(S^T A)^- S^T A$ of A onto the row space of $S^T A$. Finally, compute the best rank-k approximation $[AG^T]_k$ to AG^T, and return $\tilde{A}_k = [AG^T]_k G$. As proven below, this approximation is close to A.

Although this discussion assumes that G is computed exactly, we will show that an approximation $\hat{G}$ can be used: for an appropriate κ in $O(\log(nd))$, $\hat{G}$ is $2^{-\kappa}\lfloor 2^{\kappa} G\rfloor$, stored implicitly as a scaling factor and an integer matrix. (Here $\lfloor\,\rfloor$ denotes the floor function applied entrywise.)

Theorem 4.4. *If the rank $\rho \geq 2(k+1)$, then there is an $m = O(k\log(1/\delta)/\epsilon)$ such that, if S is an $n \times m$ sign matrix, then with probability at least $1-\delta$, the rank-k matrix $\tilde{A}_k$, as returned by the above two-pass algorithm, satisfies*

$$\|A - \tilde{A}_k\| \leq (1+\epsilon)\Delta_k.$$

An approximation $\hat{G}$ to G with $O(\log(nd))$-bit entries may be used, with the same asymptotic bounds. The space used is

$$O(m(n+d))\log(nd) = O(k\epsilon^{-1}(n+d))\log(1/\delta)\log(nd).$$

By Theorem 4.14 on page 39, the space used is optimal for fixed ϵ (and δ).

Proof. Let G be a matrix whose rows are an orthonormal basis for the rowspace of $S^T A$. From Lemma 4.3 on the preceding page, $\tilde{A}_k = [AG^T]_k G$ is the best rank-k approximation to A in the rowspace of G. Since $A_k(S^T A)^- S^T A$ has rank k, and is in the rowspace of G, its Frobenius distance to A must be larger than that of $\tilde{A}_k$. By the theorem just above, that distance is no more than $(1+\epsilon)\Delta_k$, and the theorem follows, up to the numerical approximation of G by $\hat{G}$.

For bounded precision, use $\hat{G}$ as defined above, in place of G, returning $\hat{A}_k = [A\hat{G}^T]_k \hat{G}$, where $[A\hat{G}^T]_k$ is the best rank-k approximation to $A\hat{G}^T$. First we note that

$$\|\hat{A}_k - [A\hat{G}^T]_k G\| = \|[A\hat{G}^T]_k E\|, \tag{11}$$

where $E := G - \hat{G}$. Moreover,

$$\begin{aligned}
& \sqrt{\|A - [A\hat{G}^T]_k G\|^2 - \|A - AG^T G\|^2} \\
\text{(Pyth. Thm.)} \quad & = \|AG^T G - [A\hat{G}^T]_k G\| \\
\text{(triangle ineq.)} \quad & \leq \|AG^T G - A\hat{G}^T G\| + \|A\hat{G}^T G - [A\hat{G}^T]_k G\| \\
\text{(Lemma 4.3)} \quad & \leq \|AE^T G\| + \|A\hat{G}^T G - \tilde{A}_k\| \\
\text{(triangle ineq.)} \quad & \leq \|AE^T G\| + \|A\hat{G}^T G - AG^T G\| + \|AG^T G - \tilde{A}_k\| \\
& = 2\|AE^T G\| + \|AG^T G - \tilde{A}_k\|.
\end{aligned}$$

By the Pythagorean Theorem,

$$\|A - \tilde{A}_k\|^2 = \|A - AG^T G\|^2 + \|AG^T G - \tilde{A}_k\|^2, \tag{12}$$

and rearranging as a bound on $\|A - [A\hat{G}^T]_k G\|^2$, we have

$$\begin{aligned}
&\|A - [A\hat{G}^T]_k G\|^2 \\
&\quad \le \|A - AG^T G\|^2 + (2\|AE^T G\| + \|AG^T G - \tilde{A}_k\|)^2 \\
&\quad = \|A - \tilde{A}_k\|^2 + +4\|AE^T G\|\|AG^T G - \tilde{A}_k\| + 4\|AE^T G\|^2.
\end{aligned}$$

We've already shown that $\|A - \tilde{A}_k\|^2 \le (1+\epsilon)^2\Delta_k^2$, and this implies with (12) that $\|AG^T G - \tilde{A}_k\| \le (1+\epsilon)\Delta_k$. Thus

$$\begin{aligned}
&\|A - [A\hat{G}^T]_k G\|^2 \\
&\quad \le (1+\epsilon)^2\Delta_k^2 + 4\|AE^T G\|(1+\epsilon)\Delta_k + 4\|AE^T G\|^2.
\end{aligned}$$

We have

$$\|AE^T G\| = \|AE^T\| \le \|A\|\|E\| \le 2^{-\kappa}\sqrt{md}\|A\|,$$

and with the assumption that the rank ρ of A has $\rho \ge 2(k+1)$, and Lemma 4.1 on page 25, for κ in $O(\log(nd))$, $\|AE^T G\| \le \epsilon\Delta_k$. We have

$$\|A - [A\hat{G}^T]_k G\| \le \Delta_k\sqrt{1 + 6\epsilon + 9\epsilon^2} = \Delta_k(1+3\epsilon),$$

and with (11) on the previous page,

$$\begin{aligned}
\|A - \hat{A}_k\| &\le \|A - [A\hat{G}^T]_k G\| + \|\hat{A}_k - [A\hat{G}^T]_k G\| \\
&\le (1+3\epsilon)\Delta_k + \|[A\hat{G}^T]_k E\| \\
&\le (1+4\epsilon)\Delta_k,
\end{aligned}$$

using a similar bound for $\|[A\hat{G}^T]_k E\|$ as for $\|AE^T G\|$. The results follows after adjusting constants. ☐

4.1.2 One pass for Column-wise Updates

If A is given a column at a time, or a row at a time, then an efficient streaming algorithm is possible. By Theorem 4.10 on page 33, for n within a constant factor of d, the space used by this algorithm is within a factor of $\log(nd)$ of optimal.

Theorem 4.5. *Suppose input A is given as a sequence of columns or rows. There is an $m = O(k\log(1/\delta)/\epsilon)$, such that with probability at least $1-\delta$, a matrix $\tilde{A}_k$ can be obtained that satisfies*

$$\|\tilde{A}_k - A\| \le (1+\epsilon)\Delta_k.$$

The space needed is

$$O((n+d)m) = O(k\epsilon^{-1}(n+d)\log(1/\delta)\log(nd)).$$

The update time is amortized $O(m)$ per entry.

Proof. We can assume that A is given column-wise, since the bounds are symmetric in n and d. The algorithm maintains the sketch $S^T A$, where S is an $n \times m$ sign matrix, and $m = O(k \log(1/\delta)/\epsilon)$. It also maintains $S^T AA^T$; since $AA^T = \sum_j a_{:j} a_{:j}^T$, when a column $a_{:j}$ arrives, the matrix $S^T a_{:j} a_{:j}^T$ can be computed in $O(mn)$ time and added to the current version of $S^T AA^T$. Since the pseudo-inverse of $S^T A$ can be expressed as $A^T S(S^T AA^T S)^-$, the projection of A onto the rowspace of $S^T A$ is

$$\begin{aligned} A(S^T A)^- S^T A &= AA^T S(S^T AA^T S)^- S^T A \\ &= (S^T AA^T)^T (S^T A(S^T A)^T)^- S^T A. \end{aligned}$$

That is, the matrices $S^T A$ and $S^T AA^T$ are enough to compute the projection of A onto the rowspace of $S^T A$. Using Theorem 4.2 and Lemma 4.3, the best rank-k approximation $\tilde{A}_k$ to $A(S^T A)^- S^T A$ in the rowspace of $S^T A$ satisfies the conditions of the theorem. □

4.1.3 Three passes for Row-wise Updates, With Small Space

We show the following.

Theorem 4.6. *Suppose A is given row-wise. There is $m = O(k \log(1/\delta)/\epsilon)$ such that, a matrix $\tilde{A}_k$ can be found in three passes so that with probability at least $1 - \delta$,*

$$\|\tilde{A}_k - A\| \leq (1 + \epsilon)\Delta_k.$$

The algorithm uses space

$$O(k(n + d \log(1/\delta)/\epsilon + k \log(1/\delta)^2/\epsilon^2) \log(nd)).$$

A comparable approach, without sketching, would use $\Theta((nk + d^2) \log(nd))$ space over two passes, so this result becomes interesting when $k < \epsilon d$. As mentioned in the introduction, for n larger than a constant times $\max\{d/\epsilon, k/\epsilon^2\} \log(1/\delta)$, the space bound is $O(nk \log(nd))$, which is comparable to our lower bound Theorem 4.14 on page 39, showing that $\Omega((n + d)k \log(nd))$ bits are needed even with multiple passes and randomization.

Proof. The algorithm is described assuming that exact arithmetic can be used; the analysis discusses the changes needed to allow finite precision entries.

1. In the first pass, accumulate $S^T A$, where S is an $n \times m$ sign matrix, with m a large enough value in $O(k\epsilon^{-1} \log(1/\delta))$;
   - Before the second pass, compute an orthonormal matrix G whose rows are a basis for the rowspace of $S^T A$;
2. In the second pass, for each update row a, compute $c := aG^T$, and accumulate $C^T C \mathrel{+}= c^T c$;

- That is, C is the matrix AG^T. Before the third pass, compute the SVD of C^TC, which is $V\Sigma^2V^T$ when the SVD of C is $U\Sigma V^T$. Let V_k denote the leftmost k columns of V, and Σ_k denote the $k \times k$ diagonal matrix of the largest singular values of C.

3. In the third pass, compute $c := aG^T$ for the current row a of A, then compute and store $cV_k\Sigma_k^-$, building the matrix $CV_k\Sigma_k^-$. The latter is in fact U_k, the matrix of the top k left singular vectors of C, so that $U_k\Sigma_kV_k$ is the best rank-k approximation to $C = AG^T$. Return $\tilde{A}_k := U_k\Sigma_kV_kG$.

We first discuss the algorithm under the assumption that G and V_k can be used exactly, and then consider the effects of rounding.

The algorithm constructs the projection of A onto the rowspace of S^TA, and computes the best rank-k approximation to that projection. As in the previous algorithms, the quality bound follows from Theorem 4.2 on page 25 and Lemma 4.3 on page 26. Note that the entries of S need only be $O(k+\log(1/\delta))$-wise independent, so the space needed for S is negligible.

The space required is $O(dm)$ entries for storing S^TA and G, and then $O(m^2)$ entries for storing C^TC, and finally $O(nk)$ entries for storing $CV_k\Sigma_k^-$. Thus the total space is

$$O(nk + dm + m^2) = kO(n + d\log(1/\delta)/\epsilon + k\log(1/\delta)/\epsilon^2).$$

To use matrix entries with $O(\log(nd))$ bits, as in the two pass algorithm we use a matrix $\hat{G} := 2^{-\kappa}\lfloor 2^\kappa G\rfloor$, for large enough κ in $O(\log(nd))$, and also use $\hat{V}_k := 2^{-\kappa}\lfloor 2^\kappa V_k\rfloor$, where V_k is from the exact SVD of $\hat{C}^T\hat{C}$, where $\hat{C} = A\hat{G}^T$.

We also change the algorithm so that in the third pass, it is $\hat{C}\hat{V}^T$ that is maintained. Then, after the third pass, the exact SVD of $\hat{C}^T\hat{C}$ is computed again, obtaining Σ_k and yielding $\hat{C}\hat{V}_k\Sigma_k^-$ as an estimate of U_k. We have

$$\hat{C}\hat{V}_k\Sigma_k^- = \hat{C}(V_k + E)\Sigma_k^- = U_k\Sigma_kV_k + A\hat{G}^TE\Sigma_k^-,$$

where $E := \hat{V}_k - V_k$. For sufficiently large κ in $O(\log(nd))$, and using the bounds of Lemma 4.1 on page 25 on Σ, we have $\|A\hat{G}^TE\Sigma_k^-\| \leq \epsilon\Delta_k$. Together with the above analysis for the unrounded versions, the error bound follows, using the triangle inequality and adjusting constant factors. □

4.1.4 One pass, and a Bicriteria Approximation

To obtain a low-rank approximation even for turnstile updates, we will need more space. First, we can apply Theorem 3.1 on page 14 twice to obtain a bicriteria low-rank approximation. As mentioned in the introduction, the construction is somewhat akin to the CUR decomposition [DMM08, DKM06].

Theorem 4.7. *There is an $m = O(k\log(1/\delta)/\epsilon)$ such that, if S is an $n\times(m/\epsilon)$ sign matrix, and R is a $d \times m$ sign matrix, then with probability at least $1-\delta$,*

$$\|A - \tilde{A}\| \leq (1+\epsilon)\Delta_k,$$

where $\tilde{A} := AR(S^T AR)^- S^T A$. *The entries of* S *need be at most* $\eta(k/\epsilon + \log(1/\delta))$*-wise independent, for a constant* η.

Proof. We apply Theorem 3.1 with k, A, B, and m of the theorem mapping to k/ϵ, AR, A, and m/ϵ, respectively. The result is that for $\tilde{X}$ the solution to

$$\min_X \|S^T ARX - S^T A\|,$$

we have

$$\|AR\tilde{X} - A\| \leq (1+\epsilon)\|ARX^* - A\| = (1+\epsilon)\min_X \|ARX - A\|,$$

and applying Theorem 3.1 again, with k, A, B, and m of the theorem mapping to m, A_k, A, and m, we have, with probability at least $1-\delta$,

$$\|ARX^* - A\| \leq (1+\epsilon)\|A - A_k\| = (1+\epsilon)\Delta_k. \tag{13}$$

Since $\tilde{X} = (S^T AR)^- S^T A$, we have

$$\begin{aligned}\|AR(S^T AR)^- S^T A - A\| &= \|AR\tilde{X} - A\| \\ &\leq (1+\epsilon)\|ARX^* - A\| \\ &\leq (1+\epsilon)^2 \Delta_k,\end{aligned}$$

and the theorem follows, after adjusting δ and ϵ by constant factors. ∎

Note that by computing the SVD $\tilde{U}\tilde{\Sigma}\tilde{V}^T$ of $(S^T AR)^-$, we obtain a low-rank approximation to A of the form

$$AR\tilde{U}\tilde{\Sigma}\tilde{V}^T S^T A,$$

which is of the same form as an SVD. While this decomposition has rank $O(k\epsilon^{-1}\log(1/\delta))$, and is guaranteed to approximate A only nearly as well as the best rank-k approximation, it would be much quicker to compute, and potentially could be substituted for the SVD in many applications.

A rank-k approximation is similarly obtainable, as follows.

Theorem 4.8. *Under the conditions of the previous theorem, let* U *be an orthonormal basis for the columnspace of* AR. *Then the best rank-k approximation* $U[U^T\tilde{A}]_k$ *to* $\tilde{A}$ *in the columnspace of* U *satisfies*

$$\|A - U[U^T\tilde{A}]_k\| \leq (1+\sqrt{\epsilon})\Delta_k.$$

For convenience of reference, we state a result giving a quality bound of the usual form, simply using a different ϵ.

Theorem 4.9. *There is an* $m = O(k\log(1/\delta)/\epsilon^2)$ *such that, if* S *is an* $n \times (m/\epsilon^2)$ *sign matrix, and* R *is a* $d \times m$ *sign matrix, the following succeeds with probability* $1-\delta$. *Let* U *be an orthonormal basis for the columnspace of* AR.

Then the best rank-k approximation $U[U^T\tilde{A}]_k$ to $\tilde{A} := AR(S^TAR)^-S^TA$ in the columnspace of U satisfies

$$\|A - U[U^T\tilde{A}]_k\| \le (1+\epsilon)\Delta_k.$$

The entries of S need be at most $\eta(k/\epsilon + \log(1/\delta))$-wise independent, for a constant η.

Proof. (of Theorem 4.8.) For any such U, there is a matrix Y so that $UY = AR$; in particular, we will take U to be the matrix of left singular vectors of AR, so that the corresponding Y is ΣV^T.

Consider the projections UU^TA and UU^TA_k of A and A_k to the columnspace of AR, as well as $\tilde{A}$, which is already in the columnspace of AR. We first obtain distance bounds involving these projections, by applying Lemma 3.3 on page 15 used in the proof of Theorem 3.1 on page 14; this bound is used twice, first in the setting of the first application of Theorem 3.1, and then in the setting of the second application.

The projection UU^TA can also be expressed as ARX^*, with X^* as in the first application of Theorem 3.1, and $\tilde{A}$ then equal to the corresponding $AR\tilde{X}$. From Lemma 3.3 on page 15 and (13) on the previous page,

$$\begin{aligned}\|UU^TA - \tilde{A}\| &= \|AR(X^* - \tilde{X})\| \\ &\le 2\sqrt{\epsilon}\|A - ARX^*\| \\ &\le 2\sqrt{\epsilon}(1+\epsilon)\Delta_k. \qquad (14)\end{aligned}$$

Since the projection UU_TA_k is the closest matrix in the columnspace of AR to A_k, and again from Lemma 3.3, as in the second application above, we have

$$\|UU^TA_k - A_k\| \le \|AR(A_kR)^-A_k - A_k\| \le 2\sqrt{\epsilon}\Delta_k. \qquad (15)$$

Also, since $[U^T\tilde{A}]_k$ is the closest rank-k matrix to $U^T\tilde{A} = Y(S^TAR)^-S^TA$, U^TA_k must be no closer to $U^T\tilde{A}$, and so

$$\begin{aligned}\|\tilde{A} - U[U^T\tilde{A}]_k\| &\le \|\tilde{A} - UU^TA_k\| \\ \text{(triangle ineq.)} \quad &\le \|\tilde{A} - UU^TA\| + \|UU^TA - UU^TA_k\| \\ \text{(By (14))} \quad &\le 2\sqrt{\epsilon}(1+\epsilon)\Delta_k + \|UU^T(A - A_k)\|. \qquad (16)\end{aligned}$$

Since $\|UU^TZ\| \le \|Z\|$ for any Z, we have

$$\|UU^TA - UU^TA_k\| \le \Delta_k. \qquad (17)$$

Since $(A - UU^TA)^TU = 0$, we have

$$\begin{aligned}&\|A - U[U^T\tilde{A}]_k\|^2 - \|A - UU^TA\|^2 \\ \text{(Pyth. Thm.)} \quad &= \|UU^TA - U[U^T\tilde{A}]_k\|^2 \\ \text{(triangle ineq.)} \quad &\le (\|UU^TA - \tilde{A}\| + \|\tilde{A} - U[U^T\tilde{A}]_k\|)^2 \\ \text{(By (14),(16))} \quad &\le (2\sqrt{\epsilon}(1+\epsilon)\Delta_k + [2\sqrt{\epsilon}(1+\epsilon)\Delta_k \\ &\quad + \|UU^T(A - A_k)\|])^2 \\ &= (4\sqrt{\epsilon}(1+\epsilon)\Delta_k + \|UU^T(A - A_k)\|])^2.\end{aligned}$$

Expanding the square and rearranging,

$$\begin{aligned}
&\|A - U[U^T\tilde{A}]_k\|^2 - \|A - UU^TA\|^2 - \|UU^TA - UU^TA_k\|^2 \\
&\quad \le 8\sqrt{\epsilon}(1+\epsilon)\Delta_k\|UU^TA - UU^TA_k\| \\
&\qquad + 16\epsilon(1+\epsilon)^2\Delta_k^2 \\
\text{(By (17))} &\quad \le 8\sqrt{\epsilon}(1+\epsilon)\Delta_k + 16\epsilon(1+\epsilon)^2\Delta_k^2 \\
&\quad = 8\Delta_k^2(1+\epsilon)(\sqrt{\epsilon} + 2\epsilon(1+\epsilon)).
\end{aligned}$$

Rearranging this bound,

$$\begin{aligned}
\|A - U[U^T\tilde{A}]_k\|^2 &- 8\Delta_k^2(1+\epsilon)(\sqrt{\epsilon} + 2\epsilon(1+\epsilon)) \\
&\le \|A - UU^TA\|^2 + \|UU^TA - UU^TA_k\|^2 \\
\text{(Pyth. Thm.)} \quad &= \|A - UU^TA_k\|^2 \\
\text{(triangle ineq.)} \quad &\le (\|A - A_k\| + \|A_k - UU^TA_k\|)^2 \\
\text{(By (15))} \quad &\le (\Delta_k + 2\sqrt{\epsilon}\Delta_k)^2,
\end{aligned}$$

which implies

$$\begin{aligned}
\|A - U[U^T\tilde{A}]_k\|^2 &\le 8\Delta_k^2(1+\epsilon)(\sqrt{\epsilon} + 2\epsilon(1+\epsilon)) \\
&\quad + (\Delta_k + 2\sqrt{\epsilon}\Delta_k)^2 \\
&\le \Delta_k^2(1 + 12\sqrt{\epsilon} + O(\epsilon)),
\end{aligned}$$

and the theorem follows upon taking square roots, and adjusting ϵ by a constant factor. □

4.2 Lower Bounds for Low-Rank Approximation

The next theorem shows that our 1-pass algorithm receiving entries in row or column order uses close to the best possible space of any streaming algorithm.

Theorem 4.10. *Let $\epsilon > 0$ and $k \ge 1$ be arbitrary.*

- *Suppose $d > \beta k/\epsilon$ for an absolute constant $\beta > 0$. Then any randomized 1-pass algorithm which solves the Rank-k Approximation Problem with probability at least 5/6, and which receives the entries of A in row-order, must use $\Omega(nk/\epsilon)$ bits of space.*
- *Suppose $n > \beta k/\epsilon$ for an absolute constant $\beta > 0$. Then any randomized 1-pass algorithm which solves the Rank-k Approximation Problem with probability at least 5/6, and which receives the entries of A in column-order must use $\Omega(dk/\epsilon)$ bits of space.*

Proof. We will prove the theorem when the algorithm receives entries in row-order. The proof for the case when the algorithm receives entries in column-order is analogous.

Choosing the inputs for the hard instance:

Let $a = k/(21\epsilon)$, which, for large enough $\beta > 0$, can be assumed to be at most $d/2$. We reduce from IND on strings x of length $(n-a)a$. We use the distributional version of IND, for which x and Bob's inputs $i \in [(n-a)a]$ and are independent and generated uniformly at random. Alice interprets her input x as a string in $\{-1,1\}^{(n-a)a}$. Let Z_1 be the $a \times a$ zeros matrix, Z_2 the $a \times (d-a)$ zeros matrix, Z_3 the $(n-a) \times (d-a)$ zeros matrix, and M an $(n-a) \times a$ matrix to be constructed. Define the $n \times d$ block matrix:

$$A' = \begin{pmatrix} Z_1 & Z_2 \\ M & Z_3 \end{pmatrix}$$

Alice constructs the $(n-a) \times a$ matrix M as follows simply by associating each entry of x with a unique entry of M.

Alice runs a randomized 1-pass streaming algorithm for Rank-k Approximation on the last $n-a$ rows of A', and transmits the state of the algorithm to Bob. Bob partitions the first a columns of A' into a/k contiguous groups of size k. Given $i \in [(n-a)a]$, suppose it is associated with an entry in the group G containing columns $s, s+1, \ldots, s+k-1$. Bob then feeds the $(s+j)$-th rows $P \cdot e_{s+j}$ to the stream, where P is a large positive integer to be determined, e_{s+j} is the standard unit vector in direction $s+j$, and where j ranges from 0 to $k-1$. Bob also feeds the remaining rows of A' as zero rows to the algorithm.

Note that each entry occurs once, and the streaming algorithm can be assumed to see entire rows at a time. Moreover, the entries can each be described with $O(\log(nd))$ bits, assuming that P can (P will turn out to be $2n^4$). Let A denote the resulting underlying matrix in the stream. We partition the first a columns of A into contiguous submatrices $A^1, \ldots, A^{a/k}$ each containing k columns of A. Suppose G is associated with the submatrix A^r.

The plan is to show that the columns in $\tilde{A}_k$ corresponding to the columns in A^r must be linearly independent if the error of the streaming algorithm is small. This is done in Lemma 4.11. This enables us to express the error $\|A - \tilde{A}_k\|^2$ solely in terms of linear combinations of these k columns. We bound the coefficients of these combinations in Lemma 4.12. This allows us to show that most of the error of the streaming algorithm is actually on columns other than the k columns in A^r. Hence, most of the signs of entries in $\tilde{A}_k$ in these k columns must agree with the signs of the corresponding entries in A^r, as otherwise the error on these k columns would be too large. It follows that Bob can recover the sign of his entry with reasonable probability, and hence solve the IND problem.

Properties of a near-optimal solution:

Notice that columns of A^r have length $\left(P^2 + n - a\right)^{1/2}$, whereas the remaining columns among the first a columns have length $(n-a)^{1/2}$. The last $n-a$ columns of A have length 0. It follows that $\|A - A_k\|^2 \le (a-k)(n-a)$. Hence, to solve the Rank-k Approximation Problem, a matrix $\tilde{A}_k$ must be output with

$$\|\tilde{A}_k - A\|^2 \le (1+\epsilon)^2 \|A_k - A\|^2 \le (1+3\epsilon)(a-k)(n-a) \le (a-k)(n-a) + \frac{k(n-a)}{7},$$

where we have used the definition of a. Let $\mathcal{E}$ be the event that the algorithm succeeds in outputting such an $\tilde{A}_k$, so $\Pr[\mathcal{E}] \ge 5/6$. We let $\tilde{A}_k^1, \ldots, \tilde{A}_k^{a/k}$ denote the contiguous submatrices of $\tilde{A}_k$, each containing k columns of $\tilde{A}_k$. Suppose G is associated with the submatrix $\tilde{A}_k^r$.

Lemma 4.11. *For $P \ge 3n^2$, the columns of $\tilde{A}_k^r$ are linearly independent.*

Proof. We will show that if the columns of $\tilde{A}_k^r$ were not linearly independent, then the event $\mathcal{E}$ could not occur. In fact, we shall show that the restrictions of these k columns to rows $r(a/k)+1, \ldots, (r+1)(a/k)$, are linearly independent. Let the restrictions of these k columns to these k rows be denoted $w^1, \ldots, w^k$, and suppose these vectors were not linearly independent. Then there is an index $i_0 \in [k]$ and real numbers α_i for each $i \in T := [k] \setminus \{i_0\}$ for which $w^{i_0} = \sum_{i \in T} \alpha_i w^i$. Now,

$$\sum_{i=1}^{k} (w_i^i - P)^2 \le \|\tilde{A}_k - A\|^2 \le (a-k)(n-a) + \frac{k(n-a)}{7} \le 2n^2.$$

By choosing $P \ge 3n^2$, for all i, $w_i^i \ge n^2$. Moreover,

$$\sum_{i=1}^{k} \sum_{j \ne i} (w_j^i)^2 \le \|\tilde{A}_k - A\|^2 \le (a-k)(n-a) + \frac{k(n-a)}{7} \le 2n^2,$$

and so we can assume $|w_j^i| \le \sqrt{2}n$ for all $i \ne j$. Now we have the relation

$$w_{i_0}^{i_0} = \sum_{i \in T} \alpha_i w_{i_0}^i.$$

For this relation to hold, there must be some $i \in T$ for which

$$|\alpha_i| \ge n^2/(k\sqrt{2}n) = n/(\sqrt{2}k).$$

Let i^* be the index in T for which $|\alpha_{i^*}|$ is largest. We also have the relation

$$w_{i^*}^{i_0} = \sum_{i \in T} \alpha_i w_{i^*}^i.$$

Now, $|\alpha_{i^*} w_{i^*}^{i^*}| \ge |\alpha_{i^*}| n^2$ and $|\alpha_i w_{i^*}^i| \le |\alpha_{i^*}| \sqrt{2}n$ for all $i \in T$. It follows that

$$\left| \sum_{i \in T} \alpha_i w_{i^*}^i \right| \ge |\alpha_{i^*}|(n^2 - k\sqrt{2}n) \ge \frac{n}{\sqrt{2}k} \cdot \left(\frac{n^2}{2} \right) > \sqrt{2}n,$$

where the last inequality holds if $k < n/(2\sqrt{2})$, which holds for large enough $\beta > 0$. This contradicts $|\sum_{i \in T} \alpha_i w_{i^*}^i| = |w_{i^*}^{i_0}| \le \sqrt{2}n$. □

The previous lemma implies that all columns of $\tilde{A}_k$ are in the span of the columns of $\tilde{A}_k^r$. Denote these columns $v^1, \ldots, v^k$, and let the restriction of these columns to rows $a+1, \ldots, d$ be denoted $\tilde{v}^1, \ldots, \tilde{v}^k$. We may assume the vectors v^i have entries 1 through $r(a/k)-1$ set to 0, as well as entries $(r+1)(a/k)+1$ through a set to 0. We may also assume the last $d-a$ columns of $\tilde{A}_k$ are 0. The i-th column of $\tilde{A}_k$, for any $1 \le i \le a$, can thus be written as $\sum_{j=1}^k \beta_{i,j} v^j$. We may assume the vectors v^i have entries 1 through $r(a/k)-1$ set to 0, as well as entries $(r+1)(a/k)+1$ through a set to 0. We may also assume the last $d-a$ columns of $\tilde{A}_k$ are 0. The i-th column of $\tilde{A}_k$, for any $1 \le i \le a$, can thus be written as $\sum_{j=1}^k \beta_{i,j} v^j$.

Let $m^1, \ldots, m^a$ denote the columns of M. Let $S = \{r(a/k)+1, \ldots, (r+1)(a/k)\}$. Then,

$$\begin{aligned}
\|\tilde{A}_k - A\|^2 \quad \geq \quad & \sum_{i=1}^{k} (P - v^i_{r(a/k)+i})^2 + \sum_{i=1}^{k} \sum_{j \neq i} (v^j_{r(a/k)+i})^2 + \sum_{i=1}^{k} \|m^{r(a/k)+i} - \tilde{v}^i\|^2 \\
& + \sum_{i \notin S} \sum_{j=1}^{k} \left(\beta_{i,j} v^j_{r(a/k)+j} + \sum_{j' \neq j} \beta_{i,j'} v^{j'}_{r(a/k)+j} \right)^2 \\
& + \sum_{i \notin S} \|m^i - \sum_{j=1}^{k} \beta_{i,j} \tilde{v}^j\|^2,
\end{aligned}$$

which needs to be at most $(a-k)(n-a) + \frac{k(n-a)}{7} \le 2n^2$.

Lemma 4.12. *For $P \ge 2n^4$, for all $i \notin S$ and all j, $|\beta_{i,j}| \le 2/n^3$.*

Proof. For each $i \neq j$ with $i, j \in [k]$, we have $|v^j_{r(a/k)+i}| \le \sqrt{2}n$, as otherwise the expression above would be larger than $2n^2$. If $P \ge 2n^4$, then each of the $v^i_{r(a/k)+i}$ values is at least n^4, as otherwise the expression above would be at least n^8. For an $i \notin S$, let $j(i)$ be the index for which $|\beta_{i,j(i)}|$ is largest. Then the expression above is at least $|\beta_{i,j(i)}|^2 (n^4 - k\sqrt{2}n)^2$, which is at least $|\beta_{i,j(i)}|^2 n^8/2$ for large enough n. But the expression is at most $2n^2$, and so $|\beta_{i,j(i)}| \le 2/n^3$. The lemma now follows from the definition of $j(i)$. $\square$

Notice that each entry in $\tilde{v}^i$ for each $i \in [k]$ has absolute value at most $\sqrt{2}n+1$, as otherwise $\sum_{i=1}^k \|m^{r(a/k)+i} - \tilde{v}^i\|^2$ would be larger than $2n^2$ (recall that the columns of M are sign vectors), which is a contradiction. From the previous lemma, for $P = 2n^4$ we have $|\beta_{i,j}| \le 2/n^3$ for all $i \notin S$ and all j, and so each entry of $\beta_{i,j}\tilde{v}^j$ has absolute value at most $(\sqrt{2}n+1)2/n^3 \le 4/n^2$. So each entry of $\sum_{j=1}^k \beta_{i,j}\tilde{v}^j$ has absolute value at most $4k/n^2 \le 1/n$. Hence, $\sum_{i \notin S} \|m^i - \sum_{j=1}^k \beta_{i,j}\tilde{v}^j\|^2 \ge \sum_{i \notin S} \|m^i\|^2 (1-1/n)^2$ since the m^i are sign vectors. This is $(a-k)(n-a) - O(a)$. Notice that P only needs $O(\log n)$ bits to describe, so the entries of A are $O(\log n)$-bit integers. Now,

$$\sum_{i=1}^{k} \|m^{r(a/k)+i} - \tilde{v}^i\|^2 = \sum_{i=1}^{k}\sum_{j=1}^{n-a}(m_j^{r(a/k)+i} - \tilde{v}_j^i)^2 \leq \frac{k(n-a)}{7} + O(a).$$

Completing the proof:

Say an entry (j, i) is *useful*, $1 \leq j \leq n-a$ and $1 \leq i \leq k$, if $\text{sign}(m_j^{r(a/k)+i} = \text{sign}(\tilde{v}_j^i)$. An entry that is not useful causes $(m_j^{r(a/k)+i} - \tilde{v}_j^i)^2$ to be at least 1. As there are $k(n-a)$ total entries (j, i), at least a $6/7 - o(1) > 5/6$ fraction are useful.

Bob's index is in one of the columns $m^{r(a/k)+1}, \ldots, m^{r(a/k)+k}$ of M. Since Bob's entry is a random entry in these columns, it follows that by outputting the sign of the corresponding entry in $\tilde{A}_k$, Bob succeeds in solving the IND problem on strings of length $(n-a)a \geq na/2 = \Omega(nk/\epsilon)$, with probability at least $5/6 - 1/6 \geq 2/3$. It follows that any 1-pass randomized algorithm for the Rank-k Approximation Problem receiving entries in row-order with error probability at most $1/6$ uses $\Omega(nk/\epsilon)$ bits of space.

The proof for the case when the algorithm receives entries in column-order works by reducing from the IND problem on strings of length $(d-a)a$, and associating the entries of Alice's input string with the entries of a $a \times (d-a)$ submatrix M. The proof is very similar to this proof. We omit the details. $\square$

We can improve the bound of Theorem 4.10 if we assume the algorithm must work in the general turnstile model.

Theorem 4.13. *Let $\varepsilon > 0$ and $k \geq 1$ be arbitrary. Suppose $\min(n, d) > \beta k \log_{10}(nd)/\varepsilon$ for an absolute constant $\beta > 0$. Then any randomized 1-pass algorithm which solves the Rank-k Approximation Problem with probability at least $5/6$ in the general turnstile model uses $\Omega((n+d)k\log(dn))/\epsilon$ bits of space.*

Proof. We only sketch the differences of the proof of this theorem and the proof of Theorem 4.10. Suppose, first, that $n \geq d$. We now let $a = k(\log_{10}(dn))/(21\varepsilon)$, which for large enough β can be assumed to be at most $d/2$. This time we reduce from the AIND problem on strings x of length $(n-a)a$. As before we define the matrix A' to be:

$$A' = \begin{pmatrix} Z_1 & Z_2 \\ M & Z_3 \end{pmatrix}$$

where now we have

$$M = \begin{pmatrix} M^0 & M^1 & \cdots & M^{\log_{10}(nd)-1} \end{pmatrix}$$

where each M^j is an $(n-a) \times a/(\log_{10}(dn))$ matrix with entries in the set $\{-10^j, 10^j\}$. Each entry of x is associated with exactly one entry in exactly one M^j. If x_i is associated with $M_{a,b}^j$, then $M_{a,b}^j = 10^j$ if $x_i = 1$, and $M_{a,b}^j = -10^j$ if $x_i = 0$. Bob is given an index $i \in [(n-a)a]$. Suppose i is associated with an entry in M^j. By the definition of the AIND problem, we can assume that Bob is also given all entries in $M^{j'}$ for all $j' > j$. Alice runs a randomized 1-pass streaming algorithm for Rank-k Approximation on A', and transmits the state of the algorithm to Bob. Bob then sets all entries in $M^{j'}$ for all $j' > j$ to 0. Let A denote the resulting matrix.

The rest of the proof is very similar to the proof of Theorem 4.10. Bob breaks the columns containing the entries of M^j into $a/(k \log_{10}(dn))$ groups, each of size k. He proceeds by inserting P times a $k \times k$ identity submatrix into A, as in the proof of Theorem 4.10. Again we can show that the k columns in $\tilde{A}_k$ corresponding to the columns for which Bob inserts the value P must be linearly independent. The crucial point is that

$$\begin{aligned} \|A - A_k\|^2 &\le \frac{(a-k)(n-a)10^{2j}}{\log_{10}(dn)} + \sum_{j'<j} \frac{a(n-a)10^{2j'}}{\log_{10}(dn)}. \\ &\le \frac{(a-k)(n-a)100^j}{\log_{10}(dn)} + \frac{a(n-a)}{\log_{10}(dn)} \cdot \frac{100^j}{99}. \end{aligned}$$

So to solve the Rank-k Approximation Problem, using the definition of a, we have that a matrix $\tilde{A}_k$ must be output with

$$\begin{aligned} \|\tilde{A}_k - A\|^2 &\le (1+3\epsilon)\left[\frac{(a-k)(n-a)100^j}{\log_{10}(dn)} + \frac{a(n-a)}{\log_{10}(dn)} \cdot \frac{100^j}{99}\right] \\ &\le \left[\frac{(a-k)(n-a)100^j}{\log_{10}(dn)} + \frac{a(n-a)}{\log_{10}(dn)} \cdot \frac{100^j}{99}\right] \\ &+ \frac{100^j k(n-a)}{7} + \frac{100^j k(n-a)}{99} \\ &\le \left[\frac{(a-k)(n-a)100^j}{\log_{10}(dn)} + \frac{a(n-a)}{\log_{10}(dn)} \cdot \frac{100^j}{99}\right] \\ &+ 100^j k(n-a) \cdot \left(\frac{1}{7} + \frac{1}{99}\right). \end{aligned}$$

By making P sufficiently large, as in the proof of Theorem 4.10, one can show that on all but the k columns of $\tilde{A}_k$ corresponding to columns for which Bob inserts the value P, the error of $\tilde{A}_k$ must be $\left[\frac{(a-k)(n-a)100^j}{\log_{10}(dn)} + \frac{a(n-a)}{\log_{10}(dn)} \cdot \frac{100^j}{99}\right]$, up to low-order terms. In the remaining k columns, if the sign of an entry in $\tilde{A}_k$ corresponding to an entry in M^j disagrees with the sign of the corresponding entry in M^j, then an error of at least 100^j is incurred. It follows that for at least a $1 - 1/7 - 1/99$ fraction of the entries in these k columns, the signs of the entries in $\tilde{A}_k$ agree with the corresponding entries in M^j. Thus, Bob can solve the AIND problem by outputting the sign of the appropriate entry in $\tilde{A}_k$.

This gives a lower bound of $\Omega((n-a)a) = \Omega(nk \log(dn))/\varepsilon$. If instead we had $d \ge n$, a similar argument would have given an $\Omega(dk \log(dn))/\varepsilon$ bound. Thus, there is an $\Omega((n+d)k \log(dn))/\varepsilon$ lower bound for the problem. □

Our $O(1)$-pass upper bounds match the following trivial lower bound, which is immediate from Corollary 1.7 on page 7.

Theorem 4.14. *For any $1 \le k \le \min(n,d)$ and any $\epsilon > 0$, any multi-pass algorithm for the Rank-k Approximation Problem with probability of error at most $1/3$ must use $\Omega((n+d)k\log(nd))$ bits of space. Moreover, this holds for any ordering of the entries of A.*

5 Rank Decision

Theorem 5.1. *Suppose A is an $n \times n$ matrix. The Rank Decision Problem can be solved in 1-pass with $O(k^2 \log n/\delta)$ bits of space with error probability at most δ.*

Proof. We need the following standard fact.

Fact 5.2. *Let $\mathbb{F}$ be a finite field containing at least $v+n$ distinct items. Let $a_1, ..., a_v$ be a subset of v distinct items of $\mathbb{F}$. Consider the $v \times n$ Vandermonde matrix V defined over $\mathbb{F}$ as follows. The entry $V_{i,j}$ is given by a_i^{j-1}. Then any subset of at most n rows of V has full rank.*

We need the following extension to Fact 5.2. Let B be an upper bound on the absolute value of an integer appearing in the stream, and in the underlying matrix represented by the stream. We assume that $\log B = O(\log n)$. According to Fact 5.2, if $q \in [B+n, 2B+2n]$ is prime, then a Vandermonde matrix V defined on items $1, 2, \ldots, B+n$ over $GF(q)$ has the property that any subset of at most n rows of V has full rank over $GF(q)$. Now treat the matrix V as an integer matrix. The claim is that any subset of at most n rows of V has full rank over $\mathbb{R}$. Indeed, otherwise there is a non-trivial linear combination among such a subset of rows. The coefficients of this linear combination can be assumed to be rational, since irrational numbers must cancel out. By scaling, the coefficients can be assumed to be integers. Finally, by dividing out common factors, one can assume the greatest common divisor of the coefficients is 1. It follows that by taking the coefficients modulo q, one obtains a non-trivial linear combination over $GF(q)$, a contradiction. We call the integer matrix thus-obtained a *real scalable Vandermonde matrix.*

Let H be a $3nk/\delta \times n$ real scalable Vandermonde matrix. The algorithm chooses $2k$ random rows $h_{i_1}, \ldots, h_{i_{2k}}$ from H. Consider the $k \times n$ matrix H' containing the first k of these rows, and the $n \times k$ matrix H'' whose columns are the last k of these rows. The algorithm maintains $M = H' \cdot A \cdot H''$, and outputs 1 iff the rank of M equals k.

By definition of H, its entries are integers expressible in $O(\log(n/\delta))$ bits. Since the entries of A are integers expressible with $O(\log n)$ bits, the matrix M can be maintained with $O(k^2 \log(n/\delta))$ bits. Choosing the $2k$ random rows of H can be done with $O(k \log(n/\delta))$ bits of space, and entries of H' and H'' can be generated on the fly. So the overall space complexity is $O(k^2 \log(n/\delta))$ bits.

Notice that if the rank of A is less than k, then the algorithm will never err. Thus, suppose it is at least k, and we will show that $H'AH''$ has rank k with probability at least $1-\delta$.

We will use the following lemma.

Lemma 5.3. *If $L \subset \mathbb{R}^n$ is a j-dimensional linear subspace, and A has rank k, then the dimension of $L_A := \{w \in \mathbb{R}^n \mid w^T A \in L\}$ is at most $n - k + j$.*

Proof. Let $v_1, \ldots, v_r$ be a basis for $L \cap \mathrm{image}(A)$, where $r \leq j$. For $1 \leq i \leq r$, let w_i satisfy $w_i^T A = v_i$. Let $\{b_1, \ldots, b_{n-k}\}$ be a basis for the kernel of A.

If $w \in L_A$, then $w^T A = \sum_i \alpha_i v_i$ for scalars α_i, but also $(\sum_i \alpha_i w_i)^T A = \sum_i \alpha_i v_i$, so $w - \sum_i \alpha_i w_i$ is in the kernel of A, so $w = \sum_i \alpha_i w_i + \sum_j \beta_j b_j$ for scalars β_j. So L_A is a subspace of $\mathrm{span}(\{w_1, \ldots, w_r, b_1, \ldots, b_{n-k}\})$, so its dimension is at most $r + n - k \leq j + n - k$. ☐

For $j < k$, consider the linear subspace L_j spanned by the first j rows of HA. By the above lemma, the dimension of the subspace $L'_j := \{w \in \mathbb{R}^n \mid w^T A \in L_j\}$ is at most $n - k + j$. Since the rows of H are linearly independent, at most $n - k + j$ of them can be in L'_j. Therefore the probability that $h'_{j+1:}A$ is not in L_j is at least $1 - (n - k + j)/(3nk/\delta - j)$, and the probability that all such events hold, for $j = 0 \ldots k - 1$, is at least

$$[1 - n/(3nk/\delta - k)]^k = \left[1 - \frac{1}{k}\frac{\delta/3}{1 - \delta/n}\right]^k \geq 1 - \delta/2,$$

for small enough δ. All such independence events occur if and only if $H'A$ has rank k, and so the probability of the latter is at least $1 - \delta/2$.

Applying a similar argument on the columnspace of $H' \cdot A$, it follows that with probability at least $1 - \delta/2 - \delta/2 = 1 - \delta$, the rank of $M = H' \cdot A \cdot H''$ is at least k. ☐

Via a few binary searches, one can design an algorithm using $O(\log \mathrm{rank}(A))$ passes and $O(\mathrm{rank}^2(A) \log(n/\delta))$ space to actually compute the rank of A based on the above decision problem. We omit the details.

Theorem 5.4. *Any randomized 1-pass algorithm which solves the Rank Decision Problem with probability of error at most 1/3 must use $\Omega(k^2)$ bits of space.*

Proof. We reduce from the IND problem on strings x of length $k^2/4$. Recall that Alice has $x \in \{0,1\}^{k^2/4}$ and Bob has $i \in [k^2/4]$. Let I be the $k/2 \times k/2$ identity matrix, Z the $k/2 \times k/2$ all zeros matrix, and M a $k/2 \times k/2$ matrix whose entries agree with the values of x under some arbitrary correspondence. Alice creates the $k \times k$ block matrix A:

$$A = \begin{pmatrix} I & Z \\ M & I \end{pmatrix}$$

Suppose x_i equals the (r,s)-th entry of M. Bob creates a $k \times k$ matrix B as follows: $B_{s,s} = 1$, $B_{k/2+r,k/2+r} = 1$, $B_{s,k/2+r} = -1$, and all other entries are set to 0. Alice and Bob engage in a 1-round protocol for the Rank Decision Problem on the matrix $A - B$. If the rank is determined to be at least k, Bob outputs 0 as his guess for x_i, otherwise he outputs 1.

To show this protocol is correct for IND with probability at least 2/3, it suffices to show that $\mathrm{rank}(A-B) = k$ iff $M_{r,s} = 0$. The important observation is that by using the top $k/2$ rows of $A-B$, excluding the s-th row, as well as the last $k/2$ columns of $A-B$, excluding the $(k/2+r)$-th column, one can row and column-reduce $A-B$ to a matrix A' for which $A'_{\ell,\ell} = 1$ for all $\ell \notin \{k, k/2+r\}$, $A'_{k/2+r,s} = M_{r,s}$, $A'_{s,k/2+r} = 1$, and all other entries are 0. It follows that if $M_{r,s} = 1$ then A' is a permutation matrix, and thus of rank k. Otherwise A' has rank $k-1$. As $\mathrm{rank}(A') = \mathrm{rank}(A-B)$, the protocol is correct for IND with probability at least 2/3, and so the space complexity of any 1-pass randomized algorithm for the Rank Decision Problem is $\Omega(k^2)$ bits. $\square$

Any algorithm which computes the rank of A also solves the Rank Decision Problem for $k = \mathrm{rank}(A)$, so it has complexity $\Omega(\mathrm{rank}(A)^2)$. As the instance in the Rank Decision Problem concerns distinguishing a rank-k from a rank-$k-1$ square $k \times k$ matrix, it also gives an $\Omega(n^2)$ lower bound for testing if an $n \times n$ matrix is invertible and for approximating the determinant to within any relative error. By adjusting the diagonal values in the upper left quadrant of the matrix A in the proof, one easily obtains an $\Omega(n^2)$ space bound for approximating the i-th largest eigenvalue for any value of i.

Acknowledgments: We thank the anonymous referees for helpful comments. The second author would also like to thank Dan Feldman, Morteza Monemizadeh, and Christian Sohler for helpful discussions.

References

[AGMS02] N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. *J. Comput. Syst. Sci.*, 64(3):719–747, 2002. 1.1

[AM07] D. Achlioptas and F. Mcsherry. Fast computation of low-rank matrix approximations. *J. ACM*, 54(2):9, 2007. 1.1

[AMS99] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. 1.1, 2.1

[BY02] Z. Bar-Yossef. The complexity of massive data set computations, 2002. 1.1

[BY03] Z. Bar-Yossef. Sampling lower bounds via information theory. In *STOC*, pages 335–344, 2003. 1.1

[CCFC02] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *ICALP*, pages 693–703, 2002. 1.1

[CM05] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005. 1.1

[Cop97] D. Coppersmith. Rectangular matrix multiplication revisited. *J. Complexity*, 13(1):42–49, 1997. 2.3

[CS91] J. I. Chu and G. Schnitger. The communication complexity of several problems in matrix computation. *J. Complexity*, 7(4):395–407, 1991. 1.1

[CS95] J. I. Chu and G. Schnitger. Communication complexity of matrix computation over finite fields. *Mathematical Systems Theory*, 28(3):215–228, 1995. 1.1

[DKM06] P. Drineas, R. Kannan, and M. W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006. 4.1.4

[DMM08] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error *cur* matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008. 1.1, 4.1.4

[DMMS07] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. Technical report, 2007. arXiv:0710.1435. 3.1, 4.1

[DV06] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In *APPROX-RANDOM*, pages 292–303, 2006. 1.1

[FKV04] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004. 1

[FMSW09] D. Feldman, M. Monemizadeh, C. Sohler, and D. Woodruff. Coresets and sketches for high dimensional subspace approximation problems, 2009. 1.1, 4.1

[HP98] X. Huang and V. Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complexity*, 14(2):257–299, 1998. 2.3

[HP06] S. Har-Peled. Low-rank approximation in linear time, 2006. 1.1

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997. 1.5

[KNW09] D. Kane, J. Nelson, and D. Woodruff. Revisiting norm estimation in data streams, 2009. 1.2

[MNSW98] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998. 1.6

[Mut05] S. Muthukrishnan. *Data streams: algorithms and applications.* Foundations and Trends in Theoretical Computer Science, 2005. 1, 1.1

[Sar06] T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 143–152, Washington, DC, USA, 2006. IEEE Computer Society. 1, 1.1, 1.1, 2.1, 3.1, 3.1, 4.1

[Vu07] V. H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007. 1.1

A Proofs of Moment Bounds

A.1 Proof of Lemma 2.3

For convenience, here is a restatement of the lemma.

Lemma 2.3 *Given matrices A and B, suppose S is a sign matrix with $m > 15$ columns, and A, B, and S have the same number of rows. Then there is an absolute constant C so that for integer $p > 1$ with $m > Cp$,*

$$\mathbf{E}_p\left[\|A^T SS^T B/m - A^T B\|^2\right] \leq 4((2p-1)!!)^{1/p}\|A\|^2\|B\|^2/m.$$

This bound holds also when S is $4p$-wise independent.

Proof. Let

$$Y_{ijk} := (a_{:i}^T s_{:k} s_{:k}^T b_{:j} - a_{:i}^T b_{:j})/m = m^{-1}\sum_{v\neq w} a_{vi}s_{vk}b_{wj}s_{wk}, \tag{18}$$

so that $\mathbf{E}\, Y_{ijk} = 0$, and

$$\begin{aligned}
\|A^T SS^T B/m - A^T B\|^2 &= \sum_{i,j}\left(\sum_k Y_{ijk}\right)^2 \\
&= \sum_{i,j}\sum_{k,\ell} Y_{ijk}Y_{ij\ell} \qquad (19)\\
&= m^{-2}\sum_{i,j}\sum_{k,\ell}\sum_{v\neq w} a_{vi}s_{vk}b_{wj}s_{wk}\sum_{v'\neq w'} a_{v'i}s_{v'\ell}b_{w'j}s_{w'\ell} \qquad (20)
\end{aligned}$$

Expanding the pth power of (20) on the previous page, it follows that

$$\begin{aligned}
&\mathbf{E}\left[\|A^T SS^T B/m - A^T B\|^{2p}\right] \\
&= m^{-2p} \sum_{\substack{i_1,j_1,k_1,\ell_1,v_1,w_1,v_1',w_1' \\ \vdots \\ i_p,j_p,k_p,\ell_p,v_p,w_p,v_p',w_p'}} \begin{array}{r} a_{v_1 i_1} b_{w_1 j_1} a_{v'_1 i_1} b_{w'_1 j_1} \cdots a_{v_p i_p} b_{w_p j_p} a_{v'_p i_p} b_{w'_p j_p} \\ \mathbf{E}[s_{v_1 k_1} s_{w_1 k_1} s_{v_1' \ell_1} s_{w_1' \ell_1} \cdots s_{v_p k_p} s_{w_p k_p} s_{v_p' \ell_p} s_{w_p' \ell_p}], \end{array}
\end{aligned} \tag{21}$$

where no v_q and w_q are equal, and also no v_q' and w_q' pair are equal, for $q \in [p]$. For odd integer z, $\mathbf{E}[s_{vk}^z] = 0$, and so the summand in the above expression is nonzero only if each s_{vk} appears an even number of times: for example, for $s_{v_1 k_1}$, there must be some other s_{vk} such that $k = k_1$ and $v = v_1$. That is, the equalities of the indices of the s terms form a matching.

Such a matching implies equalities on the k_i and ℓ_i indices; for every $q \leq 2p$, we will bound the number $T(q)$ of assignments to a set of q such k and ℓ indices, and equality constraints on the associated v, w, so that a matching results, and a summand with that index assignment and those equality constraints can be nonzero. Consider in particular the set of assignments in $[m]$ to k_1, and for $z \geq 1$, a collection of $z-1$ k and ℓ indices assigned the same value. There are at most $(2z-1)!! = (2z)!/2^z z!$ distinct matchings possible on the associated v_i, w_i, v_i', and w_i', where a matching on such variables implies a set of equality constraints. Thus

$$T(q) \leq m \sum_{2 \leq z \leq q} \binom{q-1}{z-1} (2z-1)!! T(q-z),$$

with $T(0) = 1$, $T(1) = 0$ and $T(2) = 2m$. (For the $T(2)$ bound the inequality conditions on the v_i, w_i and v_i', w_i' pairs are used.)

Lemma A.1 on the following page states that for $T()$ satisfying these conditions,

$$T(2p) \leq 2^p m^p (2p!)/p! = (4m)^p (2p-1)!!. \tag{22}$$

For fixed $i_1, j_1, \ldots i_p, j_p$, and one such matching, consider the sum of all the summands that satisfy the matching. For a given s-entry, say $s_{v_1 k_1}$, there is some other s-entry, say $s_{w_2' \ell_2}$, with $k_1 = \ell_2$ and $v_1 = w_2'$. As $v_1 = w_2'$ varies from 1 to n, the product $a_{v_1 i_1} b_{w_2' j_2}$ takes on the values of the summands in the dot product $a_{:i_1}^T b_{:j_2}$, since $v_1 = w_2'$. That is, for a fixed matching, as counted by $T()$, the total is the product of a collection of dot products, where each $a_{:i_1}$ and each $b_{:j_2}$ appears exactly twice. Since $x \cdot y \leq \|x\|\|y\|$, the total of the summands, for a fixed matching and fixed $i_1, j_1, \ldots i_p, j_p$, is

$$\|a_{:i_1}\|^2 \|b_{:j_1}\|^2 \cdots \|a_{:i_p}\|^2 \|b_{:j_p}\|^2, \tag{23}$$

and the total, over all matchings, and using (22) on the previous page, is no more than

$$m^{-2p} \sum_{i_1,j_1,\dots i_p,j_p} (4m)^p (2p-1)!! \|a_{:i_1}\|^2 \|b_{:j_1}\|^2 \cdots \|a_{:i_p}\|^2 \|b_{:j_p}\|^2$$
$$= (4m)^p (2p-1)!! \|A\|^{2p} \|B\|^{2p}.$$

A given summand may satisfy the index equalities of more than one matching, but (23) on the preceding page is an upper bound even when all summands are nonnegative; thus multiple appearances can only make the upper bound less tight. We have

$$\mathbf{E}\left[\|A^T SS^T B/m - A^T B\|^{2p}\right] \leq 4^p (2p-1)!! \|A\|^{2p} \|B\|^{2p}/m^p, \tag{24}$$

and the claims of the theorem follow by taking the p'th root. □

Lemma A.1. *If $T(q)$ is such that*

$$T(q) \leq m \sum_{2 \leq z \leq q} \binom{q-1}{z-1} (2z-1)!! T(q-z),$$

with $T(0) = 1$, $T(1) = 0$ and $T(2) = 2m$, then

$$T(2p) \leq 2^p m^p (2p!)/p! = (4m)^p (2p-1)!!.$$

Note that if the recurrence sum for $T(q)$ had only its first term, so that $T(q) \leq m(q-1)3!!T(q-2)$, then a bound like $(3m)^{q/2}(q-1)!!$, or $(3m)^p(2p-1)!!$, immediately follows. When $z = q$, the summand includes the term $(2q-1)!! = (4p-1)!!$, which is too strong a dependence on p; however, that term is only linear in m, and for m large enough relative to p, this causes no harm, as will be shown.

Proof. Suppose $T(x) \leq 2^{x/2} m^{x/2} x!/\lceil x/2 \rceil!$ for $x < q$. From the recurrence, $T(3)$ satisfies this bound when $m \geq 25/2$. Thus it remains to prove the bound inductively for $q \geq 4$.

$$\begin{aligned} T(q) &\leq m \sum_{2 \leq z \leq q} \binom{q-1}{z-1} (2z-1)!! 2^{(q-z)/2} m^{(q-z)/2} (q-z)!/\lceil (q-z)/2 \rceil! \\ &= m \sum_{2 \leq z \leq q} \frac{z}{q} \frac{q!}{z!(q-z)!} \frac{(2z)!}{2^z z!} 2^{(q-z)/2} m^{(q-z)/2} (q-z)!/\lceil (q-z)/2 \rceil! \\ &= 2^{q/2} m^{q/2} q!/\lceil q/2 \rceil! \sum_{2 \leq z \leq q} \frac{z}{q} \binom{2z}{z} 2^{-3z/2} m^{1-z/2} \lceil q/2 \rceil!/\lceil (q-z)/2 \rceil!, \end{aligned} \tag{25}$$

and we need to show that the sum is no more than one. When $z = 2$ and $q \geq 4$, the summand is

$$(2/q)(6)2^{-3} m^0 \lceil q/2 \rceil!/[\lceil (q/2 \rceil - 1]! = (3/4)(2/q)\lceil q/2 \rceil \leq 9/10.$$

When $z \geq 3$, using $\binom{2z}{z} \leq (2e/z)^z$ and $\lceil q/2 \rceil - \lceil (q-z)/2 \rceil \leq z/2$, we have

$$\frac{z}{q}\binom{2z}{z} 2^{-3z/2} m^{1-z/2} \lceil q/2 \rceil ! / \lceil (q-z)/2 \rceil ! \leq 2z(e/2)^z \left(\frac{q+2}{m}\right)^{z/2-1},$$

which for q/m less than an absolute constant C, has a sum, for $q \geq 4$ and over $z \geq 3$, of no more than $1/10$. Thus the sum in (25) on the previous page is no more than one, and the inductive step follows. For even $q = 2p$,

$$T(2p) \leq 2^p m^p (2p!)/p! = (4m)^p (2p-1)!!,$$

as claimed. $\square$

A.2 Proof of Lemma 2.7

For convenience, here is a restatement of the lemma.

Lemma 2.7 *Given matrix A and sign matrix S with the same number of rows, there is an absolute constant C so that for integer $p > 1$ with $m > Cp$,*

$$\mathbf{E}_p \left[[\|S^T A\|^2/m - \|A\|^2]^2 \right] \leq 4((2p-1)!!)^{1/p} \|A\|^4/m.$$

This bound holds also when S is $4p$-wise independent.

Proof. Let

$$Y_{ik} := (a_{:i}^T s_{:k} s_{:k}^T a_{:i} - a_{:i}^T a_{:i})/m = m^{-1} \sum_{v \neq w} a_{vi} s_{vk} s_{wk} a_{wi},$$

so that $\mathbf{E}\, Y_{ik} = 0$, and

$$\begin{aligned} \|S^T A\|^2 - \|A\|^2 &= \sum_{i,k} Y_{ik} \\ &= m^{-1} \sum_{i,k,v \neq w} a_{vi} s_{vk} s_{wk} a_{wi}. \end{aligned}$$

Expanding the $2p$th power of this expression, we have

$$\begin{aligned} &\mathbf{E}\left[(\|S^T A\|^2 - \|A\|^2)^{2p} \right] \\ &= m^{-2p} \sum_{\substack{i_1, k_1, v_1, w_1 \\ \vdots \\ i_{2p}, k_{2p}, v_{2p}, w_{2p}}} \begin{array}{l} a_{v_1 i_1} a_{w_1 i_1} \cdots a_{v_{2p} i_{2p}} a_{w_{2p} i_{2p}} \\ \mathbf{E}[s_{v_1 k_1} s_{w_1 k_1} \cdots s_{v_{2p} k_{2p}} s_{w_{2p} k_{2p}}], \end{array} \end{aligned}$$

This expression is analogous to (21) on page 44, with the summand including products of $4p$ entries of A in place of $2p$ entries of A with $2p$ entries of B. The bound analogous to (24) on the previous page holds, that is,

$$\mathbf{E}\left[(\|S^T A\|^2 - \|A\|^2)^{2p} \right] \leq 4^p (2p-1)!! \|A\|^{4p}/m^p,$$

and the lemma follows upon taking the pth root. $\square$

A.3 Extension to Other Matrix Norms

Lemma 2.3 on page 9 can be generalized, and an analog of Theorem 2.2 on page 9 involving some different matrix norms then follows.

For matrix A, the additional norms are:

- the entrywise maximum norm $\|A\|_{\max} := \sup_{i,j} |a_{ij}|$,
- the entrywise norms $\|A\|_{(r)} := \left[\sum_{i,j} a_{ij}^r\right]^{1/r}$, for $r \geq 1$, and
- the operator norm

$$\|A\|_{1\to 2} := \sup_x \|Ax\|/\|x\|_1 = \sup_j \|a_{:j}\|$$

Note that $\|A\| = \|A\|_{(2)}$, and we can regard $\|A\|_{\max}$ as $\|A\|_{(\infty)}$.

Lemma A.2. *Given matrices A and B, suppose S is a sign matrix with $m > 15$ columns, and A, B, and S have the same number of rows. Then there is an absolute constant C so that for integer $p > 1$ with $m > Cp$, and integer $r \geq 1$,*

$$\begin{aligned}
&\mathbf{E}_p\left[\|A^T SS^T B/m - A^T B\|_{(2r)}^2\right] \\
&\leq ((2pr-1)!!)^{1/pr}\left[\sum_i a_{:i}^{2r}\right]^{1/r}\left[\sum_i b_{:i}^{2r}\right]^{1/r}/m.
\end{aligned}$$

Proof. To generalize this result to a bound for $\|A^T SS^T B - A^T B\|_{(2r)}^p$, for $r > 1$, refer back to (18) on page 43, the definition of Y_{ijk}, and generalize (19) on page 43 to

$$\begin{aligned}
\|A^T SS^T B - A^T B\|_{(2r)}^{2p} &= \left[\sum_{i,j,k^1,k^2,\ldots k^{2r}} Y_{ijk^1} Y_{ijk^2}\cdots Y_{ijk^{2r}}\right]^{p/r} \\
&= \left[\sum_{\substack{i_{h'},j_{h'},k_{h'}^{g'}\\ g'=1\ldots 2r\\ h'=1\ldots p}} \prod_{\substack{g=1\ldots 2r\\ h=1\ldots p}} Y_{i_h j_h k_h^g}\right]^{1/r} \\
&= m^{-2p}\left[\sum_{\substack{i_{h'},j_{h'},k_{h'}^{g'},v_{h'}^{g'}\neq w_{h'}^{g'}\\ g'=1\ldots 2r\\ h'=1\ldots p}} \prod_{\substack{g=1\ldots 2r\\ h=1\ldots p}} a_{v_h^g i_h} s_{v_h^g k_h^g} b_{w_h^g j_h} s_{w_h^g k_h^g}\right]^{1/r}
\end{aligned}$$

Letting X denote the expression inside the brackets, so that the whole expression is $m^{-2p}X^{1/q}$, we are interested in $\mathbf{E}[m^{-2p}X^{1/q}]$. By Jensen's inequality and the concavity of the q'th root, $\mathbf{E}[m^{-2p}X^{1/q}] \leq m^{-2p}\,\mathbf{E}[X]^{1/q}$, and so we obtain

$$\mathbf{E}[\|A^T SS^T B - A^T B\|_{(2r)}^{2p}]^r \leq m^{-2p} \sum_{\substack{\sum_{i_{h'}, j_{h'}, k_{h'}^{g'}, v_{h'}^{g'} \neq w_{h'}^{g'}} \\ g'=1\ldots 2r \\ h'=1\ldots p}} \prod_{\substack{g=1\ldots 2r \\ h=1\ldots p}} a_{v_h^g i_h} b_{w_h^g j_h}\, \mathbf{E}\left[\prod_{\substack{g=1\ldots 2r \\ h=1\ldots p}} s_{v_h^g k_h^g} s_{w_h^g k_h^g}\right].$$

Comparing to (21) on page 44, the same conditions hold for assignments to the k_h^g indices, and equality constraints on the v_h^g and w_h^g indices, with the quantity $T(q)$ satisfying the same recurrence. Here, however, the maximum q of interest is $2pr$, because that is the number of s-entries. The analog of (23) on page 44 is

$$\|a_{:i_1}\|^{2r}\|b_{:j_1}\|^{2r} \cdots \|a_{:i_p}\|^{2r}\|b_{:j_p}\|^{2r}, \tag{26}$$

due to the larger product term, and so our bound is

$$\begin{aligned}
\mathbf{E}[\|A^T SS^T B - A^T B\|_{(2r)}^{2p}] &\leq m^{-2p}\left[T(2pr)\left[\sum_i a_{:i}^{2r}\right]\left[\sum_i b_{:i}^{2r}\right]\right]^{1/r} \\
&\leq m^{-2p}\left[(4m)^{pr}(2pr-1)!!\left[\sum_i a_{:i}^{2r}\right]^p\left[\sum_i b_{:i}^{2r}\right]^p\right]^{1/r},
\end{aligned}$$

and so

$$\mathbf{E}_p[\|A^T SS^T B - A^T B\|_{(2r)}^2] \leq 4((2pr-1)!!)^{1/pr}\left[\sum_i a_{:i}^{2r}\right]^{1/r}\left[\sum_i b_{:i}^{2r}\right]^{1/r}/m,$$

as claimed. □

The following is an analog of Theorem 2.2.

Theorem A.3. *For A and B matrices with n rows, and given $\delta, \epsilon > 0$, there is $m = \Theta(\log(1/\delta)/\epsilon^2)$, as $\epsilon \to 0$, so that for an $n \times m$ independent sign matrix S,*

$$\mathbf{P}\{\|A^T SS^T B/m - A^T B\|_{max} < \epsilon\|A\|_{1\to 2}\|B\|_{1\to 2}\} \geq 1 - \delta.$$

When $B = A$ has column vectors of unit norm, this result implies that all the dot products of the columns may be estimated with bounded absolute error.

Proof. By a proof analogous to that of Theorem 2.2, but using Lemma A.2,

$$\mathbf{P}\{\|A^T SS^T B/m - A^T B\|_{(2r)}^2 < \epsilon\left[\sum_i a_{:i}^{2r}\right]^{1/r}\left[\sum_i b_{:i}^{2r}\right]^{1/r}\} \geq 1 - \delta. \tag{27}$$

For n-vector x, it holds that

$$\|x\|_\infty \le \|x\|_r \le n^{1/r}\|x\|_\infty. \tag{28}$$

Similarly to the first inequality,

$$\|A^T SS^T B/m - A^T B\|_{\max}^2 \le \|A^T SS^T B/m - A^T B\|_{(2r)}^2.$$

Applying (28) to the vector of squared column norms of A, and with $r \ge \log c$, where c is the maximum number of columns of A and B,

$$\left[\sum_i a_{:i}^{2r}\right]^{1/r} \le c^{1/r}\|A\|_{1\to 2}^2 < e\|A\|_{1\to 2}^2,$$

and similarly for B. These relations, with (27) on the previous page, and adjusting constants, imply the result. $\square$