Near-Optimal Entrywise Sampling of Numerically Sparse Matrices*

Vladimir Braverman† VOVA@CS.JHU.EDU
Johns Hopkins University

Robert Krauthgamer‡ ROBERT.KRAUTHGAMER@WEIZMANN.AC.IL
Weizmann Institute of Science

Aditya Krishnan§ AKRISH23@JHU.EDU
Johns Hopkins University

Shay Sapir SHAY.SAPIR@WEIZMANN.AC.IL
Weizmann Institute of Science

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

Many real-world data sets are sparse or almost sparse. One method to measure this for a matrix $A \in \mathbb{R}^{n\times n}$ is the *numerical sparsity*, denoted $\mathsf{ns}(A)$, defined as the minimum $k \geq 1$ such that $\|a\|_1/\|a\|_2 \leq \sqrt{k}$ for every row and every column a of A. This measure of a is smooth and is clearly only smaller than the number of non-zeros in the row/column a.

The seminal work of Achlioptas and McSherry (2007) has put forward the question of approximating an input matrix A by entrywise sampling. More precisely, the goal is to quickly compute a sparse matrix $\tilde{A}$ satisfying $\|A - \tilde{A}\|_2 \leq \epsilon\|A\|_2$ (i.e., additive spectral approximation) given an error parameter $\epsilon > 0$. The known schemes sample and rescale a small fraction of entries from A.

We propose a scheme that sparsifies an almost-sparse matrix A — it produces a matrix $\tilde{A}$ with $O(\epsilon^{-2}\mathsf{ns}(A) \cdot n \ln n)$ non-zero entries with high probability. We also prove that this upper bound on $\mathsf{nnz}(\tilde{A})$ is *tight* up to logarithmic factors. Moreover, our upper bound improves when the spectrum of A decays quickly (roughly replacing n with the stable rank of A). Our scheme can be implemented in time $O(\mathsf{nnz}(A))$ when $\|A\|_2$ is given. Previously, a similar upper bound was obtained by Achlioptas et al. (2013) but only for a restricted class of inputs that does not even include symmetric or covariance matrices. Finally, we demonstrate two applications of these sampling techniques, to faster approximate matrix multiplication, and to ridge regression by using sparse preconditioners.

1. Introduction

Matrices for various tasks in machine learning and data science often contain millions or even billions of dimensions. At the same time, they often possess structure that can be exploited to design

* A full version is available at arXiv:2011.01777.

† This research was supported in part by NSF CAREER grant 1652257, NSF grant 1934979, ONR Award N00014-18-1-2364 and the Lifelong Learning Machines program from DARPA/MTO.

‡ Work partially supported by ONR Award N00014-18-1-2364, the Israel Science Foundation grant #1086/18, and a Minerva Foundation grant.

§ This research was supported in part by NSF CAREER grant 1652257, ONR Award N00014-18-1-2364 and the Lifelong Learning Machines program from DARPA/MTO.

more efficient algorithms. Sparsity in the rows and/or columns of the matrix is one such phenomenon for which many computational tasks on matrices admit faster algorithms, e.g., low-rank approximation (Ghashami et al., 2016; Huang, 2019), regression problems (Johnson and Zhang, 2013) and semi-definite programming (d'Aspremont, 2011; Arora et al., 2005). Sparsity, however, is not a numerically smooth quantity. Specifically, for a vector $x \in \mathbb{R}^n$ to be k-sparse, at least $n-k$ entries of x must be 0. In practice, many entries could be small but non-zero, e.g. due to noise, and thus the vector would be considered dense.

A smooth analogue of sparsity for a matrix $A \in \mathbb{R}^{m\times n}$ can be defined as follows. First, for a row (or column) vector $a \in \mathbb{R}^n$, define its *numerical sparsity* (Lopes, 2013; Gupta and Sidford, 2018) to be

$$\mathsf{ns}(a) \coloneqq \min\{k \geq 0 : \|a\|_1 \leq \sqrt{k}\|a\|_2\}. \tag{1}$$

This value is clearly at most the number of non-zeros in a, denoted $\|a\|_0$, but can be much smaller. Earlier work used variants of this quantity, referring to $\mathsf{ns}(a)$ as the ℓ_1/ℓ_2-sparsity of the vector (Hoyer, 2004; Hurley and Rickard, 2009). We further define the numerical sparsity of a matrix A, denoted $\mathsf{ns}(A)$, to be the maximum numerical sparsity of any of its rows and columns.

In order to take advantage of sparse matrices in various computational tasks, a natural goal is to approximate a matrix A with numerical sparsity $\mathsf{ns}(A)$ with another matrix $\tilde{A}$ of the same dimensions, that is k-sparse for $k = O(\mathsf{ns}(A))$ (i.e., every row and column is k-sparse). The seminal work of Achlioptas and McSherry (2007) introduced a framework for matrix sparsification via entrywise sampling for approximating the matrix A in spectral-norm. Specifically, they compute a sparse matrix $\tilde{A}$ by sampling and rescaling a small fraction of entries from A such that with high probability $\|A-\tilde{A}\|_2 \leq \epsilon\|A\|_2$ for some error parameter $\epsilon > 0$, where $\|\cdot\|_2$ denotes the spectral-norm. This motivates the following definition.

Definition 1 *An* ϵ-spectral-norm approximation *for* $A \in \mathbb{R}^{m\times n}$ *is a matrix* $\tilde{A} \in \mathbb{R}^{m\times n}$ *satisfying*

$$\|\tilde{A} - A\|_2 \leq \epsilon\|A\|_2. \tag{2}$$

When $\tilde{A}$ *is obtained by sampling and rescaling entries from* A*, we call it an* ϵ-spectral-norm sparsifier.

Before we continue, let us introduce necessary notations. Here and throughout, we denote the number of non-zero entries in a matrix A by $\mathsf{nnz}(A)$, the Frobenius-norm of A by $\|A\|_F$, the stable-rank of A by $\mathsf{sr}(A) \coloneqq \|A\|_F^2/\|A\|_2^2$, the i-th row and the j-th column of A by A_i and A^j, respectively, and the row-sparsity and column-sparsity of A by $\mathsf{rsp}(A) \coloneqq \max_i \|A_i\|_0$ and $\mathsf{csp}(A) \coloneqq \max_j \|A^j\|_0$, respectively.

The framework of Achlioptas and McSherry (2007) can be used as a preprocessing step that "sparsifies" numerically sparse matrices in order to speed up downstream tasks. It thus motivated a line of work on sampling schemes (Arora et al., 2006; Gittens and Tropp, 2009; Drineas and Zouzias, 2011; Nguyen et al., 2015; Achlioptas et al., 2013; Kundu and Drineas, 2014; Kundu et al., 2017), in which the output $\tilde{A}$ is an unbiased estimator of A, and the sampling distributions are simple functions of A and hence can be computed easily, say, in nearly $O(\mathsf{nnz}(A))$-time and with one or two passes over the matrix. Under these constraints, the goal is simply to minimize the sparsity of the ϵ-spectral-norm sparsifier $\tilde{A}$.

The latest work, by Achlioptas et al. (2013), provides a bound for a restricted class of "data matrices". Specifically, they look at matrices $A \in \mathbb{R}^{m\times n}$ such that $\min_i \|A_i\|_1 \geq \max_j \|A^j\|_1$,

which can be a reasonable assumption when $m \ll n$. This restricted class does not include the class of square matrices, and hence does not include symmetric matrices such as covariance matrices. Hence, an important question is whether their results extend to a larger class of matrices. Our main result, described in the next section, resolves this concern in the affirmative.

1.1. Main Results

We generalize the sparsity bound of Achlioptas et al. (2013), which is the best currently known, to all matrices $A \in \mathbb{R}^{m\times n}$. Our main result is a sampling scheme to compute an ϵ-spectral-norm sparsifier for numerically sparse matrices A, as follows.

Theorem 2 *There is an algorithm that, given a matrix $A \in \mathbb{R}^{m\times n}$ and a parameter $\epsilon > 0$, where $m \geq n$, computes with high probability an ϵ-spectral-norm sparsifier $\tilde{A}$ for A with expected sparsity*

$$\mathbb{E}(\mathsf{nnz}(\tilde{A})) = O\left(\epsilon^{-2}\,\mathsf{ns}(A)\,\mathsf{sr}(A)\log m + \epsilon^{-1}\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)n}\log m\right).$$

Moreover, it runs in $O(\mathsf{nnz}(A))$-time when a constant factor estimate of $\|A\|_2$ is given.[1]

We obtain this result by improving the main technique of Achlioptas et al. (2013). Their sampling distribution arises from optimizing a concentration bound, called the matrix-Bernstein inequality, for the sum of matrices formed by sampling entries independently. Our distribution is obtained by the same approach, but arises from considering the columns and rows simultaneously.

In addition to the sampling scheme in Theorem 2, we analyze ℓ_1-sampling from every row (in Section 2.1).[2] This gives a worse bound than the above bound, roughly replacing the $\mathsf{sr}(A)$ term with n, but has the added advantage that the sampled matrix has uniform row-sparsity.

Lower Bound. Our next theorem complements our main result with a lower bound on the sparsity of *any* ϵ-spectral-norm approximation of a matrix A in terms of its numerical sparsity $\mathsf{ns}(A)$ and error parameter $\epsilon > 0$.[3]

Theorem 3 *Let $0 < \epsilon < \frac{1}{2}$ and $n, k \geq 1$ be parameters satisfying $k \leq O(\epsilon^2 n \log^2 \frac{1}{\epsilon})$. Then, there exists a matrix $A \in \mathbb{R}^{n\times n}$ such that $\mathsf{ns}(A) = \Theta(k \log^2 \frac{1}{\epsilon})$ and, for every matrix B satisfying $\|A - B\|_2 \leq \epsilon\|A\|_2$, the sparsity of every row and every column of B is at least $\Omega(\epsilon^{-2}k\log^{-2}\frac{1}{\epsilon}) = \tilde{\Omega}(\epsilon^{-2}) \cdot \mathsf{ns}(A)$.*

While the lower bound shows that the worst-case dependence on the parameters $\mathsf{ns}(A)$ and ϵ is optimal, it is based on a matrix with stable rank $\Omega(n)$. Settling the sample complexity when the stable rank is $o(n)$ is an interesting open question that we leave for future work.

[1] A constant factor estimate of $\|A\|_2$ can be computed in $\tilde{O}(\mathsf{nnz}(A))$-time by the power method.

[2] Sampling entry A_{ij} with probability proportional to $|A_{ij}|/\|A_i\|_1$

[3] We write $\tilde{O}(f)$ as a shorthand for $O(f \cdot \mathrm{polylog}(nm))$ where n and m are the dimensions of the matrix, and write $O_\epsilon(\cdot)$ when the hidden constant may depend on ϵ.

1.2. Comparison to Previous Work

The work of Achlioptas and McSherry (2007) initiated a long line of work on entrywise sampling schemes that approximate a matrix under spectral-norm (Arora et al., 2006; Gittens and Tropp, 2009; Drineas and Zouzias, 2011; Kundu and Drineas, 2014; Kundu et al., 2017; Nguyen et al., 2015; Achlioptas et al., 2013). Sampling entries independently has the advantage that the output matrix can be seen as a sum of independent random matrices whose spectral-norm can be bounded using known matrix concentration bounds. All previous work uses such matrix concentration bounds with the exception of Arora et al. (2006) who bound the spectral-norm of the resulting matrix by analyzing the Rayleigh quotient of all possible vectors.

Natural distributions to sample entries are the ℓ_2 and ℓ_1 distributions, which correspond to sampling entry A_{ij} with probability proportional to $A_{ij}^2/\|A\|_F^2$ and $|A_{ij}|/\|A\|_1$ respectively.[4]

Prior work that use variants of the ℓ_2 sampling (Achlioptas and McSherry, 2007; Drineas and Zouzias, 2011; Nguyen et al., 2015; Kundu and Drineas, 2014) point out that sampling according to the ℓ_2 distribution causes small entries to "blow-up" when sampled. Some works, e.g. Drineas and Zouzias (2011), get around this by zeroing-out small entries or by exceptional handling of small entries, e.g. Achlioptas and McSherry (2007), while others used distributions that combine the ℓ_1 and ℓ_2 distributions, e.g. Kundu and Drineas (2014). All these works sample $\Omega(\epsilon^{-2} n\, \mathsf{sr}(A))$ entries in expectation to achieve an ϵ-spectral-norm approximation and our Theorem 2 provides an asymptotically better bound. For a full comparison see Table 1.

All these algorithms, including the algorithm of Theorem 2, sample a number of entries corresponding to $\mathsf{sr}(A)$, hence they must have an estimate of it, which requires estimating $\|A\|_2$. An exception is the bound in Theorem 8, which can be achieved without this estimate. In practice, however, and in previous work in this area, there is a sampling budget $s \geq 0$ and s samples are drawn according to the stated distribution, avoiding the need for this estimate. In this case, the algorithm of Theorem 2 can be implemented in two-passes over the data and in $O(\mathsf{nnz}(A))$ time.

1.3. Applications of Spectral-Norm Sparsification

We provide two useful applications of spectral-norm sparsification. More precisely, we use the sparsification to speed up two computational tasks on numerically sparse matrices: approximate matrix multiplication and approximate ridge regression. This adds to previous work, which showed applications to low-rank approximation (Achlioptas and McSherry, 2007), to semidefinite programming (Arora et al., 2006), and to PCA and sparse PCA (Kundu et al., 2017). These applications work in a black-box manner, and can thus employ our improved sparsification scheme.

Application I: Approximate Matrix Multiplication (AMM). Given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ and error parameter $\epsilon > 0$, the goal is to compute a matrix $C \in \mathbb{R}^{m\times p}$ such that $\|AB - C\| \leq \epsilon\|A\|\cdot\|B\|$, where the norm is usually either Frobenius-norm $\|\cdot\|_F$ or spectral-norm $\|\cdot\|_2$. In Section 3, we provide algorithms for both error regimes by combining our entrywise sampling scheme with previous AMM algorithms that sample a small number of columns of A and rows of B.

Theorem 4 *There exists an algorithm that, given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ parameter $0 < \epsilon < \frac{1}{2}$ and constant factor estimates of $\|A\|_2$ and $\|B\|_2$, computes a matrix $C \in \mathbb{R}^{m\times p}$*

[4] Here and henceforth we denote by $\|A\|_1$ the entry-wise l_1 norm.

Table 1: Comparison between schemes for ϵ-spectral-norm sparsification. The first two entries in the third column present the ratio between the referenced sparsity and that of Theorem 2.

Expected Number of Samples	**Reference**	**Compared to Thm. 2**
$O(\epsilon^{-1} n \sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)})$	Arora et al. (2006)	$\tilde{O}_\epsilon\left(\min\left(\frac{n}{\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)}}, \sqrt{n}\right)\right)$
$O(\epsilon^{-2} n\,\mathsf{sr}(A) + n\,\mathrm{polylog}(n))$	Achlioptas and McSherry (2007)	$\tilde{O}_\epsilon\left(\min\left(\frac{n}{\mathsf{ns}(A)}, \sqrt{\frac{n\,\mathsf{sr}(A)}{\mathsf{ns}(A)}}\right)\right)$
$\tilde{O}(\epsilon^{-2} n\,\mathsf{sr}(A))$	Drineas and Zouzias (2011); Kundu and Drineas (2014)	
$\tilde{O}(\epsilon^{-2}\,\mathsf{ns}(A)\,\mathsf{sr}(A) + \epsilon^{-1}\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A) n})$	Achlioptas et al. (2013); Theorem 2	Achlioptas et al. (2013) is only for data matrices
$\tilde{O}(\epsilon^{-2} n\,\mathsf{ns}(A))$	Theorem 8	bounded row-sparsity
$\Omega(\epsilon^{-2} n\,\mathsf{ns}(A)\log^{-4}\frac{1}{\epsilon})$	Theorem 3	$\mathsf{sr}(A) = \Theta(n)$

satisfying with high probability $\|AB - C\|_2 \le \epsilon\|A\|_2\|B\|_2$ *in time*

$$O(\mathsf{nnz}(A) + \mathsf{nnz}(B)) + \tilde{O}(\epsilon^{-6}\sqrt{\mathsf{sr}(A)\,\mathsf{sr}(B)}\,\mathsf{ns}(A)\,\mathsf{ns}(B)).$$

Theorem 5 *There exists an algorithm that, given matrices* $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ *and parameter* $0 < \epsilon < \frac{1}{2}$*, computes a matrix* $C \in \mathbb{R}^{m\times p}$ *satisfying* $\mathbb{E}\,\|AB - C\|_F \le \epsilon\|A\|_F\|B\|_F$ *in time*

$$O(\mathsf{nnz}(A) + \mathsf{nnz}(B) + \epsilon^{-6}\,\mathsf{ns}(A)\,\mathsf{ns}(B)).$$

Approximate Matrix Multiplication (AMM) is a fundamental problem in numerical linear algebra with a long line of formative work (Frieze et al., 2004; Drineas et al., 2006; Clarkson and Woodruff, 2009; Magen and Zouzias, 2011; Cohen et al., 2016; Ye et al., 2016; Mroueh et al., 2017) and many others. These results fall into roughly three categories; sampling based methods, random projection based methods and a mixture of sampling and projection based methods. We focus on sampling based methods in our work.

There are two main error regimes considered in the literature: spectral-norm error and Frobenius-norm error. We focus on the results of Magen and Zouzias (2011) for spectral-norm error and Drineas et al. (2006) for Frobenius-norm error. Sampling based methods, including that of Drineas et al. (2006); Magen and Zouzias (2011), propose sampling schemes that are linear time or nearly-linear time: specifically, they write the product of two matrices as the sum of n outer products $AB = \sum_{i\in[n]} A^i B_i$, and then sample and compute each outer product $A^i B_i/p_i$ with probability $p_i \propto \|A^i\|_2\|B_i\|_2$. Computing each of these rank-1 outer products takes time bounded by $O(\mathsf{csp}(A)\,\mathsf{rsp}(B))$. This estimator is repeated sufficiently many times depending on the error regime under consideration.

Our entrywise-sampling scheme compounds well with this framework for approximate matrix multiplication by additionally sampling entries from the rows/columns sampled by the AMM algorithm. We essentially replace the $\mathsf{csp}(A)\,\mathsf{rsp}(B)$ term with $\mathsf{ns}(A)\,\mathsf{ns}(B)$, up to $\tilde{O}(\mathrm{poly}(1/\epsilon))$ factors, for both Frobenius-norm and spectral-norm error regimes. It is plausible that the dependence on epsilon can be improved.

Application II: Approximate Ridge Regression. Given a matrix $A \in \mathbb{R}^{m\times n}$, a vector $b \in \mathbb{R}^m$ and a parameter $\lambda > 0$, the goal is to find a vector $x \in \mathbb{R}^n$ that minimizes $\|Ax - b\|_2^2 + \lambda\|x\|_2^2$. This problem is λ-strongly convex, has solution $x^* = (A^\top A + \lambda I)^{-1}A^\top b$ and condition number $\kappa_\lambda(A^\top A) \coloneqq \|A\|_2^2/\lambda$.

Given an initial vector $x_0 \in \mathbb{R}^n$ and a parameter $\epsilon > 0$, an ϵ-approximate solution to the ridge regression problem is a vector $\hat{x} \in \mathbb{R}^n$ satisfying $\|\hat{x} - x^*\|_{A^\top A+\lambda I} \le \epsilon\|x_0 - x^*\|_{A^\top A+\lambda I}$, where we write $\|x\|_M \coloneqq x^\top Mx$ when M is a PSD matrix. We provide algorithms for approximate ridge regression by using our sparsification scheme as a preconditioner for known linear-system solvers in composition with a black-box acceleration framework by Frostig et al. (2015). The following theorem is proved in the full version.

Theorem 6 *There exists an algorithm that, given $A \in \mathbb{R}^{m\times n}$, $x_0 \in \mathbb{R}^n$, $\lambda > 0$ and $\epsilon > 0$, computes with high probability an ϵ-approximate solution to the ridge regression problem in time*

$$O_\epsilon(\mathsf{nnz}(A)) + \tilde{O}_\epsilon\left((\mathsf{nnz}(A))^{2/3}(\mathsf{ns}(A)\,\mathsf{sr}(A))^{1/3}\sqrt{\kappa_\lambda(A^\top A)}\right).$$

Moreover, when the input matrix A has uniform column (or row) norms, the running time in Theorem 6 can be reduced by a factor of roughly $(\mathsf{sr}(A)/n)^{1/6}$, for details see the full version.

Solving linear systems using preconditioning has a rich history that is beyond the scope of this work to summarize. Recently, the work of Gupta and Sidford (2018) designed algorithms with improved running times over popular methods using the Stochastic Variance Reduced Gradient Descent (SVRG) framework of Johnson and Zhang (2013). They adapt it using efficient subroutines for numerically sparse matrices. They also suggested the idea of using spectral-norm sparsifiers as preconditioners for linear regression. While they considered the sparsification of Achlioptas et al. (2013) for computing the preconditioners, they required a stronger bound on the spectral-norm approximation than Theorem 6 does.

Our result is in general incomparable to that of Gupta and Sidford (2018). In the case when the input has uniform column (or row) norms, our running time is roughly an $(\mathsf{ns}(A)/n)^{1/6}$-factor smaller than theirs, for details see the full version.

Very recently, Carmon et al. (2020) have developed, independently of our work and as part of a suite of results on bilinear minimax problems, an algorithm for ridge regression with improved running time $\tilde{O}(\mathsf{nnz}(A) + \sqrt{\mathsf{nnz}(A)\,\mathsf{ns}(A)\,\mathsf{sr}(A)\kappa_\lambda(A^\top A)})$. Their approach is different and their techniques are more involved than ours.

2. Spectral-Norm Sparsification

In this section we state and prove our main results. We first prove the upper bound in Theorem 2. Then we analyze ℓ_1 sampling from the rows in Theorem 8, Section 2.1 that gives a slightly weaker bound but has the property that the resulting matrix has uniform row sparsity. In Section 2.2, we prove the lower bound in Theorem 3.

Theorem 2 *There is an algorithm that, given a matrix $A \in \mathbb{R}^{m\times n}$ and a parameter $\epsilon > 0$, where $m \geq n$, computes with high probability an ϵ-spectral-norm sparsifier $\tilde{A}$ for A with expected sparsity*

$$\mathbb{E}(\mathsf{nnz}(\tilde{A})) = O\left(\epsilon^{-2}\,\mathsf{ns}(A)\,\mathsf{sr}(A)\log m + \epsilon^{-1}\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)n}\log m\right).$$

Moreover, it runs in $O(\mathsf{nnz}(A))$-time when a constant factor estimate of $\|A\|_2$ is given.[5]

Before we prove Theorem 2, we start by stating a result on the concentration of sums of independent random matrices; the Matrix Bernstein Inequality.

Theorem 7 (Matrix Bernstein, Theorem 1.6 of Tropp (2012)) *Consider a finite sequence $\{Z_k\}$ of independent, random $d_1 \times d_2$ real matrices, such that there is $R > 0$ satisfying $\mathbb{E}\, Z_k = 0$ and $\|Z_k\|_2 \leq R$ almost surely. Define*

$$\sigma^2 = \max\left\{\left\|\sum_k \mathbb{E}(Z_k Z_k^\top)\right\|_2, \left\|\sum_k \mathbb{E}(Z_k^\top Z_k)\right\|_2\right\}.$$

Then for all $t \geq 0$,

$$\mathbb{P}\left(\left\|\sum_k Z_k\right\|_2 \geq t\right) \leq (d_1+d_2)\exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right).$$

Proof of Theorem 2. Let $\epsilon > 0$. Given a matrix A, define sampling probabilities as follows.

$$\begin{aligned}
p_{ij}^{(1)} &= \frac{|A_{ij}|}{\sum_{i'j'}|A_{i'j'}|}\\
p_{ij}^{(2)} &= \frac{\|A_i\|_1^2}{\sum_{i'}\|A_{i'}\|_1^2}\cdot\frac{|A_{ij}|}{\|A_i\|_1}\\
p_{ij}^{(3)} &= \frac{\|A^j\|_1^2}{\sum_{j'}\|A^{j'}\|_1^2}\cdot\frac{|A_{ij}|}{\|A^j\|_1}\\
p_{ij}^{*} &= \max_\alpha(p_{ij}^{(\alpha)}).
\end{aligned}$$

Observe that each $\alpha = 1, 2, 3$ yields a probability distribution because $\sum_{ij} p_{ij}^{(\alpha)} = 1$.

Let $s < mn$ be a parameter that we will choose later. Now sample each entry of A independently and scale it to get an unbiased estimator, i.e., compute $\tilde{A}$ by

$$\tilde{A}_{ij} = \begin{cases} \frac{A_{ij}}{p_{ij}} & \text{with prob. } p_{ij} = \min(1, s\cdot p_{ij}^*);\\ 0 & \text{otherwise.}\end{cases}$$

To bound the expected sparsity, observe that $p_{ij}^* \leq \sum_\alpha p_{ij}^{(\alpha)}$, and thus

$$\mathbb{E}[\mathsf{nnz}(\tilde{A})] = \sum_{ij} p_{ij} \leq s\sum_{ij}\sum_\alpha p_{ij}^{(\alpha)} \leq 3s.$$

[5] A constant factor estimate of $\|A\|_2$ can be computed in $\tilde{O}(\mathsf{nnz}(A))$-time by the power method.

We show that each of the above distributions bounds one of the terms in matrix Bernstein bound. For each pair of indices (i,j) define a matrix Z_{ij} that has a single non-zero at the (i,j) entry, with value $\tilde{A}_{ij} - A_{ij}$. Its spectral-norm is $\|Z_{ij}\|_2 = |\tilde{A}_{ij} - A_{ij}|$. If $p_{ij} = 1$, this is 0. If $p_{ij} < 1$ then

$$\begin{aligned}
|\tilde{A}_{ij} - A_{ij}| &\le |A_{ij}| \max(1, \frac{1}{p_{ij}} - 1) \\
&\le \frac{|A_{ij}|}{p_{ij}} \le \frac{|A_{ij}|}{sp_{ij}^{(1)}} = \frac{1}{s}\sum_{i'j'} |A_{i'j'}| \\
&\le \frac{\sqrt{\mathsf{ns}(A)}}{s}\sum_j \|A^j\|_2 \le \frac{\sqrt{\mathsf{ns}(A)n}}{s}\|A\|_F =: R,
\end{aligned}$$

where the last inequality follows from Cauchy-Schwarz inequality.

In order to bound σ^2, first notice that $\mathrm{var}(\tilde{A}_{ij}) \le \mathbb{E}(\tilde{A}_{ij}^2) = \frac{A_{ij}^2}{sp_{ij}^*}$. Now, since $Z_{ij}Z_{ij}^\top$ has a single non-zero entry at (i,i), and $Z_{ij}^\top Z_{ij}$ has a single non-zero entry at (j,j), both $\sum_{i,j} Z_{ij}Z_{ij}^\top$ and $\sum_{i,j} Z_{ij}^\top Z_{ij}$ are diagonal, where the (i,i) entry is $\sum_j (\tilde{A}_{ij} - A_{ij})^2$ in the former and the (j,j) entry is $\sum_i (\tilde{A}_{ij} - A_{ij})^2$ in the latter. Since these are diagonal matrices, their spectral-norm equals their largest absolute entry, and thus

$$\begin{aligned}
\Big\|\sum_{i,j}\mathbb{E}\left(Z_{ij}Z_{ij}^\top\right)\Big\|_2 &\le \max_i\Big(\sum_j \frac{A_{ij}^2}{sp_{ij}^*}\Big) \le \max_i\Big(\sum_j \frac{A_{ij}^2}{sp_{ij}^{(2)}}\Big) \\
&= \frac{1}{s}\max_i\Big(\sum_j \frac{|A_{ij}|\sum_{i'}\|A_{i'}\|_1^2}{\|A_i\|_1}\Big) = \frac{1}{s}\sum_{i'}\|A_{i'}\|_1^2 \\
&\le \frac{1}{s}\sum_{i'}\mathsf{ns}(A)\|A_{i'}\|_2^2 = \frac{\mathsf{ns}(A)}{s}\|A\|_F^2.
\end{aligned}$$

The same bound can be shown for $\sum_{i,j}\mathbb{E}(Z_{ij}^\top Z_{ij})$ by using $p_{ij}^* \ge p_{ij}^{(3)}$, thus by the definition of σ^2, $\sigma^2 \le \frac{\mathsf{ns}(A)}{s}\|A\|_F^2$. Finally, by the matrix-Bernstein bound,

$$\mathbb{P}\Big(\Big\|\sum_{i,j} Z_{ij}\Big\|_2 \ge \epsilon\|A\|_2\Big) \le 2m\exp\left(-\frac{\epsilon^2\|A\|_2^2/2}{\frac{\mathsf{ns}(A)}{s}\|A\|_F^2 + \epsilon\frac{\sqrt{\mathsf{ns}(A)n}}{s}\|A\|_F\|A\|_2/3}\right),$$

and since $\mathsf{sr}(A) = \frac{\|A\|_F^2}{\|A\|_2^2}$, by setting $s = O(\epsilon^{-2}\,\mathsf{ns}(A)\,\mathsf{sr}(A)\log m + \epsilon^{-1}\sqrt{\mathsf{ns}(A)\cdot n\cdot\mathsf{sr}(A)}\log m)$ we conclude that with high probability $\|\tilde{A} - A\|_2 \le \epsilon\|A\|_2$, which completes the proof of Theorem 2. ∎

2.1. A Second Sampling Scheme

We analyze ℓ_1 row sampling, i.e. sampling entry (i,j) with probability $\frac{|A_{ij}|}{\|A_i\|_1}$, as was similarly done for numerically sparse matrices in Gupta and Sidford (2018), although they employed this sampling (i) in a different setting and (ii) on one row at a time. Here, we analyze how to employ this sampling on all the rows simultaneously for ϵ-spectral-norm sparsification. This sampling is inferior to the

one in Theorem 2 in terms of $\mathsf{nnz}(\tilde{A})$, but has the additional property that the sparsity of every row is bounded. By applying this scheme to $A^\top$, we can alternatively obtain an ϵ-spectral-norm sparsifier where the sparsity of every column is bounded.

Theorem 8 *There is an algorithm that, given a matrix $A \in \mathbb{R}^{m\times n}$ and a parameter $\epsilon > 0$, computes in time $O(\mathsf{nnz}(A))$ with high probability an ϵ-spectral-norm sparsifier $\tilde{A}$ for A such that the sparsity of every row of $\tilde{A}$ is bounded by $O(\epsilon^{-2}\,\mathsf{ns}(A)\log(m+n))$.*

The algorithm is as follows. Given a matrix A and $\epsilon > 0$, define the sampling probabilities

$$\boxed{p_{ij} = \frac{|A_{ij}|}{\|A_i\|_1},}$$

and observe that for every i this induces probability distribution, i.e., $\sum_j p_{ij} = 1$. Let $s = O(\epsilon^{-2}\,\mathsf{ns}(A)\log(m+n))$. Now from each row of A sample s entries independently with replacement according to the above distribution, and scale it to get an unbiased estimator of that row; formally, for each row i and each $t = 1, \ldots, s$ draw a row vector

$$\boxed{Q_i^{(t)} = \left\{ \frac{A_{ij}}{p_{ij}} e_j^\top \quad \text{with prob. } p_{ij}, \right.}$$

where $\{e_j\}_j$ is the standard basis of $\mathbb{R}^n$. Next, average the t samples for each row, and arrange these rows in a matrix $\tilde{A}$ that is an unbiased estimator for A; formally,

$$\boxed{\tilde{A} = \sum_{i=1}^{m} e_i \frac{1}{s} \sum_{t=1}^{s} Q_i^{(t)}.}$$

Clearly $\mathbb{E}(\tilde{A}) = A$ and every row of $\tilde{A}$ has at most s non-zeros. In order to bound the probability that $\tilde{A}$ is an ϵ-spectral-norm sparsifier of A, similarly to the proof of Theorem 2, we employ the matrix-Bernstein bound stated in Theorem 7. The proof is omitted here and appears in the full version.

2.2. Lower Bounds

We provide a lower bound in Theorem 3 for spectral-norm sparsification, which almost matches the bound in Theorem 2 for a large range of ϵ and $\mathsf{ns}(A)$.

Theorem 3 *Let $0 < \epsilon < \frac{1}{2}$ and $n, k \geq 1$ be parameters satisfying $k \leq O(\epsilon^2 n \log^2 \frac{1}{\epsilon})$. Then, there exists a matrix $A \in \mathbb{R}^{n\times n}$ such that $\mathsf{ns}(A) = \Theta(k \log^2 \frac{1}{\epsilon})$ and, for every matrix B satisfying $\|A - B\|_2 \leq \epsilon \|A\|_2$, the sparsity of every row and every column of B is at least $\Omega(\epsilon^{-2} k \log^{-2} \frac{1}{\epsilon}) = \tilde{\Omega}(\epsilon^{-2}) \cdot \mathsf{ns}(A)$.*

Proof We shall assume that k divides n, and that both are powers of 2, which can be obtained with changing the bounds by a constant factor. Let $m = \frac{n}{k}$, and notice it is a power of 2 as well.

Construct first a vector $a \in \mathbb{R}^m$ by concatenating blocks of length 2^i whose coordinates have value $2^{-(1+\alpha)i}$, for each $i \in \{0, ..., \log m - 1\}$, where $1 > \alpha \geq \Omega(\log^{-1} m)$ is a parameter that we

will set later. The last remaining coordinate have value 0. Formally, the coordinates of a are given by $a_j = 2^{-(1+\alpha)\lfloor \log j \rfloor}$, except the last one which is 0. Its ℓ_1 norm is

$$\|a\|_1 = \sum_{j=1}^{m} a_j = \sum_{i=0}^{\log m - 1} 2^i \cdot 2^{-(1+\alpha)i} = \frac{1 - 2^{-\alpha \log m}}{1 - 2^{-\alpha}} = \Theta(\alpha^{-1}).$$

A similar computation shows that $\|a\|_2 = \Theta(1)$, and thus $\mathsf{ns}(a) = \Theta(\alpha^{-2})$. Denote by $a_{\mathrm{tail}(c)}$ the vector a without its c largest entries, then its ℓ_2 norm is

$$\|a_{\mathrm{tail}(c)}\|_2^2 \geq \sum_{i=\lfloor \log c \rfloor + 1}^{\log m - 1} 2^i \cdot 2^{-2(1+\alpha)i} = \Omega(c^{-(1+2\alpha)}), \tag{3}$$

which almost matches the upper bound of Lemma 3 in Gupta and Sidford (2018).

Now, for $k = 1$ we construct a circulant matrix $A \in \mathbb{R}^{m \times m}$ by letting the vector a be its first row, and the j-th row is a cyclic shift of a with offset j. By well-known properties of circulant matrices, the t-th eigenvalue of A is given by $\lambda_t = \sum_j a_j (\omega_t)^j$ where $\omega_t = \exp\left(i \frac{2\pi t}{m}\right)$ and i is the imaginary unit, so $\|A\|_2 = \|a\|_1 = \Theta(\alpha^{-1})$. Consider $B \in \mathbb{R}^{m \times m}$ satisfying $\|A - B\|_2 \leq \epsilon \|A\|_2$, and suppose some row B_j of B has s non-zeros. Then using (3),

$$\|A - B\|_2 \geq \|A_j - B_j\|_2 \geq \|a_{\mathrm{tail}(s)}\|_2 = \Omega(s^{-(\frac{1}{2}+\alpha)}).$$

By the error bound $\|A - B\|_2 \leq \epsilon \|A\|_2$, we must have $s \geq (\Omega(\epsilon/\alpha))^{-\frac{2}{1+2\alpha}} \geq \Omega((\epsilon/\alpha)^{-\frac{2}{1+2\alpha}})$, which bounds from below the sparsity of every row, and similarly also of every column, of B.

To generalize this to larger numerical sparsity, consider as a first attempt constructing a vector $a' \in \mathbb{R}^n$ by concatenating k copies of a. Then clearly $\mathsf{ns}(a') = \Theta(k\, \mathsf{ns}(a))$. The circulant matrix of a' is equivalent to $A \otimes C$, where C is the all-ones matrix of dimension $k \times k$, and $\otimes$ is the Kronecker product. But this matrix has low rank, and thus might be easier to approximate. We thus construct a different matrix $A' = A \otimes H_k$, where H_k is the $k \times k$ Hadamard matrix. Its numerical sparsity is the same as of the vector a', thus $\mathsf{ns}(A') = \Theta(k\, \mathsf{ns}(a))$. The eigenvalues of H_k are $\pm\sqrt{k}$. By properties of the Kronecker product, every eigenvalue of A' is the product of an eigenvalue of A with $\pm\sqrt{k}$, thus $\|A'\|_2 = \Theta(\sqrt{k}\|A\|_2) = \Theta(\sqrt{k}\alpha^{-1})$. We now apply the same argument we made for $k = 1$. Let $B' \in \mathbb{R}^{n \times n}$ be an ϵ-spectral-norm sparsifier of A'. If some row B'_j has s non-zeros then using (3),

$$\|A' - B'\|_2 \geq \|A'_j - B'_j\|_2 \geq \|a'_{\mathrm{tail}(s)}\|_2 = \Omega(\sqrt{k}(s/k)^{-(\frac{1}{2}+\alpha)}).$$

By the error bound $\|A' - B'\|_2 \leq \epsilon \|A'\|_2$, we must have $s \geq \Omega(k(\epsilon/\alpha)^{-\frac{2}{1+2\alpha}})$, which bounds the sparsity of every row and every column of B'.

We can set $\alpha = \log^{-1} \frac{1}{\epsilon} > \epsilon$. Note that this choice for α is in the range $[\log^{-1} \frac{n}{k}, 1]$, hence the construction hold. Now since $\frac{1}{1+2\alpha} \geq 1 - 2\alpha$, the lower bound on the sparsity of each row and each column of B' is $k(\epsilon/\alpha)^{-\frac{2}{1+2\alpha}} \geq k(\epsilon/\alpha)^{-2+4\alpha} \geq \Omega(k\epsilon^{-2} \log^{-2} \frac{1}{\epsilon})$. ■

3. Application I: Approximate Matrix Multiplication

In this section, we show how to use ℓ_1 row/column sampling for fast approximate matrix multiplication (AMM). Given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ and error parameter $\epsilon > 0$, the goal is to compute a matrix $C \in \mathbb{R}^{m\times p}$ such that $\|AB - C\| \leq \epsilon\|A\| \cdot \|B\|$, where the norm is usually either the Frobenius-norm $\|\cdot\|_F$ or spectral-norm $\|\cdot\|_2$. We provide the first results on AMM for numerically sparse matrices with respect to both norms.

Theorem 4 *There exists an algorithm that, given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ parameter $0 < \epsilon < \frac{1}{2}$ and constant factor estimates of $\|A\|_2$ and $\|B\|_2$, computes a matrix $C \in \mathbb{R}^{m\times p}$ satisfying with high probability $\|AB - C\|_2 \leq \epsilon\|A\|_2\|B\|_2$ in time*

$$O(\mathsf{nnz}(A) + \mathsf{nnz}(B)) + \tilde{O}(\epsilon^{-6}\sqrt{\mathsf{sr}(A)\,\mathsf{sr}(B)}\,\mathsf{ns}(A)\,\mathsf{ns}(B)).$$

Theorem 5 *There exists an algorithm that, given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ and parameter $0 < \epsilon < \frac{1}{2}$, computes a matrix $C \in \mathbb{R}^{m\times p}$ satisfying $\mathbb{E}\,\|AB - C\|_F \leq \epsilon\|A\|_F\|B\|_F$ in time*

$$O(\mathsf{nnz}(A) + \mathsf{nnz}(B) + \epsilon^{-6}\,\mathsf{ns}(A)\,\mathsf{ns}(B)).$$

The proofs of these theorems combine Theorem 8 with previous results on numerical sparsity and with previous results on AMM.

Lemma 9 (Lemma 4 of Gupta and Sidford (2018)) *Given a vector $a \in \mathbb{R}^n$ and a parameter $\epsilon > 0$, independently sampling $(\epsilon^{-2}\,\mathsf{ns}(a))$ entries according to the distribution $\{p_i = \frac{|a_i|}{\|a\|_1}\}_i$ and re-weighting the sampled coordinates by $\frac{1}{p_i} \cdot \frac{1}{\epsilon^{-2}\,\mathsf{ns}(a)}$, outputs a $(\epsilon^{-2}\,\mathsf{ns}(a))$-sparse vector $a' \in \mathbb{R}^n$ satisfying $\mathbb{E}\,a' = a$ and $\mathbb{E}(\|a'\|_2^2) \leq (1+\epsilon^2)\|a\|_2^2$.*

3.1. Proof of Theorem 4 (Spectral-Norm AMM)

In order to prove Theorem 4, we will use a result from Magen and Zouzias (2011). Given matrices A, B, their product is $AB = \sum_i A^i B_i$. The algorithm in Magen and Zouzias (2011) samples corresponding pairs of columns from A and rows from B, hence the time it takes to compute an approximation of AB depends on the sparsity of these rows and columns.

Lemma 10 (Theorem 3.2 (ii) of Magen and Zouzias (2011).) *There exists an algorithm that, given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$, a parameter $0 < \epsilon < 1/2$ and constant factor estimates of $\|A\|_2$ and $\|B\|_2$, computes in time*

$$O\Big(\mathsf{nnz}(A) + \mathsf{nnz}(B) + \epsilon^{-2}\,\mathsf{csp}(A)\,\mathsf{rsp}(B)\sqrt{\mathsf{sr}(A)\,\mathsf{sr}(B)}\log\left(\epsilon^{-1}\,\mathsf{sr}(A)\,\mathsf{sr}(B)\right)\Big)$$

a matrix C that satisfies

$$\mathbb{P}(\|C - AB\|_2 \geq \epsilon\|A\|_2\|B\|_2) \leq \frac{1}{\mathrm{poly}(\mathsf{sr}(A)\,\mathsf{sr}(B))}.$$

Proof of Theorem 4. Given $\epsilon > 0$, our algorithm is as follows.

1. Apply the algorithm in Theorem 8 on A with parameter $\epsilon/4$ to compute a matrix A' satisfying $\|A' - A\|_2 \leq \frac{\epsilon}{4}\|A\|_2$ and $\mathsf{csp}(A') \leq O(\epsilon^{-2}\,\mathsf{ns}(A)\log(m+n))$, and apply it on B with parameter $\epsilon/4$ to compute a matrix B' satisfying $\|B' - B\|_2 \leq \frac{\epsilon}{4}\|B\|_2$ and $\mathsf{rsp}(B') \leq O(\epsilon^{-2}\,\mathsf{ns}(B)\log(n+p))$.
2. Apply the algorithm in Lemma 10 on A', B' with parameter $\epsilon/4$ to produce a matrix C. Output C.

It holds that $\mathbb{E}\,\|A'\|_F^2 \leq \left(1 + O(\frac{\epsilon^2}{\log(m+n)})\right)\|A\|_F^2$, since the sampling in Theorem 8 satisfies the conditions for Lemma 9. Thus, with high probability, $\mathsf{sr}(A') \in (1 \pm O(\epsilon))\,\mathsf{sr}(A)$, and similarly for B'. Ignoring the $\mathsf{nnz}(\cdot)$ terms, the time it takes for the algorithm from Lemma 10 on A', B' is

$$O\Big(\epsilon^{-6}\,\mathsf{ns}(A)\,\mathsf{ns}(B)\log(m+n)\log(n+p)\sqrt{\mathsf{sr}(A)\,\mathsf{sr}(B)}\log\left(\epsilon^{-1}\,\mathsf{sr}(A)\,\mathsf{sr}(B)\right)\Big),$$

hence the stated overall running time. The output C satisfies with high probability

$$\begin{aligned}\|AB - C\|_2 &\leq \|(A - A')B\|_2 + \|(A'(B - B')\|_2 + \|A'B' - C\|_2 \\ &\leq \tfrac{\epsilon}{4}\|A\|_2\|B\|_2 + \tfrac{\epsilon}{4}\|B\|_2(1 + \tfrac{\epsilon}{4})\|A\|_2 + \tfrac{\epsilon}{4}(1 + \tfrac{\epsilon}{4})^2\|A\|_2\|B\|_2 \leq \epsilon\|A\|_2\|B\|_2.\end{aligned}$$

■

3.2. Proof of Theorem 5 (Frobenius-Norm AMM)

We provide a sampling lemma for estimating outer products in the Frobenius-norm.

Lemma 11 *There exists an algorithm that, given vectors $a \in \mathbb{R}^n, b \in \mathbb{R}^m$ and parameter $0 < \epsilon < 1$, computes in time $O(\|a\|_0 + \|b\|_0)$ vectors $a', b' \in \mathbb{R}^n$ with sparsity $\epsilon^{-2}\,\mathsf{ns}(a)$ and $\epsilon^{-2}\,\mathsf{ns}(b)$, respectively, satisfying $\mathbb{E}(a'b'^\top) = ab^\top$ and $\mathbb{E}\,\|a'b'^\top - ab^\top\|_F^2 \leq \epsilon^2\|a\|_2^2\|b\|_2^2$.*

Proof Given $0 < \epsilon < 1$, our algorithm is as follows.

1. Independently sample (with repetitions) $9\epsilon^{-2}\,\mathsf{ns}(a)$ entries from a according to the distribution $\{p_i^{(a)} = \frac{|a_i|}{\|a\|_1}\}_i$ and $9\epsilon^{-2}\,\mathsf{ns}(b)$ entries from b according to the distribution $\{p_i^{(b)} = \frac{|b_i|}{\|b\|_1}\}_i$.
2. Re-weight the sampled entries of a by $\frac{1}{p_i^{(a)}} \cdot \frac{1}{9\epsilon^{-2}\,\mathsf{ns}(a)}$ and similarly for b. Output the sampled vectors.

Denote the sampled vectors a' and b'. They satisfy the conditions of Lemma 9, hence they satisfy $\mathbb{E}(a'b'^\top) = ab^\top$ and $\mathbb{E}(\|a'\|_2^2) \leq (1 + \epsilon^2/3)\|a\|_2^2$ and similarly for b'. Thus,

$$\mathbb{E}\,\|a'b'^\top - ab^\top\|_F^2 = \mathbb{E}\,\|a'b'^\top\|_F^2 - \|ab^\top\|_F^2 = \mathbb{E}\,\|a'\|_2^2\|b'\|_2^2 - \|a\|_2^2\|b\|_2^2 \leq \epsilon^2\|a\|_2^2\|b\|_2^2.$$ ■

In order to prove Theorem 5, we will use a result from Drineas et al. (2006). The algorithm in Drineas et al. (2006) samples corresponding pairs of columns from A and rows from B, hence the time it takes to compute an approximation of AB depends on the sparsity of these rows and columns.

Lemma 12 (Lemma 4 of Drineas et al. (2006)) *There exists an algorithm that, given matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$ and parameter $0 < \epsilon < 1$, computes in time $O(\mathsf{nnz}(A) + \mathsf{nnz}(B) + \epsilon^{-2}\,\mathsf{csp}(A)\,\mathsf{rsp}(B))$ a matrix $C \in \mathbb{R}^{m\times p}$ satisfying $\mathbb{E}\,\|AB - C\|_F \leq \epsilon\|A\|_F\|B\|_F$.*

Proof of Theorem 5. Let $0 < \epsilon < 1$. Recall that $AB = \sum_i A^i B_i$. Our algorithm is as follows.

1. Apply the algorithm in Lemma 11 on each pair of vectors A^i, B_i with parameter $\epsilon/3$ to obtain their sparse estimates $\hat{A}^i$ and $\hat{B}_i$.
2. Arrange the column vectors $\{\hat{A}^i\}$ in a matrix $\hat{A}$ and the row vectors $\{\hat{B}_i\}$ in a matrix $\hat{B}$.
3. Apply the algorithm in Lemma 12 on the matrices $\hat{A}$ and $\hat{B}$ with parameter $\epsilon/3$ to obtain their approximate product C. Output C.

The sparsity of the columns of $\hat{A}$ is bounded by $\epsilon^{-2}\,\mathsf{ns}(A)$ and the sparsity of the rows of $\hat{B}$ is bounded by $\epsilon^{-2}\,\mathsf{ns}(B)$. By the triangle inequality, Jensen inequality, Lemma 11 and Cauchy-Schwarz inequality,

$$\begin{aligned}\mathbb{E}\,\|AB - \hat{A}\hat{B}\|_F = \mathbb{E}\,\|\sum_i A^i B_i - \hat{A}^i\hat{B}_i\|_F &\le \sum_i \mathbb{E}\,\|A^i B_i - \hat{A}^i\hat{B}_i\|_F \\ &\le \sum_i \sqrt{\mathbb{E}\,\|A^i B_i - \hat{A}^i\hat{B}_i\|_F^2} \le \tfrac{\epsilon}{3}\sum_i \|A^i\|_2\|B_i\|_2 \le \tfrac{\epsilon}{3}\|A\|_F\|B\|_F.\end{aligned}$$

Additionally, by Jensen's inequality and Lemma 9,

$$\mathbb{E}\,\|\hat{A}\|_F \le \sqrt{\mathbb{E}\,\|\hat{A}\|_F^2} \le \sqrt{\sum_i (1+\tfrac{\epsilon^2}{9})\|A^i\|_2^2} \le (1+\tfrac{\epsilon}{3})\|A\|_F,$$

and similarly for $\hat{B}$. By the triangle inequality and Lemma 12,

$$\begin{aligned}\mathbb{E}\,\|C - AB\|_F &\le \mathbb{E}(\|C - \hat{A}\hat{B}\|_F + \|\hat{A}\hat{B} - AB\|_F) \\ &\le \tfrac{\epsilon}{3}(1+\tfrac{\epsilon}{3})^2\|A\|_F\|B\|_F + \tfrac{\epsilon}{3}\|A\|_F\|B\|_F \le \epsilon\|A\|_F\|B\|_F.\end{aligned}$$

Except for the $\mathsf{nnz}(\cdot)$ terms, the time it takes to compute the last step is $O(\epsilon^{-6}\,\mathsf{ns}(A)\,\mathsf{ns}(B))$, and the claimed running time follows. ■

References

Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9–es, 2007.

Dimitris Achlioptas, Zohar S. Karnin, and Edo Liberty. Near-optimal entrywise sampling for data matrices. In *Advances in Neural Information Processing Systems*, pages 1565–1573, 2013.

Sanjeev Arora, Elad Hazan, and Satyen Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 339–348. IEEE, 2005.

Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 272–279. Springer, 2006.

Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Coordinate methods for matrix games. In *61st Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 283–293. IEEE, 2020.

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

Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal approximate matrix product in terms of stable rank. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

Alexandre d'Aspremont. Subsampling algorithms for semidefinite programming. *Stochastic Systems*, 1(2):274–305, 2011.

Petros Drineas and Anastasios Zouzias. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality. *Information Processing Letters*, 111(8):385–389, 2011.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning*, pages 2540–2548, 2015.

Mina Ghashami, Edo Liberty, and Jeff M. Phillips. Efficient frequent directions algorithm for sparse matrices. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 845–854, 2016.

Alex Gittens and Joel A. Tropp. Error bounds for random matrix approximation schemes. *arXiv preprint arXiv:0911.4108*, 2009.

Neha Gupta and Aaron Sidford. Exploiting numerical sparsity for efficient learning: faster eigenvector computation and regression. In *Advances in Neural Information Processing Systems*, pages 5269–5278, 2018.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5(9), 2004.

Zengfeng Huang. Near optimal frequent directions for sketching dense and sparse matrices. *Journal of Machine Learning Research*, 20(56):1–23, 2019.

Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Abhisek Kundu and Petros Drineas. A note on randomized element-wise matrix sparsification. *arXiv preprint arXiv:1404.0320*, 2014.

Abhisek Kundu, Petros Drineas, and Malik Magdon-Ismail. Recovering PCA and sparse PCA via hybrid-(l1, l2) sparse sampling of data elements. *The Journal of Machine Learning Research*, 18 (1):2558–2591, 2017.

Miles Lopes. Estimating unknown sparsity in compressed sensing. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 217–225. PMLR, 2013.

Avner Magen and Anastasios Zouzias. Low rank matrix-valued Chernoff bounds and approximate matrix multiplication. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1422–1436. SIAM, 2011.

Youssef Mroueh, Etienne Marcheret, and Vaibahava Goel. Co-Occurring Directions Sketching for Approximate Matrix Multiply. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 567–575. PMLR, 2017.

Nam H. Nguyen, Petros Drineas, and Trac D. Tran. Tensor sparsification via a bound on the spectral norm of random tensors. *Information and Inference: A Journal of the IMA*, 4(3):195–229, 2015.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Qiaomin Ye, Luo Luo, and Zhihua Zhang. Frequent direction algorithms for approximate matrix multiplication with applications in CCA. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, pages 2301–2307. IJCAI/AAAI Press, 2016.