A Sparser Johnson-Lindenstrauss Transform

Daniel M. Kane[*] Jelani Nelson[†]

Abstract

We give a Johnson-Lindenstrauss transform with column sparsity $s = \Theta(\varepsilon^{-1}\log(1/\delta))$ into optimal dimension $k = O(\varepsilon^{-2}\log(1/\delta))$ to achieve distortion $1 \pm \varepsilon$ with success probability $1 - \delta$. This is the first distribution to provide an asymptotic improvement over the $\Theta(k)$ sparsity bound for all values of ε, δ. Previous work of [Dasgupta-Kumar-Sarlós, STOC 2010] gave a distribution with $s = \tilde{O}(\varepsilon^{-1}\log^3(1/\delta))$[1], with tighter analyses later in [Kane-Nelson, CoRR abs/1006.3585] and [Braverman-Ostrovsky-Rabani, CoRR abs/1011.2590] showing that their construction achieves $s = \tilde{O}(\varepsilon^{-1}\log^2(1/\delta))$. As in the previous work, our scheme only requires limited independence hash functions. In fact, potentially one of our hash functions could be made deterministic given an explicit construction of a sufficiently good error-correcting code.

Our linear dependence on $\log(1/\delta)$ in the sparsity allows us to plug our construction into algorithms of [Clarkson-Woodruff, STOC 2009] to achieve the fastest known streaming algorithms for numerical linear algebra problems such as approximate linear regression and best rank-k approximation. Their reductions to the Johnson-Lindenstrauss lemma require exponentially small δ, and thus a superlinear dependence on $\log(1/\delta)$ in s leads to significantly slower algorithms.

1 Introduction

The Johnson-Lindenstrauss lemma states:

Lemma 1 (JL Lemma [14]). *For any integer $d > 0$, and any $0 < \varepsilon, \delta < 1/2$, there exists a probability distribution on $k \times d$ real matrices for $k = \Theta(\varepsilon^{-2}\log(1/\delta))$ such that for any $x \in \mathbb{R}^d$ with $\|x\|_2 = 1$,*

$$\mathbf{Pr}_A[|\|Ax\|_2^2 - 1| > \varepsilon] < \delta.$$

Proofs of the JL lemma can be found in [1, 2, 3, 8, 10, 12, 14, 15, 17]. The value of k in the JL lemma is known to be optimal [13] (also see a later proof in [15]). Standard proofs of the JL lemma take a distribution over dense matrices (e.g. i.i.d. Gaussian or Bernoulli entries), and thus performing the embedding naïvely takes $O(k \cdot \|x\|_0)$ time where x has $\|x\|_0$ non-zero entries. Recently Dasgupta, Kumar, and Sarlós gave a distribution over matrices where each column has at most $s = \tilde{O}(\varepsilon^{-1}\log^3(1/\delta))$ non-zero entries [7], thus speeding up the embedding time to $O(s \cdot \|x\|_0)$. Their distribution requires $O(ds\log k)$ bits of random seed to sample a matrix. They left open two

[*] Harvard University, Department of Mathematics. dankane@math.harvard.edu. Supported by an NSF Graduate Research Fellowship.

[†] MIT Computer Science and Artificial Intelligence Laboratory. minilek@mit.edu. Supported by a Xerox-MIT Fellowship, and in part by the Center for Massive Data Algorithms (MADALGO) - a center of the Danish National Research Foundation.

[1] We say $g = \tilde{\Omega}(f)$ when $g = \Omega(f/\mathrm{polylog}(f))$, and $g = \tilde{O}(f)$ when $g = O(f \cdot \mathrm{polylog}(f))$.

main directions: (1) understand the sparsity parameter s that can be achieved in a JL distribution, and (2) devise a sparse JL transform distribution which requires few random bits to sample from, for streaming applications where storing a long random seed requires prohibitively large memory.

The previous work [15] of the current authors made progress on both these questions by showing $\tilde{O}(\varepsilon^{-2}\log^2(1/\delta))$ sparsity was achievable by giving an alternative analysis of the scheme of [7] which also only required $O(\log(k/\delta)\log d)$ seed length. Braverman, Ostrovsky, and Rabani later gave an even tighter analysis which improved the sparsity and seed length further by $\log(1/\varepsilon)$ and $\log\log(1/\delta)$ factors [3]. For a discussion of other previous work concerning the JL lemma see [15].

Main Contribution: In this work, we give a new construction which achieves sparsity $s = \Theta(\varepsilon^{-1}\log(1/\delta))$ for ℓ_2 embedding into optimal dimension $k = O(\varepsilon^{-2}\log(1/\delta)$. This is the first sparsity bound which is always asymptotically smaller than k, regardless of how ε and δ are related.

We also describe another construction with sparsity $\tilde{O}(\varepsilon^{-1}\log(1/\delta))$, but which has a much simpler analysis requiring only a page long proof. We analyze the simpler construction in Section 3, and we describe our main contribution in Section 4. For our main contribution, we also show in Section 4 that our analysis is tight, so any further improvement would require a different scheme and not merely a tighter analysis.

In Section 5 we discuss how to use our new scheme to speed up the numerical linear algebra algorithms in [6] for approximate linear regression and best rank-k approximation in the streaming model of computation. We first show that *any* JL distribution automatically provides approximate matrix sketches as defined by Sarlós [19]. While Sarlós also showed this, he lost a logarithmic factor in the target dimension due to a union bound in his reduction; Clarkson and Woodruff avoided this loss in [6], but only for the JL distribution of random Bernoulli matrices. We show a simple general reduction for any JL distribution which incurs no loss in parameters. We then plug in our sparse JL transform to yield faster algorithms using the same space.

2 Conventions and Notation

Definition 2. *For $A \in \mathbb{R}^{n\times n}$, we define the* Frobenius norm *of A as $\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$.*

Definition 3. *For $A \in \mathbb{R}^{n\times n}$, we define the* operator norm *of A as*

$$\|A\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2.$$

In the case A has all real eigenvalues (e.g. it is symmetric), we also have that $\|A\|_2$ is the largest magnitude of an eigenvalue of A.

Henceforth, all logarithms are base-2 unless explicitly stated otherwise. Also, for a positive integer n we use $[n]$ to denote the set $\{1,\ldots,n\}$. S^{d-1} denotes the set of $y \in \mathbb{R}^d$ with $\|y\|_2 = 1$. We also assume $\|x\|_2 = 1$, which is without loss of generality since our embedding is linear. All vectors v are assumed to be column vectors, and v^T denotes its transpose. We often implicitly assume that various quantities are powers of 2 or 4 (such as $1/\delta$), which is without loss of generality. Whenever we discuss space complexity (as in Section 5), we always measure space in bits.

Definition 4. *The* Hamming distance *$\Delta(u,v)$ of two vectors u,v is $|\{i : u_i \neq v_i\}|$. An $(n,k,d)_q$* code *is a set of q^k vectors in $[q]^n$ with all pairwise Hamming distances at least d.*

3 A simple construction

Define $k = C \cdot \varepsilon^{-2}\log(1/\delta)$ for sufficiently large constant C. Let s be some integer dividing k satisfying $s \geq 2\varepsilon^{-1}\log(1/\delta)$. Let $\mathcal{C} = \{C_1, \ldots, C_d\}$ be any $(s, \log_{k/s} d, s - O(s^2/k))_{k/s}$ code.

We specify our JL family by describing the embedded vector y. Define hash functions $\sigma : [d] \times [s] \to \{-1, 1\}$ and $h : [d] \times [s] \to [k/s]$. The former is drawn at random from a $2\log(1/\delta)$-wise independent family, and the latter has $h(i,j)$ being the jth entry of the ith codeword in $\mathcal{C}$. We conceptually view $y \in \mathbb{R}^k$ as being in $\mathbb{R}^{s\times(k/s)}$. Our embedded vector then has $y_{i,j} = \sum_{h(r,i)=j} \sigma(r,i)x_r/\sqrt{s}$. This describes our JL family, which is indexed by σ. Note the sparsity is s.

Analysis of simple construction: We first note

$$\|Ax\|_2^2 = \|x\|_2^2 + \frac{1}{s}\sum_{i\neq j}\sum_{r=1}^{s}\eta_{i,j,r}x_i x_j \sigma(i,r)\sigma(j,r),$$

where $\eta_{i,j,r}$ is 1 if $h(i,r) = h(j,r)$, and $\eta_{i,j,r} = 0$ otherwise. We thus would like that

$$Z = \frac{1}{s}\sum_{i\neq j}\sum_{r=1}^{s}\eta_{i,j,r}x_i x_j \sigma(i,r)\sigma(j,r) \tag{1}$$

is concentrated about 0. Note Z is a quadratic form in σ which can be written as $\sigma^T T \sigma$ for an $sd \times sd$ block-diagonal matrix T. There are s blocks, each $d \times d$, where in the rth block T_r we have $(T_r)_{i,j} = x_i x_j \eta_{i,j,r}/s$ for $i \neq j$ and $(T_r)_{i,i} = 0$ for all i. Now, $\mathbf{Pr}[|Z| > \varepsilon] = \mathbf{Pr}[|\sigma^T T\sigma| > \varepsilon]$. To bound this probability, we use the Hanson-Wright inequality combined with a Markov bound.

Theorem 5 (Hanson-Wright inequality [11])**.** *Let $z = (z_1, \ldots, z_n)$ be a vector of i.i.d. Bernoulli ± 1 random variables. Then for any symmetric $B \in \mathbb{R}^{n\times n}$ and integer $\ell \geq 2$ a power of 2,*

$$\mathbf{E}\left[\left(z^T B z - \mathrm{trace}(B)\right)^\ell\right] \leq 64^\ell \cdot \max\left\{\sqrt{\ell}\cdot\|B\|_F, \ell\cdot\|B\|_2\right\}^\ell.$$

The proof of the Hanson-Wright inequality with constant 64 can be found in [9]. We prove our construction satisfies the JL lemma by applying Theorem 5 with $z = \sigma, T = B$.

Lemma 6. $\|T\|_F^2 = O(1/k)$.

Proof.

$$\|T\|_F^2 = \frac{1}{s^2}\cdot\sum_{i\neq j}x_i^2 x_j^2\cdot\left(\sum_{r=1}^{s}\eta_{i,j,r}\right) = \frac{1}{s^2}\cdot\sum_{i\neq j}x_i^2 x_j^2\cdot(s - \Delta(C_i, C_j)) \leq O(1/k)\cdot\|x\|_2^4 = O(1/k).$$

■

Lemma 7. $\|T\|_2 \leq 1/s$.

Proof. Since T is block-diagonal, its eigenvalues are the eigenvalues of each block. For a block T_r, write $T_r = (1/s)\cdot(S_r - D)$. D is diagonal with $D_{i,i} = x_i^2$, and $(S_r)_{i,j} = x_i x_j \eta_{i,j,r}$, including when $i = j$. Since S_r and D are both positive semidefinite, we have $\|T\|_2 \leq (1/s)\cdot\max\{\|S_r\|_2, \|D\|_2\}$. We have $\|D\|_2 = \|x\|_\infty^2 \leq 1$. For S_r, define u_t for $t \in [k/s]$ by $(u_t)_i = x_i$ if $h(i,r) = t$, and $(u_t)_i = 0$ otherwise. Then $u_1, \ldots, u_{k/s}$ are eigenvectors of S_r each with eigenvalue $\|u_t\|_2^2$, and furthermore they span the image of S_r. Thus $\|S_r\|_2 = \max_t \|u_t\|_2^2 \leq \|x\|_2^2 = 1$. ■

Theorem 8. $\mathbf{Pr}_\sigma[|\|Ax\|_2^2 - 1| > \varepsilon] < \delta$.

Proof. By a Markov bound applied to Z^ℓ for ℓ an even integer,

$$\mathbf{Pr}_\sigma[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_\sigma[Z^\ell].$$

Since $Z = \sigma^T T \sigma$ and $\mathrm{trace}(T) = 0$, applying Theorem 5 with $B = T$, $z = \sigma$, and $\ell \le \log(1/\delta)$ gives

$$\mathbf{Pr}_\sigma[|Z| > \varepsilon] < 64^\ell \cdot \max\left\{O(\varepsilon^{-1}) \cdot \sqrt{\frac{\ell}{k}}, \varepsilon^{-1}\frac{\ell}{s}\right\}^\ell. \tag{2}$$

since the ℓth moment is determined by $2\log(1/\delta)$-wise independence of σ. We conclude the proof by noting that the expression in Eq. (2) is at most δ for $\ell = \log(1/\delta)$ and our choices for s, k. ■

Remark 9. Only using that $\mathcal{C}$ has sufficiently high minimum distance, it is impossible to improve our analysis further. For example, for any $(s, \log_{k/s} d, s - O(s^2/k))_{k/s}$ code $\mathcal{C}$, create a new code $\mathcal{C}'$ which simply replaces the first letter of each codeword with "1"; $\mathcal{C}'$ then still has roughly the same minimum distance. However, in our construction this corresponds to all indices colliding in the first chunk of k/s coordinates, which creates an error term of $(1/s) \cdot \sum_{i \ne j} x_i x_j \sigma(i,r)\sigma(j,r)$. Now, suppose x consists of $t = (1/2) \cdot \log(1/\delta)$ entries each with value $1/\sqrt{t}$. Then, with probability $\sqrt{\delta} \gg \delta$, all these entries receive the same sign under σ and contribute a total error of $\Omega(t/s)$ in the first chunk alone. We thus need $t/s = O(\varepsilon)$, which implies $s = \Omega(\varepsilon^{-1}\log(1/\delta))$.

Remark 10. It is of course important to know whether an $(s, \log_{k/s} d, s - O(s^2/k))_{k/s}$ code exists. By picking h at random from an $O(\log(d/\delta))$-wise independent family and setting $s \ge \Omega(\varepsilon^{-1}\sqrt{\log(d/\delta)\log(1/\delta)})$, it is not too hard to show via the Chernoff bound (or more accurately, Markov's bound applied with the $O(\log(d/\delta))$th moment bound implied by integrating the Chernoff bound) followed by a union bound over all pairs of $\binom{d}{2}$ vectors that h defines a good code with probability $1 - \delta$. We do not perform this analysis here since Section 4 obtains better parameters. We also point out that we may assume without loss of generality that $d = O(\varepsilon^{-2}/\delta)$. This is because there exists an embedding into this dimension with sparsity 1 using only 4-wise independence with distortion $(1+\varepsilon)$ and success probability $1 - \delta$ [5, 20]. It is worth noting that in the construction in this section, potentially h could be deterministic given an explicit code with our desired parameters.

4 An Improved Construction

Our improved construction is the same as in Section 3, except rather than let $\mathcal{C}$ be an arbitrary code, we let the underlying hash function $h : [d] \times [s] \to [k/s]$ be drawn at random from a $2\log(1/\delta)$-wise independent hash family. Note the seed length is $O(\log(1/\delta)\log d)$.

We perform our analysis by bounding the ℓth moment of Z from first principles for $\ell = \log(1/\delta)$ an even integer (for this particular scheme, it seems the Hanson-Wright inequality does not simplify any details of the proof). We then use Markov's inequality to say $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] < \varepsilon^{-\ell} \cdot \mathbf{E}_{h,\sigma}[Z^\ell]$.

Let $Z_r = \sum_{i \ne j} \eta_{i,j,r} x_i x_j \sigma(i,r)\sigma(j,r)$ so that $Z = (1/s) \cdot \sum_{r=1}^s Z_r$. We first bound the tth moment of each Z_r for $1 \le t \le \ell$. As in the Frobenius norm moment bound of [15], and also used later in [3], the main idea is to observe that monomials appearing in the expansion of Z_r^t can be thought of in correspondence with graphs. Notice

$$Z_r^t = \sum_{i_1 \ne j_1, \ldots, i_t \ne j_t} \prod_{u=1}^t \eta_{i_u, j_u, r} x_{i_u} x_{j_u} \sigma(i_u, r)\sigma(j_u, r) \tag{3}$$

Each monomial corresponds to a directed multigraph with labeled edges whose vertices correspond to the distinct i_u and j_u. An $x_{i_u}x_{j_u}$ term corresponds to a directed edge with label u from the vertex corresponding to i_u to the vertex corresponding to j_u. The main idea to bound $\mathbf{E}_{h,\sigma}[Z_r^t]$ is then to group monomials whose corresponding graphs are isomorphic, then do some combinatorics.

Lemma 11. *For $t \le \log(1/\delta)$, $\mathbf{E}_{h,\sigma}[Z_r^t] \le 2^{O(t)} \cdot \begin{cases} s/k & t < \log(k/s) \\ (t/\log(k/s))^t & \text{otherwise} \end{cases}$.*

Proof. Let $\mathcal{G}_t$ be the set of isomorphism classes of directed multigraphs with t labeled edges with distinct labels in $[t]$, where each edge has positive and even degree (the sum of in- and out-degrees), and the number of vertices is between 2 and t. Let $\mathcal{G}'_t$ be similar, but with labeled vertices and connected components as well, where vertices have distinct labels between 1 and the number of vertices, and components have distinct labels between 1 and the number of components. Let f map the monomials appearing in Eq. (3) to the corresponding graph isomorphism class. By $2t$-wise independence of σ, any monomial in Eq. (3) whose corresponding graph does not have all even degrees has expectation 0. For a graph G, we let v denote the number of vertices, m the number of connected components, n_i the number of vertices of degree i, and v_i the number of vertices in the ith component. Let d_v denote the degree of a vertex v. Then,

$$\begin{aligned}
\mathbf{E}_{h,\sigma}[Z_r^t] &= \sum_{i_1\neq j_1,\ldots,i_t\neq j_t} \left(\prod_{u=1}^t x_{i_u}x_{j_u}\right) \cdot \mathbf{E}\left[\prod_{u=1}^t \eta_{i_u,j_u,r}\right] \\
&= \sum_{G\in\mathcal{G}_t} \sum_{\substack{i_1\neq j_1,\ldots,i_t\neq j_t \\ f((i_u,j_u)_{u=1}^t)=G}} \left(\prod_{u=1}^t x_{i_u}x_{j_u}\right) \cdot \mathbf{E}\left[\prod_{u=1}^t \eta_{i_u,j_u,r}\right] \\
&= \sum_{G\in\mathcal{G}_t} \sum_{\substack{i_1\neq j_1,\ldots,i_t\neq j_t \\ f((i_u,j_u)_{u=1}^t)=G}} \left(\frac{s}{k}\right)^{v-m} \cdot \left(\prod_{u=1}^t x_{i_u}x_{j_u}\right) && (4) \\
&\le \sum_{G\in\mathcal{G}_t} \left(\frac{s}{k}\right)^{v-m} \cdot \left(\prod_{i=2}^t n_i!\right) \cdot \frac{1}{\binom{t}{d_1/2,\ldots,d_v/2}} && (5) \\
&\le \sum_{G\in\mathcal{G}_t} \left(\frac{s}{k}\right)^{v-m} \cdot v! \cdot \frac{1}{\binom{t}{d_1/2,\ldots,d_v/2}} && (6) \\
&\le \sum_{G\in\mathcal{G}'_t} \left(\frac{s}{k}\right)^{v-m} \cdot \frac{1}{m!} \cdot \frac{1}{\binom{t}{d_1/2,\ldots,d_v/2}}. && (7)
\end{aligned}$$

We now justify these inequalities. The justification of Eq. (4) is similar to that in the Frobenius norm bound in [15]. That is, $\prod_{u=1}^t \eta_{i_u,j_u,r}$ is determined by $h(i_u,r), h(j_u,r)$ for each $u \in [t]$, and hence its expectation is determined by $2t$-wise independence of h. This product is 1 if i_u and j_u hash to the same element for each u and is 0 otherwise. Each i_u, j_u pair hash to the same element if and only if for each connected component of G, all elements of $\{i_1,\ldots,i_t,j_1,\ldots,j_t\}$ corresponding to vertices in that component hash to the same value. We can choose one element of $[k/s]$ for each component to be hashed to, thus giving $(k/s)^m$ possibilities. The probability of any particular hashing is $(k/s)^{-v}$, and this gives that the expectation of the product is $(s/k)^{v-m}$.

For Eq. (5), note that $(\|x\|_2^2)^t = 1$, and the coefficient of $\prod_{u=1}^{v} x_{a_u}^{d_u}$ in its expansion for $\sum_u d_u = t$ is $\binom{t}{d_1/2,\ldots,d_v/2}$. Meanwhile, the coefficient of this monomial that arises when summing over all $i_1 \neq j_1, \ldots, i_t \neq j_t$ for a particular $G \in \mathcal{G}_\ell$ is $\prod_{i=2}^{t} n_i!$, since we may permute the assignment from indices in x to vertices in G, and this would yield the same monomial if and only if the vertices are of equal degree. Eq. (6) then follows since allowing all permutations only provides an upper bound.

For Eq. (7), we move from isomorphism classes in $\mathcal{G}_t$ to those in $\mathcal{G}'_t$. For any isomorphism class in $\mathcal{G}_t$, there are $v!$ ways to label vertices and $m!$ ways to label connected components.

Fix $v_1, \ldots, v_m$, $t_1, \ldots, t_m$ (where there are t_i edges in the ith component C_i), and the assignment of vertex and edge labels to connected components. We now upper bound the summation in Eq. (7) by considering building the graph G edge by edge, starting with 0 edges. Let the initial graph be G_0, so that we form $G = G_t$ by adding edges in increasing label order. We then want to bound the sum of $1/\binom{t}{d_1/2,\ldots,d_v/2}$ over $G \in \mathcal{G}'_\ell$ which satisfy the quantities we have fixed. Note $1/\binom{t}{d_1/2,\ldots,d_v/2}$ equals $t^{-t} \cdot \prod_{u=1}^{v} \cdot \left(\sqrt{d_u}^{d_u}\right)$ up to a $2^{O(t)}$ factor. Initially, when $t = 0$, our sum is $S_0 = 1$. When considering all ways to add the next edge to G_{u+1} from G_u, an edge $i \to j$ contributes $S_u \cdot \sqrt{d_i d_j}/t$ to S_{u+1}. Summing over all vertices $i \neq j$,

$$\sum_{i \neq j} \sqrt{d_i d_j}/t \leq \frac{1}{t} \cdot \sum_{w=1}^{m} \left(\sum_{i \in \mathrm{C}_\mathrm{w}} \sqrt{d_i} \right)^2 \leq \frac{1}{t} \cdot \sum_{i=1}^{m} t_i v_i,$$

by Cauchy-Schwarz. Since there are $\binom{v}{v_1,\ldots,v_m} \cdot \binom{t}{t_1,\ldots,t_m}$ ways to assign edge and vertex labels to components, Eq. (7) gives

$$\begin{aligned}
\mathbf{E}_{h,\sigma}[Z_r^t] &\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{m=1}^{v/2} \sum_{v_1,\ldots,v_m} \sum_{t_1,\ldots,t_m} \left(\frac{s}{k}\right)^{v-m} \cdot \frac{1}{m^m} \cdot \binom{v}{v_1,\ldots,v_m} \cdot \binom{t}{t_1,\ldots,t_m} \cdot \frac{\left(\prod_{i=1}^{m} (v_i t_i)^{t_i}\right)}{t^t} \\
&\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{m=1}^{v/2} \sum_{v_{\max}=2}^{v-2m+2} \left(\frac{s}{k}\right)^{v-m} \cdot \left(\frac{v^v}{m^m}\right) \cdot v_{\max}^{t-v} \qquad (8) \\
&\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{m=1}^{v/2} \sum_{v_{\max}=2}^{v-2m+2} \left(\frac{s}{k}\right)^{v-m} \cdot (v-m)^t \qquad (9) \\
&\leq 2^{O(t)} \cdot \sum_{v=2}^{t} \sum_{q=1}^{v/2} \left(\frac{s}{k}\right)^{q} \cdot q^t
\end{aligned}$$

where $v_{\max} = \max_i v_i$. Eq. (8) holds since there are at most 2^{v+t} ways to choose the v_i, t_i and $t_i \geq v_i$. Eq. (9) follows since $v \geq 2m$ and thus $v, v_{\max} = O(v-m)$. Setting $q = v - m$ and under the constraint $q \geq 1$, $(s/k)^q \cdot q^t$ is maximized when $q = \max\{1, \Theta(t/\log(k/s))\}$. The lemma follows. ■

Theorem 12. *Our construction in this section gives a JL family with sparsity* $s = O(\varepsilon^{-1} \cdot \log(1/\delta))$.

Proof. For $\ell = \log(1/\delta)$ an even integer and by 2ℓ-wise independence of h, σ,

$$
\begin{aligned}
s^\ell \cdot \mathbf{E}_{h,\sigma}[Z^\ell] &= \sum_{\substack{i_1 < \ldots < i_r \\ \forall j \ t_j > 1 \\ \sum_j t_j = r}} \prod_{j=1}^{r} \mathbf{E}_{h,\sigma}[Z_{i_j}^{t_j}] \\
&\le 2^{O(\ell)} \cdot \sum_{\substack{\ell'=1 \\ q=\ell-\ell'}}^{\ell} \binom{\ell}{q} \cdot \left(\sum_{r'=1}^{\ell'/2} \left(\frac{s}{k}\right)^{r'} \cdot \binom{s}{r'} \cdot r'^{\ell} \right) \qquad (10) \\
&\quad \times \left(\sum_{r=1}^{\lfloor q/\log(k/s) \rfloor} \sum_{\substack{(\ell_1,\ldots,\ell_r) \\ \forall i \ \ell_i \ge \log(k/s) \\ \sum_i \ell_i = q}} \binom{s}{r} \cdot \binom{q}{\ell_1, \ldots, \ell_r} \cdot \left(\prod_{i=1}^{r} \ell_i^{\ell_i} \right) \cdot \log^{-q}(k/s) \right) \\
&\le 2^{O(\ell)} \cdot \sum_{\ell'=1}^{\ell} \left[\max_{1 \le r' \le \ell'/2} \left\{ r'^{\ell'} \cdot \left(\frac{s^2}{kr'} \right)^{r'} \right\} \cdot \max_{1 \le r \le q/\log(k/s)} \left\{ \left(\frac{s}{r} \right)^{r} \cdot q^q \cdot \log^{-q}(k/s) \right\} \right].
\end{aligned}
$$

Eq. (10) follows by separately considering the $t_j < \log(k/s)$ (in which case $\mathbf{E}[Z_{i_j}^{t_j}] = 2^{O(t_j)} \cdot s/k$) and the $t_j \ge \log(k/s)$ (in which case $\mathbf{E}[Z_{i_j}^{t_j}] = 2^{O(t_j)} \cdot (t/\log(k/s))^{t_j}$). The value r' represents the number of distinct i_j such that $t_j < \log(k/s)$, and r represents the number of distinct i_j with $t_j \ge \log(k/s)$. In the former case, we have $\binom{s}{r'}$ choices of i_j, then as we expand Z^ℓ, for each of the ℓ' times we pick a Z_{i_j} from amongst these we have at most r' choices. We also have $\binom{\ell}{q}$ choices of which variables belong to this case when expanding Z^ℓ. For the latter case, for any vector of exponents $(\ell_1, \ldots, \ell_r)$, we have at most $\binom{s}{r}$ choices of Z_{i_j} and $\binom{q}{\ell_1,\ldots,\ell_r}$ ways to form the monomial while still maintaining the same vector of exponents.

Now, by Markov's inequality $\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] \le \varepsilon^{-\ell} \cdot \mathbf{E}_{h,\sigma}[Z^\ell]$. Plugging this into the above and writing $s = C\varepsilon^{-1}\ell$ and $k = C\varepsilon^{-2}\ell$ for C a sufficiently large constant,

$$
\begin{aligned}
\mathbf{Pr}_{h,\sigma}[|Z| > \varepsilon] &< 2^{O(\ell)} \cdot \sum_{\ell'=1}^{\ell} \left[\max_{1 \le r' \le \ell'/2} \left\{ (s\varepsilon)^{-\ell'} r'^{\ell'} \cdot \left(\frac{s^2}{kr'} \right)^{r'} \right\} \right. \\
&\qquad \left. \times \cdot \max_{1 \le r \le q/\log(k/s)} \left\{ (s\varepsilon)^{-q} \left(\frac{s}{r} \right)^{r} \cdot q^q \cdot \log^{-q}(k/s) \right\} \right] \qquad (11) \\
&= 2^{O(\ell)} \cdot \max_{1 \le \ell' \le \ell} \left\{ \max_{1 \le r' \le \ell'/2} \left\{ (C\ell)^{-\ell'} r'^{\ell'} \cdot \left(\frac{C^2 \ell}{Cr'} \right)^{r'} \right\} \right. \\
&\qquad \left. \times \max_{1 \le r \le q/\log(k/s)} \left\{ (C\ell)^{-q} \left(\frac{s}{r} \right)^{r} \cdot q^q \cdot \log^{-q}(1/\varepsilon) \right\} \right\} \\
&\le 2^{O(\ell)} \cdot C^{-\ell'/2} \cdot (C\ell)^{-q} \cdot \left(\frac{\ell \cdot \log(1/\varepsilon)}{C\varepsilon q} \right)^{q/\log(1/\varepsilon)} \cdot q^q \cdot \log^{-q}(1/\varepsilon) \\
&\le 2^{O(\ell)} \cdot C^{-\ell'/2} \cdot C^{-q} \cdot \log^{-q}(1/\varepsilon) \\
&< \delta
\end{aligned}
$$

■

Remark 13. It is worth noting that if one wants distortion $1 \pm \varepsilon_i$ with probability $1 - \delta_i$ simultaneously for all i in some set S, it suffices to set $s = C \cdot \sup_{i \in S} \varepsilon_i^{-1} \log(1/\delta_i)$ and $k = C \cdot \sup_{i \in S} \varepsilon_i^{-2} \log(1/\delta_i)$. Then, for any particular i, we can write $s = C \cdot \varepsilon_i^{-1} \log(1/\delta_i) \cdot t_i$ and $k = C \cdot \varepsilon_i^{-2} \log(1/\delta_i) \cdot t_i'$ with $t_i, t_i' \geq 1$. Then, the bound for the first max term in Eq. (11) is identical, except for an extra multiplicative term $(t_i^2/t_i')^{r'} \cdot t_i^{-\ell'} \leq 1$. For the second max term in Eq. (11), define $\alpha = \log(k/s)$ so that $t_i = \varepsilon^{-1} \cdot 2^{-\alpha}$, where $\alpha \geq 3$ (we can assume $\alpha \geq 3$ since we can assume $\sup_{i \in S} \varepsilon_i < 1/8$ without loss of generality). Then, this term's contribution to the moment is at most $C^{-q} \cdot 2^{\alpha q} \cdot \varepsilon^{q(1-2/\alpha)}$. Then, $f(\alpha) = 2^{\alpha q} \cdot \varepsilon^{q(1-2/\alpha)}$ is $2^{O(q)}$ for $\alpha = 3$ and $\alpha \geq \log(1/\varepsilon)$, and taking derivatives there is only one extremum for f in the interval $[3, \log(1/\varepsilon)]$. This extremum occurs at some $\alpha = \Theta(\sqrt{\log(1/\varepsilon)})$, for which one can verify that again $f(\alpha) = 2^{O(q)}$ at this point.

We now show our analysis is tight. First we note the following standard inequality.

Fact 14 ([18, Proposition B.3]). *For all $t, n \in \mathbb{R}$ with $n \geq 1$ and $|t| \leq n$,*

$$e^t(1 - t^2/n) \leq (1 + t/n)^n \leq e^t.$$

Theorem 15. *For δ smaller than a universal constant (which depends on C where $k = C\varepsilon^{-2}\log(1/\delta)$), our scheme requires $s = \Omega(\varepsilon^{-1}\log(1/\delta))$ to obtain distortion $1 \pm \varepsilon$ with probability $1 - \delta$.*

Proof. First suppose $s \leq 1/(2\varepsilon)$. Consider a vector with $t = \lfloor 1/(s\varepsilon) \rfloor$ non-zero coordinates each of value $1/\sqrt{t}$. If there is exactly one set i, j, r with $i \neq j$ such that $h(i, r) = h(j, r)$ (i.e. exactly one collision), then the total error is $2/(ts) \geq 2\varepsilon$. It just remains to show that this happens with probability larger than δ.

The probability of exactly one collision is

$$\begin{aligned} s \cdot \left[\frac{t! \cdot \binom{k/s}{t}}{(k/s)^t}\right]^{s-1} \cdot \binom{t}{2} \cdot \binom{k}{s} \cdot \left[\frac{(t-2)! \cdot \binom{k/s-1}{t-2}}{(k/s)^t}\right] &\geq s \cdot \left(1 - \frac{st}{k}\right)^{t(s-1)} \cdot \binom{t}{2} \cdot \left(\frac{s}{k}\right)\left(1 - \frac{st}{k}\right)^{t-2} \\ &= \frac{s^2 t(t-1)}{2k} \cdot \left(1 - \frac{st}{k}\right)^{st-2} \\ &\geq \frac{s^2 t(t-1)}{2k} \cdot e^{-s^2t^2/k} \cdot \left(1 - \left(\frac{s^2t^2}{k}\right)^2\right) \quad (12) \\ &= \Omega(1/\log(1/\delta)), \end{aligned}$$

which is larger than δ for δ smaller than a universal constant. Eq. (12) follows from Fact 14.

Now consider $1/(2\varepsilon) < s < c \cdot \varepsilon^{-1}\log(1/\delta)$ for some small constant c. Consider the vector $x = (1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots, 0)$. Suppose there are exactly $2s\varepsilon$ collisions, i.e. $2s\varepsilon$ distinct values of r such that $h(1, r) = h(2, r)$ (to avoid tedium we disregard floors and ceilings and just assume $s\varepsilon$ is an integer). Also, suppose that in each colliding chunk r we have $\sigma(1, r) = \sigma(2, r)$. Then, the total error would be 2ε. It just remains to show that this happens with probability larger than δ. The probability of signs agreeing in exactly $2\varepsilon s$ chunks is $2^{-2\varepsilon s} > 2^{-2c\log(1/\delta)}$, which is larger than $\sqrt{\delta}$ for $c < 1/4$. The probability of exactly $2\varepsilon s$ collisions is

$$\binom{s}{2\varepsilon s}\left(\frac{s}{k}\right)^{2\varepsilon s}\left(1 - \frac{s}{k}\right)^{(1-2\varepsilon)s} \geq \left(\frac{s}{2\varepsilon k}\right)^{2\varepsilon s}\left(1 - \frac{s}{k}\right)^{(1-2\varepsilon)s}$$

It suffices for the right hand side to be at least $\sqrt{\delta}$ since h is independent of σ, and thus the total probability of error larger than 2ε would be greater than $\sqrt{\delta}^2 = \delta$. Taking natural logarithms, it suffices to have

$$2\varepsilon s \ln\left(\frac{2\varepsilon k}{s}\right) - (1-2\varepsilon)s\ln\left(1-\frac{s}{k}\right) \le \ln(1/\delta)/2.$$

Writing $s = q/\varepsilon$ and $a = 2C\log(1/\delta)$, the left hand side is $2q\ln(a/q) + \Theta(s^2/k)(1-2\varepsilon)$. Taking a derivative shows $2q\ln(a/q)$ is monotonically increasing for $q < a/e$. Thus as long as $q < ca$ for a sufficiently small constant c, $2q\ln(a/q) < \ln(1/\delta)/4$. Also, the $\Theta(s^2/k)$ term is at most $\ln(1/\delta)/4$ for c sufficiently small. ■

5 Faster numerical linear algebra streaming algorithms

The works of [6, 19] gave algorithms to solve various approximate numerical linear algebra problems given small memory and a only one or few passes over the matrix. They considered models where one only sees a row or column at a time of some matrix $A \in \mathbb{R}^{d\times n}$. Another update model considered was the turnstile streaming model. In this model, the matrix A starts off as 0. One then sees a sequence of m updates $(i_1, j_1, v_1), \ldots, (i_m, j_m, v_m)$, where each update (i, j, v) triggers the change $A_{i,j} \leftarrow A_{i,j} + v$. The goal in all these models is to compute some functions of A at the end of seeing all rows, columns, or turnstile updates. The algorithm should use little memory (much less than what is required to store A explicitly). Both works [6, 19] solved problems such as approximate linear regression and best rank-k approximation by reducing to the problem of sketches for approximate matrix products. Before delving further, first we give a definition.

Definition 16. *A distribution over $\mathbb{R}^{k\times d}$ has (ε, δ)-*JL moments *if for $\ell = \log(1/\delta)$ and $\forall x \in S^{d-1}$,*

$$\mathbf{E}_S\left[\left|\|Sx\|_2^2 - 1\right|^\ell\right] \le (\varepsilon/2)^\ell.$$

Now, the following theorem is a generalization of [6, Theorem 2.1]. The theorem states that any distribution with JL moments also provides a sketch for approximate matrix products. A similar statement was made in [19, Lemma 6], but that statement was slightly weaker in its parameters because it resorted to a union bound, which we avoid using Minkowski's inequality.

Theorem 17. *Given $0 < \varepsilon, \delta < 1/2$, let $\mathcal{D}$ be any distribution over matrices with d columns with the (ε, δ)-JL moment property. Then for A, B any real matrices with d rows and $\|A\|_F = \|B\|_F = 1$,*

$$\mathbf{Pr}_{S\sim\mathcal{D}}\left[\|A^TS^TSB - A^TB\|_F > 3\varepsilon/2\right] < \delta.$$

Proof. Let $x, y \in \mathbb{R}^d$ each have ℓ_2 norm 1. Then

$$\langle Sx, Sy\rangle = \frac{\|Sx\|_2^2 + \|Sy\|_2^2 - \|S(x-y)\|_2^2}{2}$$

so that

$$\begin{aligned}\mathbf{E}\left[|\langle Sx, Sy\rangle - \langle x, y\rangle|^\ell\right] &= \frac{1}{2^\ell}\cdot\left(\mathbf{E}\left[\left|(\|Sx\|_2^2 - 1) + (\|Sy\|_2^2 - 1) - (\|S(x-y)\|_2^2 - \|x-y\|_2^2)\right|^\ell\right]\right)\\ &\le \frac{3^\ell}{2^\ell}\cdot\max\left\{\mathbf{E}\left[\left|\|Sx\|_2^2 - 1\right|^\ell\right], \mathbf{E}\left[\left|\|Sy\|_2^2 - 1\right|^\ell\right], \mathbf{E}\left[\left|\|S(x-y)\|_2^2 - \|x-y\|_2^2\right|^\ell\right]\right\}\\ &\le \left(\frac{3\varepsilon}{4}\right)^\ell\end{aligned}$$

with the middle inequality following by Minkowski's inequality. Now, if A has n columns and B has m columns, label the columns of A as $x_1, \ldots, x_n \in \mathbb{R}^d$ and the columns of B as $y_1, \ldots, y_m \in \mathbb{R}^d$. Define the random variable $X_{i,j} = 1/(\|x_i\|_2\|y_j\|_2) \cdot (\langle Sx_i, Sy_j\rangle - \langle x_i, y_j\rangle)$. Then $\|A^TS^TSB - A^TB\|_F^2 = \sum_{i\neq j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot X_{i,j}^2$. Then again by Minkowski's inequality,

$$\begin{aligned}
\mathbf{E}\left[\left(\|A^TS^SB - A^TB\|_F^2\right)^{\ell/2}\right] &= \mathbf{E}\left[\left|\sum_{i\neq j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot X_{i,j}^2\right|^{\ell/2}\right] \\
&\le \left(\sum_{i\neq j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot \mathbf{E}[|X_{i,j}|^{\ell}]^{2/\ell}\right)^{\ell/2} \\
&\le \left(\sum_{i\neq j} \|x_i\|_2^2 \cdot \|y_j\|_2^2 \cdot (3\varepsilon/4)^2\right)^{\ell/2} \\
&\le (3\varepsilon/4)^{\ell} \cdot (\|A\|_F^2 \cdot \|B\|_F^2)^{\ell/2} \\
&= (3\varepsilon/4)^{\ell}.
\end{aligned}$$

For $\ell = \log(1/\delta)$, $\mathbf{Pr}\left[\|A^TS^SB - A^TB\|_F > 3\varepsilon/\,2\right] < (2\varepsilon/3)^{-\ell} \cdot \mathbf{E}\left[\|A^TS^SB - A^TB\|_F^{\ell}\right] \le \delta$. ■

Remark 18. Often when one constructs a JL distribution $\mathcal{D}$ over $k \times d$ matrices, it is shown that

$$\forall x \in S^{d-1}\ \forall \varepsilon > 1/\sqrt{k}\ \mathbf{Pr}_{S\sim\mathcal{D}}\left[\left|\|Sx\|_2^2 - 1\right| > \varepsilon\right] < e^{-\Theta(\varepsilon^2 k)}$$

Any such distribution automatically satisfies the $(\varepsilon, e^{-\Theta(\varepsilon^2 k)})$-JL moment property for any $\varepsilon > 1/\sqrt{k}$ by converting the tail bound into a moment bound via integration by parts.

Now we arrive at the main point of this section. Several algorithms for approximate linear regression and best rank-k approximation in [6] simply maintain SA as A is updated, where S comes from the JL distribution with $\Omega(\log(1/\delta))$-wise independent $\pm 1/\sqrt{k}$ entries. In fact though, their analyses of their algorithms only use the fact that this distribution satisfies the approximate matrix product sketch guarantees of Theorem 17. Due to Theorem 17 though, we know that *any* distribution satisfying the (ε, δ)-JL moment condition gives an approximate matrix product sketch. Thus, random Bernoulli matrices may be replaced with our sparse JL distribution of Section 4. We now state some of the algorithmic results given in [6] and describe how our scheme provides improvements in the update time (the time to process new columns, rows, or turnstile updates).

As in [6], when stating our results we will ignore the space and time complexities of storing and evaluating the hash functions in our JL distribution. We deal with this later in Remark 21.

5.1 Linear regression

In this problem we have a $A \in \mathbb{R}^{d\times n}$ and $b \in \mathbb{R}^d$. We would like to compute a vector $\tilde{x}$ such that $\|A\tilde{x} - b\|_F \le (1+\varepsilon) \cdot \min_{x^*} \|Ax^* - b\|_F$ with probability $1 - \delta$. In [6], it is assumed that the entries of A, b require $O(\log(nd))$ bits of precision to store precisely. Both A, b receive turnstile updates.

Theorem 3.2 of [6] proves that such an $\tilde{x}$ can be computed with probability $1 - \delta$ from SA and Sb, where S is drawn from a distribution that simultaneously satisfies both the $(1/2, \eta^{-r}\delta)$ and

$(\sqrt{\varepsilon/r}, \delta)$-JL moment properties for some fixed constant $\eta > 1$, and where $\operatorname{rank}(A) \le r \le n$. Thus due to Remark 13, we have the following.

Theorem 19. *There is a one-pass streaming algorithm for linear regression in the turnstile model where one maintains a sketch of size $O(n^2\varepsilon^{-1}\log(1/\delta)\log(nd))$. Processing each update requires $O(n + \sqrt{n/\varepsilon}\cdot\log(1/\delta))$ arithmetic operations and hash function evaluations.*

Theorem 19 improves the update complexity of [6], which was $O(n\varepsilon^{-1}\log(1/\delta))$.

5.2 Low rank approximation

In this problem, we have an $A \in \mathbb{R}^{d\times n}$ of rank ρ with entries that require precision $O(\log(nd))$ to store. We would like to compute the best rank-r approximation A_r to A. We define $\Delta_r \overset{\text{def}}{=} \|A - A_r\|_F$ as the error of A_r. We relax the problem by only requiring that we compute a matrix A'_r such that $\|A - A'_r\|_F \le (1+\varepsilon)\Delta_r$ with probability $1-\delta$ over the randomness of the algorithm.

Two-pass algorithm: Theorem 4.4 of [6] gives a 2-pass algorithm where in the first pass, one maintains SA where S is drawn from a distribution that simultaneously satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. It is also assumed that $\rho \ge 2r+1$. The first pass is thus sped up again as in Theorem 19.

One-pass algorithm for column/row-wise updates: Theorem 4.5 of [6] gives a one-pass algorithm in the case that A is seen either one whole column or row at a time. The algorithm maintains both SA and SAA^T where S is drawn from a distribution that simultaneously satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. This implies the following.

Theorem 20. *There is a one-pass streaming algorithm for approximate low rank approximation with row/column-wise updates where one maintains a sketch of size $O(r\varepsilon^{-1}(n+d)\log(1/\delta)\log(nd))$. Processing each update requires $O(r + \sqrt{r/\varepsilon}\cdot\log(1/\delta))$ amortized arithmetic operations and hash function evaluations per entry of A.*

Theorem 20 improves the amortized update complexity of [6], which was $O(r\varepsilon^{-1}\log(1/\delta))$.

Three-pass algorithm for row-wise updates: Theorem 4.6 of [6] gives a three-pass algorithm using less space in the case that A is seen one row at a time. Again, the first pass simply maintains SA where S is drawn from a distribution that satisfies both the $(1/2, \eta^{-r}\delta)$ and $(\sqrt{\varepsilon/r}, \delta)$-JL moment properties. This pass is sped up using our sparser JL distribution.

Remark 21. In the algorithms above, we counted the number of hash function evaluations that must be performed. Standard constructions of t-wise independent hash functions over universes with elements fitting in a machine word require $O(t)$ time to evaluate [4]. In our case, this would blow up our update time by factors such as n or r, which could be large. Instead, we use fast multipoint evaluation of polynomials. The standard construction [4] of our desired hash functions mapping some domain $[z]$ onto itself for z a power of 2 takes a degree-$(t-1)$ polynomial p with random coefficients in $\mathbb{F}_z$. The hash function evaluation at some point y is then the evaluation $p(y)$ over $\mathbb{F}_z$. Theorem 22 below states that p can be evaluated at t points in total time $\tilde{O}(t)$. We note that in the theorems above, we are always required to evaluate some t-wise independent hash function on many more than t points per stream update. Thus, we can group these evaluation points into groups of size t then perform fast multipoint evaluation for each group. We borrow this idea from [16], which used it to give a fast algorithm for moment estimation in data streams.

Theorem 22 ([21, Ch. 10])**.** *Let* $\mathbf{R}$ *be a ring, and let* $q \in \mathbf{R}[x]$ *be a degree-*t *polynomial. Then, given distinct* $x_1, \ldots, x_t \in \mathbf{R}$*, all the values* $q(x_1), \ldots, q(x_t)$ *can be computed using* $O(t \log^2 t \log\log t)$ *operations over* $\mathbf{R}$*.*

Acknowledgments

We thank Venkat Chandar, Venkatesan Guruswami, Swastik Kopparty, and Madhu Sudan for useful discussions about error-correcting codes. We thank David Woodruff for answering several questions about [6].

References

[1] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[2] Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006.

[3] Vladimir Braverman, Rafail Ostrovsky, and Yuval Rabani. Rademacher chaos, random Eulerian graphs and the sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1011.2590, 2010.

[4] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[5] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 693–703, 2002.

[6] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2009.

[7] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. A sparse Johnson-Lindenstrauss transform. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 341–350, 2010.

[8] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003.

[9] Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), to appear (see also CoRR abs/0911.3389)*, 2010.

[10] Peter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory. Ser. B*, 44(3):355–362, 1988.

[11] David Lee Hanson and Farroll Tim Wright. A bound on tail probabilities for quadratic forms in independent random variables. *Ann. Math. Statist.*, 42(3):1079–1083, 1971.

[12] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[13] T. S. Jayram and David P. Woodruff. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with low error. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*, 2011.

[14] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[15] Daniel M. Kane and Jelani Nelson. A derandomized sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1006.3585, 2010.

[16] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. Fast moment estimation in data streams in optimal space. *CoRR*, abs/1007.4191, 2010.

[17] Jirí Matousek. On variants of the Johnson-Lindenstrauss lemma. *Random Struct. Algorithms*, 33(2):142–156, 2008.

[18] Rajeev Motwani and Prabakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[19] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[20] Mikkel Thorup and Yin Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 615–624, 2004.

[21] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999.