Matrix Norms in Data Streams: Faster, Multi-Pass and Row-Order

Vladimir Braverman [1] **Stephen Chestnut** [2] **Robert Krauthgamer** [3] **Yi Li** [4] **David Woodruff** [5] **Lin Yang** [6]

Abstract

Given the prevalence of large scale linear algebra problems in machine learning, recently there has been considerable effort in characterizing which functions can be approximated efficiently of a matrix in the data stream model. We study a number of aspects of estimating matrix norms – an important class of matrix functions – in a stream that have not previously been considered: (1) multi-pass algorithms, (2) algorithms that see the underlying matrix one row at a time, and (3) time-efficient algorithms. Our multi-pass and row-order algorithms use less memory than what is provably required in the single-pass and entrywise-update models, and thus give separations between these models (in terms of memory). Moreover, all of our algorithms are considerably faster than previous ones. We also prove a number of lower bounds, and obtain for instance, a near-complete characterization of the memory required of row-order algorithms for estimating Schatten p-norms of sparse matrices. We complement our results with numerical experiments.

1. Introduction

Modern datasets, from text documents and images to social graphs, are often represented as a large matrix $A \in \mathbb{R}^{m\times n}$. In many application domains, including database queries, data mining, network transactions and sensor networks (see e.g. (Liberty, 2013; Wei et al., 2016; Huang & Kasiviswanathan, 2015) for recent examples), the input matrix A is presented to the algorithm as a data stream, i.e., a sequence of items/updates that can take several forms. In the *entry-wise (or insertion-only) model*, each item specifies (i, j, A_{ij}) and provides the value of one entry, in arbitrary order (and the unspecified entries are set to 0). The *row-order model* is similar, except that the items follow the natural order (sorted with i as the primary key, and j as the secondary one). In the *turnstile model*, each stream item has the form (i, j, δ) and represents an update $A_{ij} \leftarrow A_{ij} + \delta$ for $\delta \in \mathbb{R}$ (after initializing A to the all-zeros matrix). These models capture different access patterns, but all three can represent sparse matrices quite efficiently, because zero entries are implicit. As usual, the key parameters of an algorithm in the data-stream model are its memory (also referred to as storage/space requirements) and its running time (per update and to report its output).

Many properties of a matrix are directly related to its spectral characteristics, i.e., its singular values. For example, the number of non-zero singular values is just the matrix rank, which determines the degrees of freedom of a corresponding linear system; the maximum and minimum singular values of a matrix determine its condition number, which in turn determines the hardness of many problems, such as optimization problems; the leading singular values of a matrix determine how well a matrix can be represented by the principal components; and so forth. It is generally hard to compute directly the singular values of a matrix, especially in the streaming model, but luckily, the Schatten norms of the matrix can often be used as surrogates for its spectrum, see e.g. (Zhang et al., 2015; Kong & Valiant, 2016; Di Napoli et al., 2016; Khetan & Oh, 2017). Formally, the *Schatten p-norm* of a matrix $A \in \mathbb{R}^{m\times n}$ is defined, for every $p \geq 1$, as

$$\|A\|_{S_p} \coloneqq \left(\sum_{j\geq 1} \sigma_j^p\right)^{1/p},$$

where $\sigma_1 \geq \cdots \geq \sigma_{\min(m,n)}$ are the singular values of A. This definition naturally extends to all $0 < p < 1$ although then it is not a norm, and also to $p = 0, \infty$ by taking the limit. This is a very important family of matrix norms, and includes as special cases the well-known trace/nuclear norm $||A||_* = \sum_{j\geq 1} \sigma_j = \|A\|_{S_1}$, the Frobenius norm $||A||_F = \left(\sum_{j\geq 1} \sigma_j^2\right)^{1/2} = \|A\|_{S_2}$, and the spectral/operator norm $\|A\|_{op} = \sigma_1(A) = ||A||_{S_\infty}$.

We study algorithms that approximate the Schatten p-norm of a matrix A presented in a data stream. While this problem has attracted significant attention lately (Andoni & Nguyên, 2013; Li et al., 2014; Li & Woodruff, 2016a;b; 2017), our results address three new aspects. First, we design faster and more space-efficient *multi-pass* algorithms. Second, we consider the *row-order model*, which is a com-

*Equal contribution [1]Johns Hopkins University [2]ETH Zurich [3]Weizmann Institute of Science [4]Nanyang Technological University [5]Carnegie Mellon University [6]Princeton University. Correspondence to: Lin F. Yang <lin.yang@princeton.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

mon access pattern for matrix data (see, e.g. (Liberty, 2013)). Third, we design algorithms with faster *update time and/or query time*. The above three aspects were not considered previously for matrix norms, and our work opens the door for further diversification of prevailing models (and thereby of current algorithms). In particular, our results can be applicable to classical scenarios, e.g., where data is stored on disk (or any media where a linear scan is much faster than random access), and potentially lead to performance improvements in other such domains. In the next few subsections, we present our contributions in more detail.

1.1. New Estimator for PSD Matrices (or Even p)

Our first results rely on a new method for estimating the Schatten p-norm $||A||_{S_p}$ of a positive semidefinite matrix (PSD) matrix $A \in \mathbb{R}^{n\times n}$ for integer $p \geq 2$. This method yields two new streaming algorithms in the turnstile model, which require, respectively, one pass and $\lceil p/2 \rceil$ passes over the input. Both algorithms are at least as good as the previous ones in all three standard performance measures of storage, update time, and query time; and each algorithm offers significant improvements in two out of these three. Our one-pass algorithm achieves update time $O(1)$ compared with the previous $\mathrm{poly}(n)$, and query time $O(n^{\omega(1-p/2)})$, where $\omega \leq 2.373$ is the matrix multiplication exponent (Le Gall, 2014), compared with the previous n^{p-2}. And our multi-pass algorithm requires storage that is sublinear in n, compared with $O(n)$ previously. We note that if p is even, then the above results extend to arbitrary $A \in \mathbb{R}^{m\times n}$ (and not only PSD) by a standard argument. A detailed comparison of the bounds is given in Table 1, and the results themselves appear in Section 3.

Throughout the paper, a matrix is called *sparse* if it has at most $O(1)$ non-zero entries per row and per column. We write $\widetilde{O}(f)$ as a shorthand for $O(f \cdot \log^{O(1)} f)$, and write $O_a(f)$ to indicate that the constant in O-notation depends on some parameter a.

Techniques Our technical innovation is an unbiased estimator of $\mathrm{tr}(A^p)$ for a *symmetric* (and not only PSD) matrix $A \in \mathbb{R}^{n\times n}$. To see why this is useful, denote the eigenvalues of A by $\lambda_1 \geq \cdots \geq \lambda_n$, and observe that if A is PSD (or alternatively if p is even), then $\mathrm{tr}(A^p) = \sum_i \lambda_i^p = \sum_i \sigma_i(A)^p = ||A||_{S_p}^p$. Our estimator has the form

$$X \coloneqq \mathrm{tr}(G_1 A G_2^T G_2 A G_3^T \cdots G_p A G_1^T), \qquad (1)$$

where $G_i \in \mathbb{R}^{t\times n}$ are certain random matrices. This estimator X can be computed from the p bilinear sketches $\{G_i A G_{i+1}^T\}_{i\in[p]}$ by straightforward matrix multiplication, where $G_{p+1} \coloneqq G_1$ by convention. And if, say, $t = O(n^{1-2/p})$, then each bilinear sketch has dimension $O(t^2) = O(n^{2-4/p})$. These determine the streaming algorithm's storage requirement and query time, and, if the matrices $\{G_i\}_{i\in[p]}$ have sparse columns, the updates will be fast.

The main difficulty is to bound the estimator's variance, which highly depends on the choice of the matrices $\{G_i\}_{i\in[p]}$. The basics of this technique can be seen in the case $p = 4$, if the G_i's satisfy the following definition.

Definition 1.1. *A random matrix $S \in \mathbb{R}^{t\times n}$ is called an (ϵ, δ, d)-*Johnson-Lindenstrauss Transformation (JLT) *if for every $V \subseteq \mathbb{R}^n$ of cardinality $|V| \leq d$ it holds that*

$$\Pr\left[\forall x \in V,\ \|Sx\|_2^2 \in (1 \pm \epsilon)\|x\|_2^2\right] \geq 1 - \delta.$$

An (ϵ, δ, d)-JLT can be constructed with $t = O(\epsilon^{-2}\log(d/\delta))$ rows, which is optimal (see (Kane et al., 2011) or (Jayram & Woodruff, 2013)). While using independent $N(0, 1/t)$ Gaussians entries works, there is a construction with only $O(\epsilon^{-1}\log(1/\delta))$ non-zero entries per column (Kane & Nelson, 2014).

The case $p = 4$ has a particularly short and simple analysis, whenever G_1 and G_2 are independent (ϵ, δ, n)-JLT matrices, which we can achieve with $t = O(\epsilon^{-2}\log n)$. The first idea is to "peel off" G_i from both sides, using that for any PSD matrix M, with high probability $\mathrm{tr}(G_i M G_i^T) \in (1 \pm \epsilon)\,\mathrm{tr}(M)$ (see Lemma 3.2 for a precise statement). A second idea is to use the identity $\mathrm{tr}(BC) = \mathrm{tr}(CB)$ to rewrite $\mathrm{tr}(AA^T G_2^T G_2 AA^T) = \mathrm{tr}(G_2 AA^T AA^T G_2^T)$. Now using the first idea once again, we are likely to arrive at an approximation to $\mathrm{tr}(AA^T\ AA^T) = ||A||_{S_4}$. The full details are given in Section 3.1.

The sketching method extends from $p = 4$ to any integer $p \geq 2$, but the simple analysis above breaks (because for $p > 4$ the "inside" matrix M is no longer PSD) and thus our analysis is much more involved. We first analyze G_i's with independent Gaussian entries, by a careful expansion of the fourth moment of X, which exploits certain cancellations occurring (only) for Gaussians. We then consider G_i's that are sampled from a particular sparse JLT due to (Thorup & Zhang, 2004), and employ a symmetrization-and-decoupling argument to compare the variance of X in this case with that of Gaussian G_i's.

We make two technical remarks. First, proving $\mathbb{E}[X] = \mathrm{tr}(A^p)$ is straightforward. Indeed, by the second idea above, we can rewrite $X = \mathrm{tr}(G_1 A G_2^T G_2 A G_3^T \cdots G_p A G_1^T)$ as $X = \mathrm{tr}(G_1^T G_1 A G_2^T G_2 A \cdots\ G_p^T G_p A)$. Now using $\mathbb{E}[G_i^T G_i] = I$ together with linearity of trace and of expectation, we obtain that $\mathbb{E}[X] = \mathrm{tr}(A^p)$. Second, after setting $t = O(n^{1-2/p})$ (independent of ϵ), our bound on the variance is $O(\mathbb{E}[X]^2)$, which we can decrease in a standard way, taking $O(1/\epsilon^2)$ repetitions. See Sections 3.2 and 3.4 for details.

The multi-pass streaming algorithm is implemented slightly differently, in that $G_1 \in \mathbb{R}^{1\times n}$, i.e., has only one row. The other matrices $G_2, \ldots, G_p \in \mathbb{R}^{t\times n}$ are as before, although we now set $t = O(n^{1-1/(p-1)})$. Our es-

Problem: Schatten p-norm of PSD A, integer $p \geq 2$ (or general A, even p)				
passes	space	update time	query time	
1	$\epsilon^{-2}n^{2-4/p}$	$\epsilon^{-2}n^{2-4/p}$	$\epsilon^{-2}n^{p-2}$	(Li et al., 2014)
1	$\epsilon^{-2}n^{2-4/p}$	ϵ^{-2}	$\epsilon^{-2}n^{(1-2/p)\omega}$	Theorems 3.3 and 3.8
$\lceil p/2 \rceil$	$\epsilon^{-2}n$	ϵ^{-2}	$\epsilon^{-2}n$	Theorem 6.1 of (Woodruff, 2014)
$\lceil p/2 \rceil$	$\epsilon^{-2}n^{1-1/(p-1)}$	ϵ^{-2}	$\epsilon^{-2}n^{(1-1/(p-1))}$	Theorems 3.6 and 3.8

Table 1. Streaming algorithms for $(1+\epsilon)$-approximation of the Schatten p-norm of $A \in \mathbb{R}^{n\times n}$. The bounds for storage/time omit $O_p(1)$ factors, and count space in words.

timator X can be computed in $\lceil p/2 \rceil$ passes with space only $2t$ as follows. In the first pass, compute vectors $X_L \leftarrow G_1 A G_2^T \in \mathbb{R}^{1\times t}$ and $X_R \leftarrow G_p^T A G_1 \in \mathbb{R}^{t\times 1}$, and then on the i-th pass update $X_L \leftarrow X_L G_i^T A G_{i+1}$ and $X_R \leftarrow G_{p-i+1} A G_{p-i+2}^T X_R$. Notice that the computation in each pass is linear in A. For even p, after completing $p/2$ passes, compute and output $X' = X_L X_R \in \mathbb{R}$ (and similarly for odd p). This X' is similar to the estimator X described above, except for the new dimensions of the G_i's. See Sections 3.3 and 3.4.

This multi-pass algorithm offers a very significant space savings over the one-pass algorithm. It is also a bit surprising because its space is getting close to the corresponding vector norm, namely, the ℓ_p-norm on $\mathbb{R}^n$, for which the optimal space for $O(p)$ passes is $\widetilde{O}(n^{1-2/p})$ bits. In fact, for the vector norm, $O(p)$ passes do not significantly reduce the storage needed compared with one pass, which stands in sharp contrast to Schatten p-norms. As mentioned before, if p is even then the algorithms extends to arbitrary $A \in \mathbb{R}^{m\times n}$ by a standard argument.

1.2. Lower Bound for PSD Matrices

Recent work (Li & Woodruff, 2016a) has improved the storage lower bound for estimating Schatten p-norms for non-integer values of p, by showing that $(1+\epsilon)$-approximation (in the one-pass entry-wise model) requires storage $n^{1-g(\epsilon)}$, for some function $g(\epsilon) \to 0$ as $\epsilon \to 0$, even for a sparse matrix. This contrasts with our algorithms for PSD matrices (from Section 1.1), where the exponent is independent of ϵ and bounded away from 1. However, the hard distribution used by (Li & Woodruff, 2016a) is not over PSD matrices, leaving open the possibility that PSD matrices admit algorithms that use storage $O(n^c)$ for $c < 1$ independent of ϵ.

We close this gap in Section 4, by adapting the lower bound of (Li & Woodruff, 2016a) to PSD matrices, to show, for every non-integer $p > 0$, a storage lower bound of $\Omega(n^{1-g'(\epsilon)})$ for some function $g'(\epsilon) \to 0$ as $\epsilon \to 0$ (again, in the one-pass entry-wise model and even for a sparse matrix). A key feature of our lower bounds for PSD matrices is that they hold in the model in which each entry of the matrix occurs exactly once in the stream. This models applications where the matrix resides in external memory and is being streamed through main memory; in such a model multiple updates to an entry may not appear. While it is possible to obtain lower bounds for PSD matrices by embedding the multiplayer SET-DISJOINTNESS lower bound (Bar-Yossef et al., 2002) for vectors onto the diagonal of a matrix, to apply such lower bounds the diagonal entries need to be incremented repeatedly, that is, one such diagonal entry needs to be updated $n^{\Omega(1)}$ times. In contrast, in our lower bounds each matrix entry occurs exactly once in the stream, i.e., there are no updates to entries.

1.3. Results for Row-Order Model

For sparse matrices, estimating Schatten p-norms in the row-order model can be reduced to estimating Schatten $(p/2)$-norms in the turnstile model. Consider estimating $\|A\|_{S_p}^p$ for some sparse matrix A. The algorithm first forms $A^T A = \sum_i A_i^T A_i$ "on the fly", by reading each row A_i and immediately generating a stream of updates that corresponds to the non-zero entries in $A_i^T A_i$, and then it can just estimate the Schatten $(p/2)$-norm of that stream, because $\|A^T A\|_{S_{p/2}}^{p/2} = \|A\|_{S_p}^p$. Observe that each row A_i has only $O(1)$ non-zero entries, hence also $A_i^T A_i$ has only $O(1)$ non-zero entries, and the algorithm only needs $O(1)$ space to generate the updates to $A^T A$. Moreover, since A is sparse, also $A^T A$ is sparse. It was shown in (Li & Woodruff, 2016a) how to estimate the Schatten p-norm, for an even integer p, using $\widetilde{O}_{p,\epsilon}(n^{1-2/p})$ bits of space, even in the turnstile model. For $p \in 4\mathbb{Z}$, the above yields an algorithm in the row-order model that uses $\widetilde{O}_{p,\epsilon}(n^{1-4/p})$ bits of space for sparse matrices.

In Sections C and D, we study the problem in the row-order model for all $p > 0$. When p is not an even integer, we prove that $(1+\epsilon)$-approximating the Schatten p-norm in the one-pass entry-wise model requires $\Omega_\epsilon(n^{1-g(\epsilon)})$ bits of space where $g(\epsilon) \to 0$ as $\epsilon \to 0$. This bound holds even for sparse matrices, in which case it is almost tight. When $p \geq 4$ is an even integer, we prove a lower bound of $\Omega_p(n^{1-4/p})$ bits of space, matching up to logarithmic factors the algorithm from above for $p \in 4\mathbb{Z}$. For the remaining case $p \equiv 2 \pmod 4$, we present an algorithm using $\widetilde{O}_{p,\epsilon}(n^{1-4/(p+2)})$ space, leaving a slight polynomial gap from the lower bound of $\Omega_p(n^{1-4/p})$.

Problem: Schatten p-norm of a sparse matrix in row-order stream			
	which $p > 0$	space	
Algorithms:	all p	$\widetilde{O}(n)$	trivial (by sparsity), $\epsilon = 0$
	$p \equiv 0 \pmod 4$	$\widetilde{O}_{p,\epsilon}(n^{1-4/p})$	Section 1.3
	$p \equiv 2 \pmod 4$	$\widetilde{O}_{p,\epsilon}(n^{1-4/(p+2)})$	Theorem D.1, $p \geq 6$
Lower Bounds:	$p \in 2\mathbb{Z}, p \geq 4$	$\Omega(n^{1-4/p})$	Theorem C.4, for $\epsilon < \epsilon_0(p)$, even multi-pass
	$p \notin 2\mathbb{Z}$	$\Omega_t(n^{1-1/t})$	Theorem C.3, for $\epsilon < \epsilon_0(t,p)$

Table 2. Bounds for $(1+\epsilon)$-approximation of the Schatten p-norm of a sparse matrix $A \in \mathbb{R}^{n\times n}$ in the one-pass row-order model. Space is counted in bits.

1.4. Previous Work

The aforementioned algorithm of (Li et al., 2014) uses a single sketching matrix G, for example, if A is PSD, then their sketch is $S = GAG^T$, where $G \in \mathbb{R}^{t\times n}$ is a Gaussian matrix. Its estimate for $||A||_{S_p}$ is produced by summing over all "cycles" $S_{i_1,i_2}S_{i_2,i_3}\cdots S_{i_p,i_1}$, where $i_1,\ldots,i_p \in [t]$ are distinct. Our sketch improves upon theirs in both update time and query time. The only other streaming algorithm for Schatten p-norms that we are aware of is that of (Li & Woodruff, 2016a) (Theorem 7), which uses space $O(n^{1-\frac{2}{p}}\operatorname{poly}(\frac{1}{\epsilon},\log n))$ but works only for matrices that have $O(1)$-entries per row and per column.

One possible approach to improve the update time would be to replace the Gaussian matrices in (Li et al., 2014) with a distribution over matrices that admit a fast multiplication algorithm. The analysis done in (Li et al., 2014) relies on the Gaussian entries (rotational invariance, in particular), so the replacement matrix should preserve the distribution of the sketch. Kapralov, Potluru, and Woodruff (Kapralov et al., 2016) present just such a distribution on matrices $\widetilde{G}$, where the multiplication $\widetilde{G}A$ can be computed quickly and $\widetilde{G}A$ is close to GA in total variation distance. Unfortunately, under the distribution of (Kapralov et al., 2016), or any other with a similar guarantee on total variation distance, each coordinate update to A results in a dense rank-one update to the sketch, which means that the update time is not improved.

Several strong lower bounds are known for approximating Schatten p-norms and other matrix functions, both for the dimension of a sketch and for storage requirement (bits). Li, Nguyen and Woodruff (Li et al., 2014) prove that for $0 \leq p < 2$ every linear sketch that can approximate rank and Schatten p-norm must have dimension $\Omega(\sqrt{n})$ and every bilinear sketch must have dimension $\Omega(n^{1-\epsilon})$. Li and Woodruff (Li & Woodruff, 2016b) show that every linear sketch for Schatten p-norms, $p \geq 2$, requires dimension $\Omega(n^{2-4/p})$. In (Li & Woodruff, 2016a), they prove space complexity lower bounds that hold even when the input matrix is sparse. Specifically, they show that one-pass streaming algorithms which $(1\pm\epsilon)$-approximate various functions of the singular values, including Schatten p-norms when p is not an even integer, require $\Omega(n^{1-g(\epsilon)})$ bits of space for some function $g(\epsilon) \to 0$ as $\epsilon \to 0$. Additional space lower bounds, e.g., for $p \in [1,2)$, can be deduced from a general statement of (Andoni et al., 2015), see Table 1 of (Li & Woodruff, 2016a).

2. Notation and Preliminaries

The space bounds of sketching algorithms in the turnstile model are stated in terms of sketch dimension (number of entries). The number of bits required can be larger by a $\log nM$ factor, where M is the absolute ratio of the largest element in the matrix to the smallest. We call a matrix a *Gaussian matrix* if its entries are independent $N(0,1)$ random variables. A matrix G of dimension $t \times n$ is a *column-normalized* Gaussian matrix if $G = G'/\sqrt{t}$, where G' is a Gaussian matrix. Now-standard techniques such as Nisan's pseudorandom generator or k-wise independence can be used to derandomize Gaussian matrices for use in sketching algorithms. Column-normalized Gaussian matrices serve as JLTs. In particular, there exists a constant c such that if G be a $t \times n$ column-normalized Gaussian matrix with $t \geq \frac{c}{\epsilon^2}\log\frac{d}{\delta}$, then G is a (ϵ,δ,d)-JLT (Indyk & Motwani, 1998).

3. New Estimator for PSD Matrices (and Integer p)

The main result in this section is a new one-pass streaming algorithm for estimating the Schatten p-norm, for integer $p \geq 2$. When p is odd, it additionally requires that the input matrix is PSD. The first version of this algorithm, described in Section 3.2, has the same storage requirement of $\widetilde{O}_p(n^{2-4/p}/\epsilon^2)$ bits as the previous algorithm of (Li et al., 2014) that uses cycle sums, but has a simpler analysis and faster query time[1], which is roughly matrix multiplication time, n^ω, instead of n^p. Moreover, it is based on a new method that leads to a $\lceil p/2\rceil$-pass algorithm with storage requirement $\widetilde{O}_p(n^{1-1/(p-1)}/\epsilon^2)$ bits, as described

[1] In (Kong & Valiant, 2016), Kong and Valiant independently improve the algorithm in (Li et al., 2014) to the same runtime as Theorem 3.3 in this paper by considering only "increasing cycles".

in Section 3.3. Previously, the algorithm in Theorem 6.1 of (Woodruff, 2014) has the same number of passes but larger storage requirement $O(n/\epsilon^2)$.[2] Finally, we improve the update time, as described in Section 3.4, by employing the sketching matrices G_i that are certain sparse matrices instead of Gaussians.

We start in Section 3.1 with the case $p = 4$, which is based on the same sketch but is significantly easier to analyse.

3.1. Schatten 4-Norm using JLT matrices

Theorem 3.1. *Let $G_1, G_2 \in \mathbb{R}^{t\times n}$ be independent $(\epsilon, \frac{\delta}{n}, 1)$-JLT matrices. Then for every $A \in \mathbb{R}^{n\times m}$,*

$$\Pr\left[\operatorname{tr}(G_1AA^TG_2^TG_2AA^TG_1^T) \in (1\pm 2\epsilon)^2||A||_{S_4}^4\right] = 1-2\delta.$$

Thus, one can find a $(1\pm\epsilon)$-approximation to the Schatten-4 norm of a general matrix $A \in \mathbb{R}^{n\times m}$ using a linear sketch of dimension $O(\epsilon^{-2}n\log n)$.

Before proving the theorem, we remark that if each column of G_i has only s non-zero entries, it is easy to see that the update time of this linear sketch is $O(s)$, assuming any entry of G_1 and G_2 can be accessed in $O(1)$ time (in a streaming algorithm, the entries are usually computed from a small random seed in $\operatorname{polylog}(n)$ time). The query time is dominated by multiplying a matrix of size $t \times n$ with one of size $n \times t$, and thus takes $O(t^\omega \cdot n/t) = \widetilde{O}(n^\omega/\epsilon^{2(\omega-1)})$ time.

Now we prove Theorem 3.1, for which we need the following lemma.

Lemma 3.2. *Let $G \in \mathbb{R}^{t\times n}$ be an $(\epsilon, \delta/n, 1)$-JLT matrix. Then for every PSD matrix $A \in \mathbb{R}^{n\times n}$,*

$$\Pr\left[\operatorname{tr}(GAG^T) \in (1\pm\epsilon)\operatorname{tr}(A)\right] \geq 1-\delta.$$

Proof. By the Spectral Theorem, $A = U\Lambda U^T$, where Λ is a diagonal matrix and U is an orthonormal matrix. Then $G' = GU$ is still an $(\epsilon, \delta/n, 1)$-JLT. Thus

$$\begin{aligned}\operatorname{tr}(GAG^T) &= \operatorname{tr}(G'\Lambda G'^T) = \operatorname{tr}(\sqrt{\Lambda}G'^TG'\sqrt{\Lambda}) \\ &= \sum_{i=1}^n \lambda_i e_i^T G'^T G' e_i = \sum_{i=1}^n \lambda_i ||G'e_i||_2^2.\end{aligned}$$

By the JLT guarantee and a union bound, with probability at least $1-\delta$, for all $i \in [n]$ we have $||G'e_i||_2^2 \in [1-\epsilon, 1+\epsilon]$, in which case $\operatorname{tr}(GAG^T) \in (1\pm\epsilon)\operatorname{tr}(A)$. □

of Theorem 3.1. Apply Lemma 3.2 to the PSD matrix AA^TAA^T, to get that with probability at least $1-\delta$ (over the choice of G_2),

$$\begin{aligned}\operatorname{tr}(G_2AA^TAA^TG_2^T) &\in (1\pm 2\epsilon)\operatorname{tr}(AA^TAA^T) \\ &= (1\pm 2\epsilon)||A||_{S_4},\end{aligned}$$

where the left-hand side is equal to $\operatorname{tr}(AA^TG_2^TG_2AA^T)$, by the identity $\operatorname{tr}(MM^T) = \operatorname{tr}(M^TM)$. Now suppose (by conditioning) that G_2 is already fixed, and apply the same lemma to the PSD matrix $AA^TG_2^TG_2AA^T$, to get that with probability at least $1-\delta$ (over the choice of G_1),

$$\operatorname{tr}(G_1AA^TG_2^TG_2AA^TG_1^T) \in (1\pm 2\epsilon)\operatorname{tr}(AA^TG_2^TG_2AA^T).$$

The proof follows by a union bound.

The linear sketch of A consists of the two matrices G_1A and G_2A, which suffices to estimate $||A||_{S_4}^4$ as above with $\delta = 1/8$. This sketch is linear and its dimension is $2tn$, where we can use say Gaussians to obtain $t = O(\epsilon^{-2}\log n)$. □

3.2. Schatten p-norm Using Gaussians

We now design a sketch for Schatten-p norm that uses column-normalized Gaussian matrices. We will later extend and refine it to improve the per-update processing time.

Theorem 3.3. *For every $0 < \epsilon < 1/2$ and integer $p \geq 2$, there is an algorithm that outputs a $(1\pm\epsilon)$-approximation to the Schatten-p norm of a PSD matrix $A \in \mathbb{R}^{n\times n}$ using a randomized linear sketch of dimension $s = O_p(\epsilon^{-2}n^{2-4/p})$. The update time (for each entry in A) is $O(s)$ and the query time (for computing the estimate) is $O(\epsilon^{-2}n^{(1-2/p)\omega})$, where $\omega < 2.373$ is the matrix multiplication constant.*

If p is even, the above algorithm extends to a general matrix $A \in \mathbb{R}^{n\times m}$.

The first part of the theorem (for PSD matrices) follows directly from Proposition 3.4 below. The proposition is applicable to all symmetric matrices, but $\|A\|_{S_p}^p = \operatorname{tr}(A^p)$ only for PSD matrices or even p. The linear sketch stores $G_iAG_{i+1}^T$ for $i = 1, \ldots, p$, where by convention $G_{p+1} = G_1$, repeated independently in parallel $O_p(1/\epsilon^2)$ times. Thus, the sketch has dimension $O_p(\epsilon^{-2}t^2)$. The estimator is obtained by computing the $O_p(1/\epsilon^2)$ independent copies of X and reporting their average. To analyze its accuracy, notice that a PSD matrix A satisfies $\mathbb{E}[X] = \operatorname{tr}(A^p) = ||A||_p^p$. Then setting $t = n^{1-2/p}$ gives $\operatorname{Var}(X) \leq O_p(||A||_{S_p})^{2p}$ and averaging multiple independent copies of X reduces the variance.

The second part (for general matrices), follows by using the same sketch for the symmetric matrix $B = \left(\begin{smallmatrix} 0 & A \\ A^T & 0 \end{smallmatrix}\right)$, because the nonzero singular values of B are those of A repeated twice and $||B||_{S_p}^p = 2||A||_{S_p}^p = 2\operatorname{tr}(A^p)$, where the last equality uses the assumption that p is even.

Because the correctness of the algorithm comes from bounding the variance of X, it is enough that the entries in each Gaussian matrix are four-wise independent, which is crucial for applications with limited storage like streaming.

Proposition 3.4. *For integer $p \geq 2$ and $t \geq 1$, let*

[2] We note that also in Theorem 6.1 of (Woodruff, 2014) it is required that p is even or that the input matrix is PSD, but this is erroneously omitted.

$G_1, \ldots, G_p$ be independent $t \times n$ column-normalized Gaussian matrices. Then for every symmetric matrix $A \in \mathbb{R}^{n\times n}$, the estimator $X = \operatorname{tr}\left(G_1 A G_2^T G_2 A \ldots G_p^T G_p A G_1^T\right)$ satisfies

$$\mathbb{E}[X] = \operatorname{tr}(A^p) \text{ and } \operatorname{Var}(X) = O_p\left(1 + \sum_{z=2}^{\lfloor \frac{p}{2} \rfloor + 1} \left(\frac{n^{1-\frac{2}{p}}}{t}\right)^z + \sum_{z=2}^{p} \left(\frac{n^{1-\frac{2}{z}}}{t}\right)^z\right) ||A||_{S_p}^{2p}.$$

The full proof of this proposition is postponed to Section E. We outline the general idea here. It is standard that a Gaussian matrix is rotational invariant, i.e., G and GU are identically distributed for any orthogonal matrix U. Thus, by the Spectral Theorem, instead of considering symmetric matrix $A = U\Lambda U^T$, we can consider only its diagonalization Λ.

The proof of this proposition proceeds first by expanding X in terms of inner products of columns of the matrix G, i.e., $X = \sum_{i_1,i_2,\ldots,i_p \in [n]} \lambda_{i_1}\lambda_{i_2}\ldots\lambda_{i_p} \cdot \langle g_{i_1}^{(1)}, g_{i_2}^{(1)}\rangle \cdot \langle g_{i_2}^{(2)}, g_{i_3}^{(2)}\rangle \ldots \langle g_{i_p}^{(p)}, g_{i_1}^{(p)}\rangle$, where λ_i is the i-th eigenvalue of A and $g_{i_j}^{(j)}$ is the i_j-th column of G_j. We then expand $\mathbb{E}(X^2)$. The non-zero terms in $\mathbb{E}(X^2)$ are composed by only those terms of even powers in every eigenvalue. Computing the expectation of each term is straightforward because the entries of G are independent Gaussian random variables, but the crux of the proof is in bounding the sum of the terms. We introduce a collection of diagrams that aid in enumerating the terms according to their structure and computing the sum.

3.3. Multi-Pass Algorithm

The proof of Proposition 3.4 relies on the matrices G_i being Gaussians in two places. First, we assume that the matrix A is diagonal, and in general we need to consider G_iU instead of G_i. Second, the columns of these matrices have small variance/moments, as described in (7)-(8). We now generalize the proof to relax these requirements (e.g., to 4-wise independence) and obtain a multi-pass algorithm.

Lemma 3.5. *For integers $p \geq 2$ and $1 \leq t' \leq t$, let $G_1 \in \mathbb{R}^{t'\times n}$ and $G_2, \ldots, G_p \in \mathbb{R}^{t\times n}$ be independent column-normalized Gaussian matrices with 4-wise independent entries. Then for every symmetric matrix $A \in \mathbb{R}^{n\times n}$, the estimator $X = \operatorname{tr}\left(G_1 A G_2^T G_2 A \ldots G_p^T G_p A G_1^T\right)$ satisfies*

$$\mathbb{E}[X] = \operatorname{tr}(A^p) \text{ and } \operatorname{Var}(X) = O_p\left(1 + \sum_{z=2}^{\lfloor p/2 \rfloor} \frac{n^{z-1-2(z-1)/p}}{t' t^{z-1}} + \sum_{z=2}^{p} \frac{n^{z-2}}{t' t^{z-1}}\right) \|A\|_{S_p}^{2p}.$$

The proof of this lemma is postponed to Section F. It is a direct corollary of the proof of Proposition 3.4, except that t', the size of the first sketch matrix, is emphasized.

We can now use the above sketch to approximate the Schatten p-norm using $\widetilde{O}(n^{1-1/(p-1)})$ bits of space with $\lceil p/2 \rceil$ passes over the input.

Theorem 3.6. *Let $p \geq 2$ be an even integer. There is a $\lceil p/2 \rceil$-pass streaming algorithm, that on input matrix $A \in \mathbb{R}^{n\times m}$ with $n \geq m$ given as a stream, outputs an estimate X such that with probability at least 0.9, $X \in (1 \pm \epsilon)||A||_{S_p}^p$ and uses $O_p(n^{1-1/(p-1)}/\epsilon^2)$ words of space. The above extends to all integers $p \geq 2$ if A is PSD.*

The full proof is presented in Appendix A. We here sketch the proof. We take $G_1 \in \mathbb{R}^{1\times n}$ and $G_2, G_3, \ldots, G_p \in \mathbb{R}^{t\times n}$ as independent column normalized Gaussian matrix, where $t = O(n^{1-1/(p-1)})$. We then show an algorithm that computes in $\lceil p/2 \rceil$-pass the estimator $X = G_1 A G_2^T G_2 \ldots G_p A G_1^T$ and uses space at most t. As shown in Lemma 3.5, $X = \operatorname{tr}(X)$ is a unbiased estimator for Schatten p-norm with constant variance. By repeating the algorithm $O(1/\epsilon^2)$ times in parallel, we reach the desired accuracy.

3.4. Faster Update Time

Since Gaussian matrices are dense, a change to one coordinate of the input matrix A may lead to a change of every entry in the sketch. This means long update times for a streaming algorithm based on the sketch. In this section we extend our result for Gaussian sketching matrices to a distribution over $\{-1, 0, 1\}$ valued matrices with only one non-zero entry per column. The new sketch can be used to improve the update time of algorithms in the last two sections.

Definition 3.7 (Sparse ZD-sketch)**.** *Let $\mathcal{D}_{t,n}$ be the distribution over matrices $G := ZD \in \mathbb{R}^{t\times n}$, where $Z = (z_1, z_2, \ldots, z_n) \in \mathbb{R}^{t\times n}$ and $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$ are generated as follows. Let $h : [n] \to [t]$ be a 4-wise independent hash function, and set $Z_{i,j} = \mathbb{1}_{\{i=h(j)\}}$, i.e., in each z_j only the $h(j)$-th coordinate is set to 1, and all other coordinates are 0. The diagonal entries of D are four-wise independent uniform $\{-1, 1\}$ random variables, and D is independent from Z.*

Notice that each column of G has a single non-zero entry, which is actually a random sign, and the n columns are four-wise independent. This random matrix G is similar to the sketching matrix used in (Thorup & Zhang, 2004) to speed up the update time when estimating the second frequency moment of a vector in $\mathbb{R}^n$. Also note that the ZD-sketch is a version of sparse JL matrices (see e.g., (Kane & Nelson, 2014; Dasgupta et al., 2010)). In this paper we do not aim at optimizing the sparsity as we focus on approximating Schatten norms.

It is fairly easy to show that ZD-sketch works for approximating Schatten p-norm of matrices with all entries non-negative. The proof is presented in Section G. We now show that the conclusion of Theorem 3.3 and Theorem 3.6

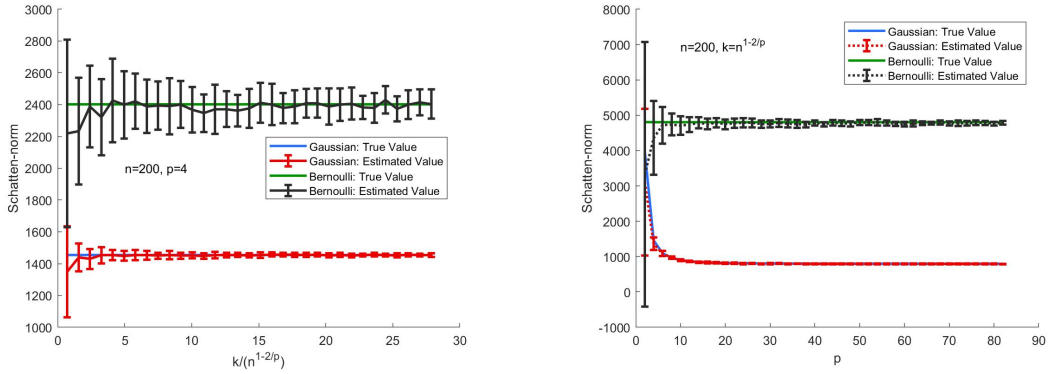

Figure 1. Accuracy of our estimator: accuracy as a function of the compression size (left) and of p (right).

still hold if we replace the Gaussian matrices in the sketch with independent samples from the sparse ZD-sketch. A major difficulty that arises in replacing the Gaussian matrix with the sparse ZD-sketch is the latter's lack of rotational invariance. To prove Theorem 3.3 we were able to expand X^2 in terms of the eigenvalues of A and compute the expectation term-by-term, but this is not possible for the sparse ZD-sketch. For example, let G be a Gaussian matrix. For any orthogonal matrix U, the matrix GU is again a Gaussian matrix with an identical distribution to G. This does not hold for sparse ZD-sketch. As a consequence, in the expansion of $\mathbb{E}(X^2)$ in the proof of Proposition 3.4, the non-zero terms would also include those monomials of odd powers of $\lambda_i(A)$. For example, for the Schatten 3-norm, one cannot bound $\sum_{i_1,i_2,\ldots i_6\in[n]}\prod_{j=1}^{6}\lambda_{i_j}$ by $O(\|A\|_{S_3}^6)$. But this term appears in the expansion of $\mathbb{E}(X^2)$ of the Schatten 3-norm estimator if using the sparse ZD-sketch matrices.

To resolve this problem, we use a technique similar to the proof of the Hanson-Wright Inequality in (Rudelson & Vershynin, 2013) to bound the variance of X. The proof is composed of three major steps. The first step is to decouple the dependent summands by injecting independence. The second step is to replace the independent random vectors with fully independent Gaussian vectors while preserving the variance. We can then apply our techniques for Gaussians to bound the variance of the final random variable. The case $p = 1$ is useful to illustrate the technique, even though Schatten 1-norm approximation can be easily accomplished in other ways. Let $G \in \mathbb{R}^{t\times n}$ be the sparse JLT matrix and let $A \in \mathbb{R}^{n\times n}$ be PSD. The sketch is GAG^T and

$$\mathrm{tr}(GAG^T) - \mathrm{tr}(A) = \sum_{i\neq j} a_{i,j}\langle g_i, g_j\rangle. \qquad (2)$$

Since $i \neq j$, g_i and g_j are independent. However the summands are subtly dependent. We first decouple the summand by choosing $\delta_i \sim$Bernoulli$(1/2)$, and write $\langle g_i, g_j\rangle = 4\,\mathbb{E}(\delta_i(1-\delta_j)\langle g_i, g_j\rangle)$. Let $V = \{i : \delta_i = 1\}$, then $\sum_{i\neq j} a_{i,j}\langle g_i, g_j\rangle = 4\,\mathbb{E}_\delta \sum_{i\in V, j\in\bar{V}} a_{i,j}\langle g_i, g_j\rangle$. Thus conditioning on δ and $\{g_j : j \in \bar{V}\}$, the set $\{\langle g_i, \sum_{j\in\bar{V}} a_{i,j}g_j\rangle : i \in V\}$ is a set of independent random variables. We can match these random variables with Gaussian random variables of the same variance, and thus replace g_i with independent Gaussian vectors. The same process can be repeated for $g_j : j \in \bar{V}$, and replace every vector $g_i : i \in [n]$ by independent Gaussian vectors. This lets us apply similar techniques as used in the proof of Proposition 3.4 to bound the variance of the resulting random variable, and thus bound the variance of the original random variable $\mathrm{tr}(GAG^T) - \mathrm{tr}(A)$.

The analogue of (2) for the case of our general estimator, $X - \mathrm{tr}(A^p)$, is much more complicated than the $p = 1$ case. We observe that these terms can be grouped as a sum of products of consecutive *walks*, i.e., $a_{i_1,i_2}a_{i_2,i_3}\ldots a_{i_z,j_{z+1}}\langle g_{j_{z+1}}^{(z+1)}, g_{i_{z+1}}^{(z+1)}\rangle$ for some z. Notice that $\langle g_{j'}^{(z')}, g_{j'}^{(z')}\rangle = 1$ for any j' and z'. For each walk, we can apply a similar idea to replace the g_i vectors with independent Gaussian vectors. Again, we apply similar techniques as used in the proof of Proposition 3.4 to bound the variance of each group. As a result, when replacing the Gaussian matrices by sparse JLT matrices, Lemma 3.5 still holds.

Using the sparse ZD-sketch, we are able to achieve the same space bound and query time as in Theorem 3.3 and Theorem 3.6. But our update time is improved to $O(1/\epsilon^2)$. We present the full statement of our theorem below. The full proof can be found in (Braverman et al., 2016).

Theorem 3.8. *For every $0 < \epsilon < 1/2$ and integer $p \geq 2$, there is a randomized one-pass streaming algorithm $\mathcal{A}$ with space requirement $O(n^{2-4/p}/\epsilon^2)$, that given as input a PSD matrix $A \in \mathbb{R}^{n\times n}$, outputs with high probability a $(1+\epsilon)$-approximation of $||A||_{S_p}^p$. The algorithm processes an update in time $O(1/\epsilon^2)$, and computes the output (after the updates) in time $O(n^{(1-2/p)\omega})/\epsilon^2$, where $\omega < 3$ is the matrix multiplication constant.*

There is similarly a randomized $\lceil p/2 \rceil$-pass streaming algorithm $\mathcal{B}$ with space requirement $O(n^{1-1/(p-1)}/\epsilon^2)$, update time in a pass $O(1/\epsilon^2)$, and output time $O(n^{(1-2/p)}/\epsilon^2)$.

For even $p \geq 2$, both algorithms extend to general input $A \in \mathbb{R}^{n \times m}$ with $m \leq n$.

4. Lower Bound For PSD Matrices

In this section we show the lower bounds for sketching Schatten-norms for PSD matrices. This lower bounds suggest that our upper bound is nearly tight. The proof is presented in Section B.

Theorem 4.1. *Suppose that $p > 0$ and $X \in \mathbb{R}^{n \times n}$ is a PSD matrix given in the entry-wise streaming model.*

(a) When $p \in \mathbb{Z}$, there is $c = c(p) > 0$ such that every one-pass streaming algorithm that $(1+c)$-approximates $\|X\|_{S_p}$ with probability $2/3$ must use $\Omega_p(n^{1-2/p})$ bits of space for even p, and $\Omega_p(n^{1-2/(p-1)})$ bits of space for odd p.

(b) When $p \notin \mathbb{Z}$, for every integer $t \geq 2$, there is $c = c(p,t) > 0$ such that every one-pass streaming algorithm that $(1+c)$-approximates $\|X\|_{S_p}$ with probability $2/3$ must use $\Omega_{p,t}(n^{1-1/t})$ bits of space.

We remark that all lower bounds in Theorem 4.1 even hold for sparse matrices, since the hard instances are sparse. The lower bounds for non-integers p and even integers p are strengthenings of the same lower bounds in (Li & Woodruff, 2016a), and are almost tight and tight up to poly-logarithmic factors, respectively.

5. Experiments

In this section we show numerical experiments that illustrate the performance of our Schatten-norm estimator described in Section 3. We consider two sets of synthetic inputs, which roughly represent the extreme cases for all inputs. One is a matrix drawn from a standard Gaussian distribution, i.e., $A = GG^\top$, where each entry of $G \in \mathbb{R}^{n \times n}$ is an independent $\mathcal{N}(0,1)$ random variable. The other is a matrix drawn from a Bernoulli distribution, i.e., $A = BB^\top$, where each entry of $B \in \mathbb{R}^{n \times n}$ is an independent $\mathcal{B}(0.5)$ random variable. We chose $n = 200$ in both cases. We construct our estimator using the native pseudo-random generator in matlab as our hash function. We measure the error of our estimator when varying the hidden constant in our choice of k (recall that our sketching matrix is of size $k \times n$ for $k = O(n^{1-2/p})$) and varying p. These results are presented in Figure 1. We then compared the update time of our sparse estimator with the estimators described in (Li & Woodruff, 2016a) and (Kong & Valiant, 2016) that are based on a dense Gaussian distribution. The result is shown in Figure 2. Our estimators are comparably or a little less accurate than theirs but are two orders of magnitude faster in terms of update time. In Figure 2, we used our ZD sketch from Definition 3.7. Since each update only updates a single entry to the matrix, the update time is almost 0. On the other hand, the dense Gaussian sketch requires at least $\Theta(n^{2-4/p})$ operations.

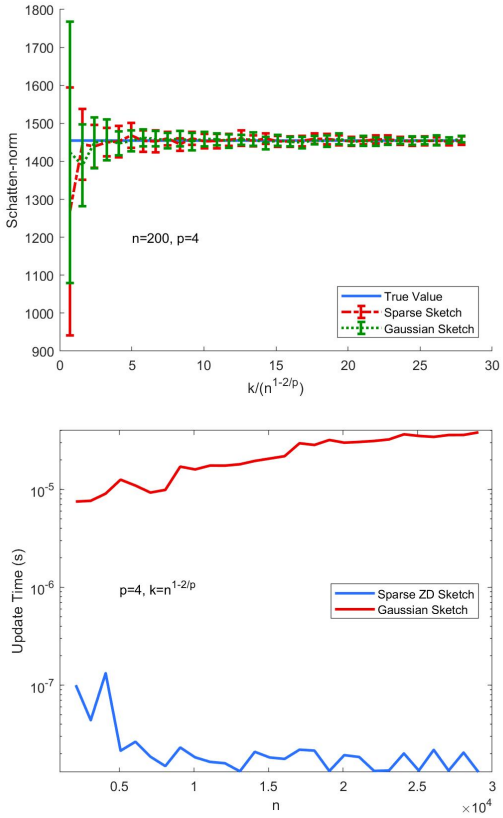

Figure 2. Comparison with Gaussian-based estimator. Top: accuracy comparison. Bottom: update time comparison.

Acknowledgment

This material is based upon work supported in part by the National Science Foundation under Grants No. 1447639, 1650041, 1652257 and CCF-1815840, the ONR Award N00014-18-1-2364, the Israel Science Foundation grant #897/13 and by a Minerva Foundation grant.

References

Andoni, Alexandr and Nguyên, Huy. Eigenvalues of a matrix in the streaming model. In *24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1729–1737. SIAM, 2013. doi: 10.1137/1.9781611973105.124.

Andoni, Alexandr, Krauthgamer, Robert, and Razenshteyn, Ilya. Sketching and embedding are equivalent for norms. In *47th Annual ACM Symposium on Theory of Computing*, pp. 479–488. ACM, 2015. ISBN 978-1-4503-3536-2. doi: 10.1145/2746539.2746552.

Andrews, George E., Askey, Richard, and Roy, Ranjan. *Special Functions*. Cambridge University Press, 1999.

Bar-Yossef, Ziv, Jayram, Thathachar S, Kumar, Ravi, and Sivakumar, D. An information statistics approach to data stream and communication complexity. In *43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 209–218. IEEE, 2002. doi: 10.1109/SFCS.2002.1181944.

Braverman, Vladimir, Chestnut, Stephen R., Krauthgamer, Robert, Li, Yi, Woodruff, David P., and Yang, Lin F. Matrix norms in data streams: Faster, multi-pass and row-order. `arXiv:1609.05885 [cs.DS]`, 2016.

Dasgupta, Anirban, Kumar, Ravi, and Sarlós, Tamás. A sparse johnson: Lindenstrauss transform. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 341–350. ACM, 2010.

Di Napoli, Edoardo, Polizzi, Eric, and Saad, Yousef. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692, 2016.

Efraimidis, Pavlos S. and Spirakis, Paul G. Weighted random sampling with a reservoir. *Information Processing Letters*, 97(5):181 – 185, 2006.

Gronemeier, Andre. Asymptotically Optimal Lower Bounds on the NIH-Multi-Party Information Complexity of the AND-Function and Disjointness. In *26th International Symposium on Theoretical Aspects of Computer Science*, volume 3, pp. 505–516, Dagstuhl, Germany, 2009. doi: 10.4230/LIPIcs.STACS.2009.1846.

Huang, Hao and Kasiviswanathan, Shiva Prasad. Streaming anomaly detection using randomized matrix sketching. *Proc. VLDB Endow.*, 9(3):192–203, November 2015. ISSN 2150-8097. doi: 10.14778/2850583.2850593.

Indyk, Piotr and Motwani, Rajeev. Approximate nearest neighbors: towards removing the curse of dimensionality. In *30th Annual ACM Symposium on Theory of Computing*, pp. 604–613. ACM, 1998. doi: 10.1145/276698.276876.

Jayram, Thathachar S and Woodruff, David P. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with subconstant error. *ACM Transactions on Algorithms*, 9(3):26, 2013. doi: 10.1145/2483699.2483706.

Kane, Daniel, Meka, Raghu, and Nelson, Jelani. Almost optimal explicit Johnson-Lindenstrauss families. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 628–639. Springer, 2011. doi: 10.1007/978-3-642-22935-0_53.

Kane, Daniel M and Nelson, Jelani. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):4, 2014. doi: 10.1145/2559902.

Kapralov, Michael, Potluru, Vamsi, and Woodruff, David. How to fake multiply by a gaussian matrix. In *International Conference on Machine Learning*, pp. 2101–2110, 2016.

Khetan, Ashish and Oh, Sewoong. Matrix norm estimation from a few entries. *NIPS*, 2017.

Kogan, Dmitry and Krauthgamer, Robert. Sketching cuts in graphs and hypergraphs. In *Conference on Innovations in Theoretical Computer Science*, pp. 367–376. ACM, 2015. doi: 10.1145/2688073.2688093.

Kong, Weihao and Valiant, Gregory. Spectrum estimation from samples. *arXiv preprint arXiv:1602.00061*, 2016.

Le Gall, François. Powers of tensors and fast matrix multiplication. In *39th International Symposium on Symbolic and Algebraic Computation*, pp. 296–303. ACM, 2014. doi: 10.1145/2608628.2608664.

Li, Yi and Woodruff, David P. On approximating functions of the singular values in a stream. In *48th Annual ACM Symposium on Theory of Computing*, pp. 726–739. ACM, 2016a. ISBN 978-1-4503-4132-5. doi: 10.1145/2897518.2897581.

Li, Yi and Woodruff, David P. Tight bounds for sketching the operator norm, Schatten norms, and subspace embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 60 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 39:1–39:11. Schloss Dagstuhl,

2016b. ISBN 978-3-95977-018-7. doi: 10.4230/LIPIcs.APPROX-RANDOM.2016.39.

Li, Yi and Woodruff, David P. Embeddings of Schatten Norms with Applications to Data Streams. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 60:1–60:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. ISBN 978-3-95977-041-5. doi: 10.4230/LIPIcs.ICALP.2017.60.

Li, Yi, Nguyen, Huy L., and Woodruff, David P. On sketching matrix norms and the top singular vector. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pp. 1562–1581. SIAM, 2014. ISBN 978-1-611973-38-9. doi: 10.1137/1.9781611973402.114.

Liberty, Edo. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 581–588. ACM, 2013. doi: 10.1145/2487575.2487623.

Rudelson, Mark and Vershynin, Roman. Hanson-Wright inequality and sub-gaussian concentration. *Electron. Commun. Probab*, 18(82):1–9, 2013. doi: 10.1214/ECP.v18-2865.

Thorup, Mikkel and Zhang, Yin. Tabulation based 4-universal hashing with applications to second moment estimation. In *SODA*, volume 4, pp. 615–624, 2004.

Verbin, Elad and Yu, Wei. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the 22nd ACM-SIAM SODA*, pp. 11–25, 2011. doi: 10.1137/1.9781611973082.2.

Wei, Zhewei, Liu, Xuancheng, Li, Feifei, Shang, Shuo, Du, Xiaoyong, and Wen, Ji-Rong. Matrix sketching over sliding windows. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pp. 1465–1480. ACM, 2016. ISBN 978-1-4503-3531-7. doi: 10.1145/2882903.2915228.

Woodruff, David P. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10:1–157, 2014. doi: 10.1561/0400000060.

Zhang, Yuchen, Wainwright, Martin, and Jordan, Michael. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 457–465, 2015.