מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Thesis for the degree
Master of Science

עבודת גמר (תזה) לתואר
מוסמך למדעים

Submitted to the Scientific Council of the
Weizmann Institute of Science
Rehovot, Israel

מוגשת למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

By
Shay Sapir

מאת
שי ספיר

דגימה כמעט אופטימלית של איברי מטריצות בעלות דלילות נומרית
Near-Optimal Entrywise Sampling of Numerically Sparse Matrices

Advisor:
Prof. Robert Krauthgamer

מנחה:
פרופ' רוברט קראוטגמר

March 2021

ניסן התשפ"א

Abstract

Large-scale matrix computations require a significant amount of computational resources and are sometimes infeasible. Many real-world data sets are sparse or almost sparse, which can relax this computational burden. 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.

Our first set of results focuses on approximating an input matrix A by entrywise sampling, a question raised by the seminal work of Achlioptas and Mcsherry [AM07]. More precisely, the goal is to quickly compute a sparse matrix $\tilde{A}$ satisfying $\|A - \tilde{A}\|_2 \leq \epsilon\|A\|_2$ (an additive spectral norm approximation) for a given error parameter $\epsilon > 0$. We propose a scheme that sparsifies a numerically-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 the number of non-zero entries, denoted as $\mathsf{nnz}(\tilde{A})$ 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, Karnin and Liberty [AKL13], 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.

Another way to relieve the computational burden of large-scale matrix computations is by restricting ourselves to the streaming model, where the matrix is accessed via a row-order stream, and the algorithm never stores the entire matrix. Our second set of results considers this setting of row-order streams when the rows are time-sensitive and expire after some number of steps, called the sliding-window row-order model. We provide the first $(1+\epsilon)$-approximation of Schatten p-norms in this setting. Our main technical contribution is a proof that Schatten p-norms in row-order streams are smooth, and thus fit the smooth-histogram technique of Braverman and Ostrovsky [BO07] for sliding-window streams.

Acknowledgements

I would like to thank my advisor Robert Krauthgamer, whose support and insightful comments were invaluable during the course of my study, especially considering the ongoing pandemic. I wish to thank Vladimir Braverman and Aditya Krishnan for collaborating with Robi and myself on the first set of results. It was a pleasure both professionally and personally. I thank the people of the department of Mathematics and Computer Science in the Weizmann Institute, for creating an amazing and supportive research environment. I would also like to express my gratitude to my family and friends for their moral support and warm encouragements. Special thanks to my partner Ella Fishman, for her love and support.

Contents

Chapter 1

Introduction

Vectors and matrices are common tools for organization of data, as they encompass its geometric and algebraic properties. Linear algebra offers a rich algorithmic toolset for manipulating matrices, and is used in many applications, e.g. in machine learning and data science. A central tool is singular value decomposition (SVD), which can reduce noise and find trends in the data. Classical approaches to numerical linear algebra provide high precision solutions, but for large-scale computation, they typically require a significant amount of time, space or communication resources. The recent field of randomized numerical linear algebra (RandNLA) offers to reduce the computational burden by using Monte-Carlo algorithms, i.e. with approximation and randomization. Two major techniques in RandNLA are (i) sampling, where the input is sampled according to a well-designed distribution and (ii) linear sketching, where the input undergoes a linear transformation, often described by multiplication with a random matrix. For matrix computations, there are three prominent algorithmic paradigms in RandNLA.

1. Nearly linear time. Algorithms that solve a desired problem for an input matrix A in time that is nearly linear in the number of non-zero elements of A, denoted as $\mathsf{nnz}(A)$. Usually, the time is of the form $O(\mathsf{nnz}(A) + f(A))$, where $O(\mathsf{nnz}(A))$ time is needed to just read all of the data, and $f(A)$ is some function of A (e.g. depending on rank, desired accuracy, etc). The goal is then to design an algorithm for which this expression is small. A simple classical algorithm that fits this type is the power method, which given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, iteratively applies A to an arbitrary vector $x \in \mathbb{R}^n$. This method can $(1+\epsilon)$-approximate the top eigenvalue of A using $O(\epsilon^{-1}\log n)$ iterations. Each iteration consists of a matrix-vector multiplication, which takes $O(\mathsf{nnz}(A))$ time. There are other algorithms for approximating the top eigenvalue in nearly linear time, e.g. Lanczos method and the Stochastic Variance Reduced Gradient Descent (SVRG) method [JZ13, GS18], which succeeds to separate the $O(\mathsf{nnz}(A))$ term from the ϵ^{-1} factor. Even more complicated problems admit randomized algorithms of this type, e.g. low rank approximation [FKV04, Woo14] and linear regression [Woo14]. A well studied (and still open) problem is the computation time of multiplying two $n \times n$ matrices. It can be computed naively in $O(n^3)$ time, or in $O(n^\omega)$ time using Coppersmith-Winograd [CW90] like algorithm, where the current bound on the exponent is $\omega < 2.37286$ [AW21], but the hidden constant factor is impractically large. There is also a randomized algorithm that approximates matrix multiplication in $O(\epsilon^{-2}n^2)$ time [DKM06]. We advance this front further in section 2.3.

2. Streaming model. In this model, the input is a sequence of items that can be read only sequentially, usually in one pass. Algorithms in this model are restricted to have small memory compared to the input size, and cannot access earlier input portions, hence they effectively have to compress the input. One famous setting is the frequency-vector model, where the input is a stream of items $(i, \delta) \in [n] \times \mathbb{R}$, representing updates of the form $x_i \leftarrow x_i + \delta$ to a vector $x \in \mathbb{R}^n$. A typical goal is to estimate $\|x\|_p^p$, called the ℓ_p frequency moment [AMS99]. $(1 + \epsilon)$-approximation of the frequency moments (with constant success probability) for $0 < p \leq 2$ requires $\Theta(\epsilon^{-2} \log n)$ bits of space [KNW10], accomplished by a linear sketch using p-stable distributions [Ind06]. For $p \geq 2$ and constant $\epsilon > 0$, one can $(1 + \epsilon)$-approximate the frequency moments with constant probability using $\tilde{O}(n^{1-2/p})$ bits of space [IW05]. The polynomial dependence in n (i.e. $n^{1-2/p}$) is optimal, while there are poly$(\log n, \epsilon, \delta)$ improvements, see [GW18] and further references provided therein. We discuss a similar setting for matrices (and not vectors) in Chapter 3. Another famous problem is ℓ_2 point queries, where the goal is to estimate each entry of x up to $\pm\epsilon\|x\|_2$ additive error, which can be solved using $O(\epsilon^{-2} \log^2 n)$ bits of space [CCF02].

3. Sparsification algorithms. The goal is to sparsify an input matrix while preserving certain properties. The result can then be used in downstream applications. Two prominent schemes are (i) sparsifying the entire input and (ii) selecting a subset of rows/columns as representatives for the input. An example that fits the first scheme is graph sparsification. Spielman and Srivastava [SS11] showed that given a graph G with Laplacian L, sampling $O(\epsilon^{-2} n \log n)$ edges according to the effective resistance yields with high probability a spectral sparsifier of G, i.e. a graph with Laplacian $\tilde{L}$ such that $(1-\epsilon)L \preceq \tilde{L} \preceq (1+\epsilon)L$.[1] In matrix computation, it is well known that the best rank-k approximation of a matrix A (when measuring distance in any unitarily invariant norm) is achieved by truncating the small singular values in the SVD (i.e. keeping only the k largest singular values). Although this truncated SVD provides the smallest approximation error, the rows of its output are linear combinations of the rows of A, which can be sometimes meaningless (e.g. in biological applications). One can resolve this by employing the second sparsification scheme in the form of the CUR matrix decomposition, where C and R are made of columns and rows of A, respectively, and U is a small carefully constructed matrix (roughly $\frac{k}{\epsilon}^2 \times \frac{k}{\epsilon}^2$) such that $A \approx CUR$ [MD09]. One can compute such a decomposition with relative-error guarantees by sampling rows/columns according to their leverage scores [DMM08].[2] We discuss the problem of matrix sparsification in Chapter 2. Our setting is more general than the Laplacian example above, but our approximation guarantees are weaker, as discussed in the next paragraph.

Measures of matrix approximation. For matrix B that approximates another matrix A, we denote $E = A - B$. Two common measures of error are (i) Frobenius-norm, denoted $\|E\|_F$, which is simply the ℓ_2-norm of the vector form of E; and (ii) spectral-norm (also known as operator-norm), which is $\|E\|_2 = \max_{\|x\|_2=1} \|Ex\|_2$ and it also equals the largest singular value of E. The goal is then to have $\|E\|_F \leq \epsilon\|A\|_F$ or $\|E\|_2 \leq \epsilon\|A\|_2$, respectively. Minimizing $\|E\|_2$ is a natural goal, as it minimizes the amount by which E stretches different vectors, i.e. it makes it a near rotation. See [AKL13, section 2] for further discussion. The sparsification we provide in Chapter 2 is with respect to the spectral norm. Another common measure of error is called spectral approximation, defined for PSD matrices A, B as $(1-\epsilon)A \preceq B \preceq (1+\epsilon)A$. It immediately implies that $\|E\|_2 \leq \epsilon\|A\|_2$ and $\|E\|_F \leq \epsilon\|A\|_F$, hence it is stronger than the spectral-norm and Frobenius-norm measures.

[1] For symmetric matrices A, B, we denote $A \succeq 0$ if A is positive semi-definite (PSD), and $A \preceq B$ if $B - A$ is PSD.

[2] The rows leverage scores are the norms of the rows of the $n \times k$ matrix whose columns are the top k right singular vectors of A, divided by k.

The rest of this thesis is organized as follows. Chapter 2 is based on our manuscript [BKKS20a],which was submitted to COLT. It proposes a scheme that sparsifies a matrix with respect to the spectral-norm, and shows applications to approximate matrix multiplication (Section 2.3) and to ridge regression (Section 2.4). Chapter 3 is based on our manuscript [KS21], which was submitted to IPL. It gives the first $(1 + \epsilon)$-approximation of Schatten p-norms in sliding-window streams.

Chapter 2

Near-Optimal Entrywise Sampling of Numerically Sparse Matrices

2.1 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 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 [GLP16, Hua18], regression problems [JZ13] and semi-definite programming [d'A11, AHK05]. 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* [Lop13, GS18] to be

$$\mathsf{ns}(a) \coloneqq \min\{k \geq 0 : \|a\|_1 \leq \sqrt{k}\|a\|_2\}. \tag{2.1}$$

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 [Hoy04, HR09]. 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 [AM07] 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 2.1.1. 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

$$\|\tilde{A} - A\|_2 \le \epsilon \|A\|_2. \tag{2.2}$$

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 [AM07] 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 [AHK06, GT09, DZ11, NDT15, AKL13, KD14, KDMI17], 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 [AKL13], provides a bound for a restricted class of "data matrices". Specifically, they look at matrices $A \in \mathbb{R}^{m\times n}$ such that $\min_i \|A_i\|_1 \ge \max_j \|A^j\|_1$, 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.

2.1.1 Main Results

We generalize the sparsity bound of [AKL13], 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 2.1.2. *There is an algorithm that, given a matrix $A \in \mathbb{R}^{m\times n}$ and a parameter $\epsilon > 0$, where $m \ge n$, computes with high probability an ϵ-spectral-norm sparsifier $\tilde{A}$ for A with expected sparsity*

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

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 [AKL13]. 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.

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

In addition to the sampling scheme in Theorem 2.1.2, we analyze ℓ_1-sampling from every row (in Section 2.2.1).[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 2.1.3. *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.

2.1.2 Comparison to Previous Work

The work of [AM07] initiated a long line of work on entrywise sampling schemes that approximate a matrix under spectral-norm [AHK06, GT09, DZ11, KD14, KDMI17, NDT15, AKL13]. 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 [AHK06] 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 [AM07, DZ11, NDT15, KD14] point out that sampling according to the ℓ_2 distribution causes small entries to "blow-up" when sampled. Some works, e.g. [DZ11], get around this by zeroing-out small entries or by exceptional handling of small entries, e.g. [AM07], while others used distributions that combine the ℓ_1 and ℓ_2 distributions, e.g. [KD14]. All these works sample $\Omega(\epsilon^{-2} n \, \mathsf{sr}(A))$ entries in expectation to achieve an ϵ-spectral-norm approximation and our Theorem 2.1.2 provides an asymptotically better bound. For a full comparison see Table 2.1.

All these algorithms, including the algorithm of Theorem 2.1.2, 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 2.2.2, 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 2.1.2 can be implemented in two-passes over the data and in $O(\mathsf{nnz}(A))$ time.

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

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

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

Table 2.1: 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 2.1.2.

Expected Number of Samples	**Reference**	**Compared to Thm. 2.1.2**
$O(\epsilon^{-1}n\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)})$	[AHK06]	$\tilde{O}_\epsilon\Big(\min\Big(\frac{n}{\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)}}, \sqrt{n}\Big)\Big)$
$O(\epsilon^{-2}n\,\mathsf{sr}(A) + n\,\mathrm{polylog}(n))$	[AM07]	$\tilde{O}_\epsilon\Big(\min\Big(\frac{n}{\mathsf{ns}(A)}, \sqrt{\frac{n\,\mathsf{sr}(A)}{\mathsf{ns}(A)}}\Big)\Big)$
$\tilde{O}(\epsilon^{-2}n\,\mathsf{sr}(A))$	[DZ11, KD14]	
$\tilde{O}(\epsilon^{-2}\,\mathsf{ns}(A)\,\mathsf{sr}(A)+\epsilon^{-1}\sqrt{\mathsf{ns}(A)\,\mathsf{sr}(A)n})$	[AKL13]; Theorem 2.1.2	[AKL13] is only for data matrices
$\tilde{O}(\epsilon^{-2}n\,\mathsf{ns}(A))$	Theorem 2.2.2	bounded row-sparsity
$\Omega(\epsilon^{-2}n\,\mathsf{ns}(A)\log^{-4}\frac{1}{\epsilon})$	Theorem 2.1.3	$\mathsf{sr}(A) = \Theta(n)$

2.1.3 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 [AM07], to semidefinite programming [AHK06], and to PCA and sparse PCA [KDMI17]. 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 $\|AB - C\| \le \epsilon\|A\| \cdot \|B\|$, where the norm is usually either Frobenius-norm $\|\cdot\|_F$ or spectral-norm $\|\cdot\|_2$. In Section 2.3, 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 2.1.4. *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 $\|AB - C\|_2 \le \epsilon\|A\|_2\|B\|_2$ 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 2.1.5. *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 [FKV04, DKM06, CW09, MZ11, CNW16, YLZ16, MMG17] 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 [MZ11] for spectral-norm error and [DKM06] for Frobenius-norm error. Sampling based methods, including that of [DKM06, MZ11], 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.

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 $\|\hat{x} - x^*\|_{A^\top A+\lambda I} \le \epsilon \|x_0 - x^*\|_{A^\top A+\lambda I}$.[5] We provide algorithms in Appendix 2.4 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 [FGKS15].

Theorem 2.1.6. *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 2.1.6 can be reduced by a factor of roughly $(\mathsf{sr}(A)/n)^{1/6}$, see Section 2.4.2 .

Solving linear systems using preconditioning has a rich history that is beyond the scope of this work to summarize. Recently, the work of [GS18] designed algorithms with improved running times over popular methods using the Stochastic Variance Reduced Gradient Descent (SVRG) framework of [JZ13]. 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 [AKL13] for computing the preconditioners, they required a stronger bound on the spectral-norm approximation than Theorem 2.1.6 does.

[5] For a PSD matrix M, we denote $\|x\|_M \coloneqq x^\top M x$.

Our result is in general incomparable to that of [GS18]. 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, see Theorem 2.4.4 in Appendix 2.4.2 for details.

Very recently, [CJST20] 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.

2.2 Spectral-Norm Sparsification

In this section we state and prove our main results. We first prove the upper bound in Theorem 2.1.2. Then we analyze ℓ_1 sampling from the rows in Theorem 2.2.2, Section 2.2.1 that gives a slightly weaker bound but has the property that the resulting matrix has uniform row sparsity. In Section 2.2.2, we prove the lower bound in Theorem 2.1.3.

Theorem 2.1.2. *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*

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

Moreover, it runs in $O(\mathsf{nnz}(A))$-time when a constant factor estimate of $\|A\|_2$ is given. [6]

Before we prove Theorem 2.1.2, we start by stating a result on the concentration of sums of independent random matrices; the Matrix Bernstein Inequality.

Theorem 2.2.1 (Matrix Bernstein, Theorem 1.6 of [Tro12])**.** *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*

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

Then for all $t \geq 0$,

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

Proof of Theorem 2.1.2. Let $\epsilon > 0$. Given a matrix A, define sampling probabilities as follows.

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

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

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

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

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

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

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

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

where the last inequality follows from Cauchy-Schwarz inequality.

In order to bound σ^2, first notice that $\mathrm{var}(\tilde{A}_{ij}) \leq \mathbb{E}(\tilde{A}_{ij}^2) = \frac{A_{ij}^2}{sp^*_{ij}}$. 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

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

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

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

and since $\mathsf{sr}(A) = \frac{\|A\|_F^2}{\|A\|_2^2}$, by setting $s = O(\epsilon^{-2} \mathsf{ns}(A)\, \mathsf{sr}(A) \ln m + \epsilon^{-1}\sqrt{\mathsf{ns}(A) \cdot n \cdot \mathsf{sr}(A)} \ln m)$ we conclude that with high probability $\|\tilde{A} - A\|_2 \leq \epsilon\|A\|_2$, which completes the proof of Theorem 2.1.2. □

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 [GS18], 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 2.1.2 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 2.2.2. *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 $s = O(\epsilon^{-2} \mathsf{ns}(A) \ln(m + n))$. 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 $t = 1, \dots, s$ draw a row vector

$$\boxed{Q_i^{(t)} = \begin{cases} \frac{A_{ij}}{p_{ij}} e_j^\top & \text{with prob. } p_{ij}, \end{cases}}$$

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 2.1.2, we employ the matrix-Bernstein bound as stated in Theorem 2.2.1.

Proof of Theorem 2.2.2. Given a matrix A and $\epsilon > 0$, let $k = \mathsf{ns}(A)$ and apply the algorithm described above. Note that by the definition of $\mathsf{ns}(A)$ and by spectral-norm properties, the i-th row of A satisfies

$$\|A_i\|_1 \leq \sqrt{k}\|A_i\|_2 \leq \sqrt{k}\|A\|_2. \tag{2.3}$$

For each random draw, define a matrix $Z_{(it)}$ with exactly one non-zero row formed by placing $A_i - Q_i^{(t)}$ at the i-th row; formally, let $Z_{(it)} = e_i(A_i - Q_i^{(t)})$. Where it is clear from context we will omit the superscript from $Q_i^{(t)}$. The spectral-norm of $Z_{(it)}$ is

$$\|Z_{(it)}\|_2 = \|A_i - Q_i^{(t)}\|_2 \le \|A_i\|_2 + \|Q_i^{(t)}\|_2 = \|A_i\|_2 + \|A_i\|_1 \le 2\sqrt{k}\|A\|_2 =: R.$$

To bound σ^2, notice that $Z_{(it)}Z_{(it)}^\top$ has a single non-zero at the (i,i) entry with value $\|A_i - Q_i^{(t)}\|_2^2$, hence

$$\begin{aligned}\big\| \mathbb{E} \sum_{i,t} Z_{(it)}Z_{(it)}^\top \big\|_2 &= s \max_i \mathbb{E}\, \|A_i - Q_i\|_2^2 = s \max_i \mathbb{E}\, \|Q_i\|_2^2 - \|A_i\|_2^2 \\ &\le s \max_i \sum_j \|A_i\|_1 \cdot |A_{ij}| \le sk\|A\|_2^2.\end{aligned}$$

The other term $Z_{(it)}^\top Z_{(it)}$ satisfies $\mathbb{E}(Z_{(it)}^\top Z_{(it)}) = \mathbb{E}\left(Q_i^\top (Q_i - A_i)\right) = \mathbb{E}(Q_i^\top Q_i) - A_i^\top A_i$. The matrix $\mathbb{E}(Q_i^\top Q_i)$ is diagonal with value $|A_{ij}| \cdot \|A_i\|_1$ at the (j,j) entry, hence

$$\begin{aligned}\big\| \sum_{i,t} \mathbb{E}(Z_{(it)}^\top Z_{(it)}) \big\|_2 &= s \big\| \sum_i (\mathbb{E}(Q_i^\top Q_i) - A_i^\top A_i) \big\|_2 \\ &= s\| \sum_i \mathbb{E}(Q_i^\top Q_i) - A^\top A\|_2 \\ &\le s\big(\big\| \sum_i \mathbb{E}(Q_i^\top Q_i)\big\|_2 + \|A^\top A\|_2\big) \\ &= s\big(\max_j \sum_i |A_{ij}| \cdot \|A_i\|_1 + \|A\|_2^2\big) \\ &\le s\sqrt{k}\big(\|A\|_2 \max_j \sum_i |A_{ij}| + \|A\|_2^2\big) \\ &= s\sqrt{k}\big(\|A\|_2 \max_j \|A^j\|_1 + \|A\|_2^2\big) \le 2s \cdot k \cdot \|A\|_2^2 =: \sigma^2.\end{aligned}$$

Now, by the matrix-Bernstein bound as stated in Theorem 2.2.1,

$$\begin{aligned}\mathbb{P}(\|A - \tilde{A}\|_2 \ge \epsilon\|A\|_2) &= \mathbb{P}\big(\big\| \sum_{i,t} Z_{(it)} \big\|_2 \ge s\epsilon\|A\|_2\big) \\ &\le (m+n)\exp\Big(- \frac{s\epsilon^2\|A\|_2^2/2}{2k\|A\|_2^2 + \frac{2\epsilon}{3}\sqrt{k}\|A\|_2^2} \Big),\end{aligned}$$

and by setting $s = O(\epsilon^{-2}k\ln(m+n))$ we conclude that with high probability $\|\tilde{A} - A\|_2 \le \epsilon\|A\|_2$. □

2.2.2 Lower Bounds

We provide a lower bound in Theorem 2.1.3 for spectral-norm sparsification, which almost matches the bound in Theorem 2.1.2 for a large range of ϵ and $\mathsf{ns}(A)$.

Theorem 2.1.3. *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

$$\|a_{\mathrm{tail}(c)}\|_2^2 \geq \sum_{i=\lfloor \log c \rfloor + 1}^{\log m - 1} 2^i \cdot 2^{-2(1+\alpha)i} = \Omega(c^{-(1+2\alpha)}), \tag{2.4}$$

which almost matches the upper bound of Lemma 3 in [GS18].

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 (2.4),

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

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 (2.4),

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

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 $k(\epsilon/\alpha)^{-\frac{2}{1+2\alpha}} \geq k(\epsilon/\alpha)^{-2+4\alpha} \geq \Omega(k\epsilon^{-2} \log^{-2} \frac{1}{\epsilon})$. □

2.3 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 2.1.4. *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 2.1.5. *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 2.2.2 with previous results on numerical sparsity and with previous results on AMM.

Lemma 2.3.1 (Lemma 4 of [GS18])**.** *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 $\mathbb{E}(\|a'\|_2^2) \leq (1 + \epsilon^2)\|a\|_2^2$.*

2.3.1 Proof of Theorem 2.1.4 (Spectral-Norm AMM)

In order to prove Theorem 2.1.4, we will use a result from [MZ11]. Given matrices A, B, their product is $AB = \sum_i A^i B_i$. The algorithm in [MZ11] 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 2.3.2 (Theorem 3.2 (ii) of [MZ11].)**.** *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 2.1.4. Given $\epsilon > 0$, our algorithm is as follows.

1. Apply the algorithm in Theorem 2.2.2 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 2.3.2 on A', B' with parameter $\epsilon/4$ to produce a matrix C. Output C.

The sampling in Theorem 2.2.2 satisfies the conditions for Lemma 2.3.1, hence $\mathbb{E}\,\|A'\|_F^2 \leq \big(1 + O(\frac{\epsilon^2}{\log(m+n)})\big)\|A\|_F^2$. 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 2.3.2 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

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

□

2.3.2 Proof of Theorem 2.1.5 (Frobenius-Norm AMM)

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

Lemma 2.3.3. *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 2.3.1, hence they satisfy $\mathbb{E}(a'b'^\top) = ab^\top$ and $\mathbb{E}(\|a'\|_2^2) \le (1+\epsilon^2/3)\|a\|_2^2$ and similarly for b'. Thus,

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

□

In order to prove Theorem 2.1.5, we will use a result from [DKM06]. The algorithm in [DKM06] 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 2.3.4 (Lemma 4 of [DKM06])**.** *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 $\mathbb{E}\,\|AB - C\|_F \le \epsilon\|A\|_F\|B\|_F$.*

Proof of Theorem 2.1.5. Let $0 < \epsilon < 1$. Recall that $AB = \sum_i A^i B_i$. Our algorithm is as follows.

1. Apply the algorithm in Lemma 2.3.3 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 2.3.4 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 2.3.3 and Cauchy-Schwarz inequality,

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

Additionally, by Jensen's inequality and Lemma 2.3.1,

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

and similarly for $\hat{B}$. By the triangle inequality and Lemma 2.3.4,

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

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. □

2.4 Application II: Preconditioning for Ridge Regression

Often, problem-specific preconditioners are used to reduce the condition number of the problem, since the time it takes for iterative methods to converge depends on the condition number. Specifically, for a matrix $M \in \mathbb{R}^{n\times n}$ and a linear-system $Mx = b$, any *invertible* matrix $P \in \mathbb{R}^{n\times n}$ has the property that the solution to the preconditioned linear-system $P^{-1}Mx = P^{-1}b$, is the same as that of the original problem. Using iterative methods to solve the preconditioned problem requires to apply $P^{-1}M$ to a vector in each iteration. In the case of ridge regression, $M = A^\top A + \lambda I$. Applying $(A^\top A + \lambda I)$ to a vector can be done in $O(\mathsf{nnz}(A))$ time, and applying P^{-1} to a vector is equivalent to solving a linear-system in P, i.e. $\arg\min_x \|Px - y\|_2^2$ for some $y \in \mathbb{R}^n$. There is a trade-off between the number of iterations taken to converge for the preconditioned problem, and the time taken to (approximately) solve a linear-system in P. We show in this section how to use the sparsification scheme of Theorem 2.1.2 to construct a preconditioner for ridge-regression, and couple it with an acceleration framework by [FGKS15].

Theorem 2.1.6. *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*

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

Since the term Ax is a linear combination of the columns of A, and the regularization term $\lambda\|x\|_2^2$ penalizes each coordinate of x equally, in practice, the columns of A are often pre-processed to have uniform norms before solving ridge-regression. For this case, in section 2.4.2, we show an improvement of roughly $(n/\,\mathsf{ns}(A))^{1/6}$ over Theorem 2.1.6.

We start by showing that given a matrix $A \in \mathbb{R}^{m\times n}$ and parameter $\lambda > 0$, if $P \in \mathbb{R}^{m\times n}$ is an ϵ-spectral-norm sparsifier for A, for small enough ϵ, the preconditioned problem has a constant condition number, hence requires only a constant number of iterations as described above. This was explored by [GS18], but they demanded ϵ to be $O(\frac{\lambda}{\|A\|_2^2})$, which is much smaller than necessary. In the next lemma we provide a tighter bound for ϵ.

Lemma 2.4.1. *Given matrix $A \in \mathbb{R}^{m\times n}$, parameters $\lambda > 0$ and $0 < \epsilon' < \frac{1}{2}$, then if a matrix $P \in \mathbb{R}^{m\times n}$ satisfies $\|A - P\|_2 < \epsilon\|A\|_2$ where $\epsilon = \frac{\sqrt{\lambda}\epsilon'}{\|A\|_2}$, then*

$$(1 - 2\epsilon')(A^\top A + \lambda I) \preceq P^\top P + \lambda I \preceq (1 + 2\epsilon')(A^\top A + \lambda I).$$

Setting $\epsilon' = 1/4$ yields that all the eigenvalues of $(P^\top P + \lambda I)^{-1}(A^\top A + \lambda I)$ are in the range $[\frac{2}{3}, 2]$. Using our sampling scheme in Theorem 2.1.2 with parameter ϵ as described here, denoting its output as P, provides a preconditioner for ridge regression with constant condition number. Hence solving this preconditioned problem, i.e, the linear-system $(P^\top P + \lambda I)^{-1}(A^\top A + \lambda I)x = (P^\top P + \lambda I)^{-1}b$ for some vector $b \in \mathbb{R}^n$, with any iterative method, takes $O_\epsilon(\mathsf{nnz}(A) + T_P^\lambda)$ time, where T_P^λ is the time it takes to compute an approximate solution to $\arg\min_x \|(P^\top P + \lambda I)x - y\|_2^2$ for some vector $y \in \mathbb{R}^n$.

Proof of Lemma 2.4.1. For any $x \in \mathbb{R}^n$, by the Triangle inequality,

$$\|Px\|_2 \le \|Ax\|_2 + \|(P - A)x\|_2 \le \|Ax\|_2 + \sqrt{\lambda}\epsilon'\|x\|_2.$$

By squaring both sides and applying the AM-GM inequality,

$$\begin{aligned}\|Px\|_2^2 &\le \|Ax\|_2^2 + \lambda\epsilon'^2\|x\|_2^2 + 2\|Ax\|_2\sqrt{\lambda}\epsilon'\|x\|_2 \\ &\le \|Ax\|_2^2 + \lambda\epsilon'^2\|x\|_2^2 + \epsilon'\left(\|Ax\|_2^2 + \lambda\|x\|_2^2\right) \\ &= (1+\epsilon')\|Ax\|_2^2 + \lambda\epsilon'(1+\epsilon')\|x\|_2^2,\end{aligned}$$

and since $\epsilon' < 1$,

$$\|Px\|_2^2 + \lambda\|x\|_2^2 \le (1+\epsilon')\|Ax\|_2^2 + \lambda(1+2\epsilon')\|x\|_2^2.$$

Hence $P^\top P + \lambda I \preceq (1+2\epsilon')(A^\top A + \lambda I)$.

Similarly, we get $\|Px\|_2 \ge \|Ax\|_2 - \sqrt{\lambda}\epsilon'\|x\|_2$, thus $\|Px\|_2^2 \ge (1-\epsilon')\|Ax\|_2^2 - \lambda\epsilon'(1-\epsilon')\|x\|_2^2$ and

$$\|Px\|_2^2 + \lambda\|x\|_2^2 \ge (1-\epsilon')\|Ax\|_2^2 + \lambda(1-2\epsilon')\|x\|_2^2.$$

□

2.4.1 Proof of Theorem 2.1.6

Solving the linear-system in $P^\top P + \lambda I$ can be done by the Conjugate Gradient (CG) method, and can be accelerated by the framework of [FGKS15], that, given an algorithm to compute an approximate solution to an Empirical Risk Minimization (ERM) problem, uses the algorithm to provide acceleration in a black-box manner. We restate the guarantees for these algorithms below.

Fact 2.4.2. *For a matrix $M \in \mathbb{R}^{m\times n}$, vector $y \in \mathbb{R}^m$ and parameters $\epsilon, \lambda > 0$, the Conjugate Gradient algorithm returns an ϵ-approximate solution to $\min_x \|Mx - y\|_2^2 + \lambda\|x\|_2^2$ in time $O(\mathsf{nnz}(M)\sqrt{\kappa_\lambda(M)}\ln(\frac{1}{\epsilon}))$, which we will denote by $T_{\mathsf{CG}}^{\lambda}(M,\epsilon)$.*

Lemma 2.4.3 (Acceleration. Theorem 1.1 of [FGKS15])**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a λ strongly convex function and for all $x_0 \in \mathbb{R}^n, c > 1, \lambda' > 0$, let $f_{min} = \min_{x\in\mathbb{R}^n}(f(x) + \frac{\lambda'}{2}\|x - x_0\|_2^2)$, assume we can compute $x_c \in \mathbb{R}^n$ in time T_c such that*

$$\mathbb{E}(f(x_c)) - f_{min} \le \frac{1}{c}(f(x_0) - f_{min}),$$

then, given any $x_0 \in \mathbb{R}^n, c > 1, \lambda' \ge 2\lambda$, we can compute x_1 such that

$$\mathbb{E}(f(x_1)) - \min_x(f(x)) \le \frac{1}{c}\big(f(x_0) - \min_x(f(x))\big)$$

in time $O\Big(T_{4(\frac{2\lambda'+\lambda}{\lambda})^{1.5}}\sqrt{\frac{\lambda'}{\lambda}}\log c\Big)$.

The measure of error in the above theorem coincides with the definition we gave for ϵ-approximation to ridge regression, since if $f(x) = \|Ax - b\|_2^2 + \lambda\|x\|_2^2$ and $x^* = \operatorname{argmin}_x f(X)$ then for any $x \in \mathbb{R}^n$, $(x - x^*)^T(A^TA + \lambda I)(x - x^*) = 2(f(x) - f(x^*))$. For a proof, see for example [MNS+18, Fact 39].

Note that the term $\frac{\lambda'}{2}\|x - x_0\|_2^2$ is not exactly of the same shape as the ridge term $\lambda'\|x\|_2^2$, but since

$$\|Ax - b\|_2^2 + \lambda\|x\|_2^2 + \lambda'\|x - x_0\|_2^2 = \|Ax - b\|_2^2 + (\lambda + \lambda')\|x\|_2^2 - 2\lambda' x_0^\top x + \lambda'\|x_0\|_2^2,$$

solving $\min_x(\|Ax-b\|_2^2+\lambda\|x\|_2^2+\lambda'\|x-x_0\|_2^2)$ is at most as hard as solving ridge regression with vector $A^Tb+\lambda' x_0$ and parameter $\lambda+\lambda'$. We are now ready to prove the result for preconditioned ridge-regression using our sparsifier as a preconditioner.

Proof of Theorem 2.1.6. We first explain how to compute an approximate solution for ridge regression with parameter $\lambda > 0$ and then apply the acceleration framework of Lemma 2.4.3 as a black-box.

Apply the sparsification scheme of Theorem 2.1.2 on A with parameter $\epsilon = \frac{\sqrt{\lambda}}{4\|A\|_2}$ as specified in Lemma 2.4.1 and denote its output by P. Solve the preconditioned linear-system $(P^\top P+\lambda I)^{-1}(A^\top A+\lambda I)x = (P^\top P+\lambda I)^{-1}b$ by any iterative method. As was described earlier, this takes $O_\epsilon(\mathsf{nnz}(A)+T_P^\lambda)$ time. Use Conjugate gradients to solve each linear-system in $P^\top P+\lambda I$. It takes $O_\epsilon(\sqrt{\kappa_\lambda(P^\top P)}\,\mathsf{nnz}(P))$ time. Since $\|P\|_2 \in (1\pm\epsilon)\|A\|_2$ and $\kappa_\lambda(P^\top P) = \frac{\|P\|_2^2}{\lambda}$, by Theorem 2.1.2,

$$T_{\mathsf{CG}}^\lambda(P,\epsilon) = O_\epsilon\Big(\mathsf{nnz}(P)\sqrt{\kappa_\lambda(P^\top P)}\Big) = O_\epsilon\Big(\frac{\|A\|_2^3}{\lambda^{1.5}}\,\mathsf{ns}(A)\,\mathsf{sr}(A)\ln n + \frac{\|A\|_2^2}{\lambda}\sqrt{\mathsf{ns}(A)n\cdot\mathsf{sr}(A)}\ln n\Big).$$

Applying the acceleration framework (Lemma 2.4.3) yields a running time of

$$\tilde{O}\bigg(\Big(\mathsf{nnz}(A)+\frac{\|A\|_2^3}{\lambda'^{1.5}}\,\mathsf{ns}(A)\,\mathsf{sr}(A)+\frac{\|A\|_2^2}{\lambda'}\sqrt{\mathsf{ns}(A)n\cdot\mathsf{sr}(A)}\Big)\sqrt{\frac{\lambda'}{\lambda}}\bigg).$$

Set $\lambda' = \|A\|_2^2(\frac{\mathsf{ns}(A)\,\mathsf{sr}(A)}{\mathsf{nnz}(A)})^{2/3}$. If $n < \mathsf{ns}(A)\,\mathsf{sr}(A)\frac{\|A\|_2^2}{\lambda'} = (\mathsf{ns}(A)\,\mathsf{sr}(A))^{1/3}(\mathsf{nnz}(A))^{2/3}$, which is a reasonable assumption in many cases (for example, if $\mathsf{nnz}(A) > n^{3/2}$), then this choice for λ' balances the two major terms, resulting in the stated running time. □

2.4.2 Faster Algorithm for Inputs with Uniform Row Norms

The best running time, to our knowledge, for the ridge-regression problem on sparse matrices in general is using Stochastic Variance Reduced Gradient Descent (SVRG), originally introduced by [JZ13], coupled with the acceleration framework of [FGKS15]. We utilize this method for solving the linear-system for $P^\top P+\lambda I$, where P is the preconditioner. This method is fastest if the norms of the rows/columns of the input matrix A are uniform. We show the following theorem for solving ridge-regression on numerically sparse matrices with uniform row/column norms.

Theorem 2.4.4. *There exists an algorithm that, given a matrix $A \in \mathbb{R}^{m\times n}$ having uniform rows norms or uniform columns norms, a vector $x_0 \in R^n$ and parameters $\lambda > 0, \epsilon > 0$, computes an ϵ-approximate solution to the ridge regression problem in expected time*

$$O_\epsilon(\mathsf{nnz}(A)) + \tilde{O}_\epsilon\Big(\mathsf{nnz}(A)^{2/3}\sqrt{\mathsf{sr}(A)}\,\mathsf{ns}(A)^{1/3}n^{-1/6}\sqrt{\kappa_\lambda(A^\top A)}\Big).$$

Note that $(A^\top A+\lambda I)^{-1}A^\top = A^\top(AA^\top+\lambda I)^{-1}$. Hence, for any vector v, one can compute an ϵ-approximation for $(A^\top A+\lambda I)^{-1}A^\top v$ in time $O(\mathsf{nnz}(A))+T^\lambda(A^\top,\epsilon)$. This doesn't change the condition number of the problem, i.e, $\kappa_\lambda(A^\top A)=\kappa_\lambda(AA^\top)$. Hence we only analyze the case where A is pre-processed such that the norms of the rows are uniform.

We provide a theorem from [MNS+18] that summarizes the running time of accelerated-SVRG.

Lemma 2.4.5 (Theorem 49 of [MNS+18])**.** *For a matrix M, vector $y \in \mathbb{R}^n$ and $\lambda, \epsilon > 0$, there exists an algorithm that computes with high probability an ϵ-approximate solution to $\min_x \|Mx - y\|_2^2 + \lambda\|x\|_2^2$ in time $T^\lambda(M, \epsilon)$ such that*

$$T^\lambda(M,\epsilon) \le O_\epsilon(\mathsf{nnz}(M)) + \tilde{O}_\epsilon\left(\sqrt{\mathsf{nnz}(M) \cdot \frac{\|M\|_F^2}{\lambda} \cdot \mathsf{rsp}(M)}\right).$$

Before we prove Theorem 2.4.4, note the following properties of the sampling in Theorem 2.1.2.

Lemma 2.4.6. *Given a matrix $A \in \mathbb{R}^{m\times n}$, parameter $\epsilon > 0$ and a random matrix $P \in \mathbb{R}^{m\times n}$ satisfying $\|P - A\|_2 \le \epsilon\|A\|_2$ and $\mathbb{E}\, P = A$, then the expected ℓ_2-norm of the i-th row and of the j-th column of P are bounded as*

$$\mathbb{E}\,\|P_i\|_2^2 \le \|A_i\|_2^2 + \epsilon^2\|A\|_2^2,$$
$$\mathbb{E}\,\|P^j\|_2^2 \le \|A^j\|_2^2 + \epsilon^2\|A\|_2^2.$$

Proof. By properties of the spectral-norm, $\|P_i - A_i\|_2 \le \|P - A\|_2 \le \epsilon\|A\|_2$. Squaring this and taking the expectation yields $\mathbb{E}(\|P_i\|_2^2) - \|A_i\|_2^2 \le \epsilon^2\|A\|_2^2$ as desired. The same holds for the columns. One can similarly get an high probability statement. □

Summing over all the rows or columns yields an immediate corollary,

Corollary 2.4.7. *The expected Frobenius-norm of P is bounded as $\mathbb{E}\,\|P\|_F^2 \le \|A\|_F^2 + \epsilon^2 \min(n, m)\|A\|_2^2$.*

We are now ready to show the result for ridge-regression in the case that the norms of the rows of the input matrix A are uniform.

Proof of Theorem 2.4.4. We first explain how to compute an approximate solution for ridge regression with parameter $\lambda > 0$ and then apply the acceleration framework of Lemma 2.4.3 as a black-box.

Apply the sparsification scheme of Theorem 2.1.2 on A with parameter $\epsilon = \frac{\sqrt{\lambda}}{4\|A\|_2}$ as specified in Lemma 2.4.1 and denote its output by P. Solve the preconditioned linear-system $(P^\top P + \lambda I)^{-1}(A^\top A + \lambda I)x = (P^\top P + \lambda I)^{-1}b$ by any iterative method. As was described earlier, this takes $O_\epsilon(\mathsf{nnz}(A) + T_P^\lambda)$ time. Use Accelerated-SVRG (Lemma 2.4.5) to solve each linear-system in $P^\top P + \lambda I$.

The bulk of the running time of the Accelerated-SVRG method is in applying vector-vector multiplication in each iteration, where one of the vectors is a row of P. The number of iterations have dependence on $\mathsf{sr}(P)$, which by Corollary 2.4.7 is bounded by $O(\mathsf{sr}(A) + \frac{n}{\kappa_\lambda})$. The running time of each iteration is usually bounded by the maximum row sparsity, i.e, $\mathsf{rsp}(P)$. Instead, we can bound the expected running time with the expected row sparsity, denote as $s^*(P)$. The distribution for sampling each row is $p_i = \frac{\|P_i\|_2^2}{\|P\|_F^2}$ [MNS+18]. Hence, the expected running time will depend on $\sum_i p_i\|P_i\|_0$ instead of $\mathsf{rsp}(P)$. By Lemma 2.4.6 and the assumption that the norms of the rows of A are uniform,

$$s^*(P) = \sum_i p_i\|P_i\|_0 \le \sum_i \frac{\|A_i\|_2^2 + \lambda}{\|P\|_F^2}\|P_i\|_0 \le \mathsf{nnz}(P)\Big(\frac{1}{n} + \frac{\lambda}{\|P\|_F^2}\Big) \tag{2.5}$$

Now, by Lemma 2.4.5, equation 2.5 and corollary 2.4.7,

$$\begin{aligned}
T^{\lambda}(\mathsf{P},\epsilon) &\le O_{\epsilon}\Big(\mathsf{nnz}(P) + \sqrt{\mathsf{nnz}(P) s^{*}(P)\,\mathsf{sr}(P)\cdot\kappa_{\lambda}(P^{\top}P)}\Big) \\
&\le O_{\epsilon}\Bigg(\mathsf{nnz}(P) + \mathsf{nnz}(P)\sqrt{\frac{\mathsf{sr}(P)\cdot\kappa_{\lambda}(P^{\top}P)}{n}+1}\Bigg) \\
&\le O_{\epsilon}\Bigg(\mathsf{nnz}(P) + \mathsf{nnz}(P)\sqrt{\frac{\mathsf{sr}(A)\cdot\kappa_{\lambda}(A^{\top}A)}{n}}\Bigg) \\
&\le O_{\epsilon}(\mathsf{nnz}(P)) + \tilde{O}_{\epsilon}\Big(\frac{\kappa_{\lambda}(A^{\top}A)^{3/2}\,\mathsf{ns}(A)\,\mathsf{sr}(A)^{3/2}}{\sqrt{n}}\Big).
\end{aligned}$$

The last inequality is by plugging in $\mathsf{nnz}(P)$ for the second term. Applying the acceleration framework (Lemma 2.4.3) to the preconditioned problem (i.e, P is a $\frac{c}{\sqrt{\kappa_{\lambda'}(A^{\top}A)}}$-spectral-norm sparsifier of A), yields running time of

$$\tilde{O}_{\epsilon}\Bigg(\mathsf{nnz}(A) + \Big(\mathsf{nnz}(A) + \frac{\kappa_{\lambda'}(A^{\top}A)^{3/2}\,\mathsf{ns}(A)\,\mathsf{sr}(A)^{3/2}}{\sqrt{n}}\Big)\sqrt{\frac{\lambda'}{\lambda}}\Bigg)$$

Setting $\lambda' = \frac{\|A\|_2^2\,\mathsf{ns}(A)^{2/3}\,\mathsf{sr}(A)}{n^{1/3}\,\mathsf{nnz}(A)^{2/3}}$ results in the stated running time. □

Chapter 3

Smoothness of Schatten Norms and Sliding-Window Matrix Streams

3.1 Introduction

Many modern data-sets are viewed as matrices that have millions or even billions of dimensions. Access to such large matrices is often done in a streaming fashion, meaning that the input is a sequence of items that can be read only sequentially, usually in one pass. Algorithms in this model are restricted to have small memory compared to the input size, and cannot access earlier input portions, hence they effectively have to compress the input.

In matrix streams, three settings are often considered. In all of them, there is an underlying matrix $A \in \mathbb{R}^{N\times m}$ initialized to the all-zeros matrix. In the row-order model, the input is a stream of the rows of A, i.e. vectors in $\mathbb{R}^m$ (see e.g. [Lib13]). In the entry-wise model, the input is a stream of non-zero entries of A in arbitrary order, and in the turnstile model, the input is a stream of additive updates to the entries of A, where these updates can also be negative.

Scenarios where the input is time-sensitive are often modeled by a sliding-window, meaning that at any point in time, the input is only the most recent n stream items (earlier items are ignored) called the active window [DGIM02]. Particularly, in row-order matrix streams, the input is a matrix made of the most recent n rows. Several linear-algebra problems were addressed in this model, e.g. covariance matrix approximation, PCA [WLL$^+$16, BDM$^+$20], spectral approximation, ℓ_1 subspace embedding and row/column subset selection [BDM$^+$20].

The spectrum of a matrix contains information related to many properties of the matrix (e.g, rank, condition number), and computing it is often a first step in data analysis. We focus on fundamental functions of the spectrum, called Schatten norms. Formally, for $p \geq 1$, the *Schatten p-norm* of a matrix $A \in \mathbb{R}^{n\times m}$ is

$$\|A\|_{S_p} = (\sum_i \sigma_i^p)^{1/p},$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{\min\{n,m\}} \geq 0$ are the singular values of A. This definition extends to $0 \leq p < 1$, for which it is not a norm. Special cases include $p = 0, 1, 2, \infty$, which are the rank, the trace norm, the Frobenius norm and the spectral/operator norm, respectively. Schatten norms can be used to estimate spectral-sum functions using Chebyshev polynomials, as explained in [KO19]; or to estimate the spectrum itself via the method of moments, as explained in [KV17]; or for matrix completion [NHD12]. There is a long line of work on computing Schatten p-norms in a stream, see [LNW14, LW16a, LW16b, BCK$^+$18, BKKS20b], and the further motivation provided therein.

We focus on approximating the Schatten p-norm of a sliding-window row-order stream. Previous work on this problem achieved $(\sqrt{2}+\epsilon)$-approximation [KR19], and as explained next, we provide the first $(1+\epsilon)$-approximation.[1]

3.1.1 Technical Contribution

A key algorithmic approach for sliding-window streams is the smooth-histograms technique of [BO07]. It is applicable to functions that (i) satisfy a certain smoothness criterion, and (ii) admit a one-pass $(1+\epsilon)$-approximation algorithm. The idea is to dynamically maintain several instances of that algorithm on different suffixes of the stream, altogether achieving $(1+\epsilon)$-approximation on the sliding-window [BO07]. Our main technical contribution is simply to prove that Schatten p-norms satisfy the smoothness criterion. An immediate corollary using the smooth-histograms technique of [BO07] is $(1+\epsilon)$-approximation algorithm for Schatten p-norm in the sliding-window row-order model, whose space requirement is comparable to the row-order one-pass (not sliding-window) setting. In contrast, previous work by [KR19] relied on relaxing the smoothness criterion of [BO07], which captures a more general family of functions, at the cost of a worse approximation factor $O(1)$.

Let us recall the smooth-histograms technique, instantiating it for our context of row-order matrix streams. Let X, Y, A and C denote real matrices with m columns and at most n rows (n is the size of the window), and interpret them also as row-order streams. Using this notation, the smoothness definition of [BO07, Definition 1] reads as follows.

Definition 3.1.1. A real-valued function f defined on real matrices is called (α, β)-smooth, where $0 < \beta \leq \alpha < 1$, if it satisfies the following.

1. Non-negative: for every real matrix A, it holds that $f(A) \geq 0$.
2. Non-decreasing: for all $A = \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$, it holds that $f(A) \geq f(X)$.
3. Bounded: for every matrix A with integral entries bounded by $\mathrm{poly}(n)$, it holds that $f(A) \leq \mathrm{poly}(n)$.
4. Smooth: for all real matrices $A = \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ and C, if $(1-\beta)f(A) \leq f(Y)$ then $(1-\alpha)f\left(\left[\begin{smallmatrix} A \\ C \end{smallmatrix}\right]\right) \leq f\left(\left[\begin{smallmatrix} Y \\ C \end{smallmatrix}\right]\right)$.

Assume that the entries of the matrices are integers bounded by $\mathrm{poly}(n)$.

Theorem 3.1.2. *[BO07, Theorem 3] Let f be an (α, β)-smooth function. If there is an algorithm Λ that maintains (ϵ, δ)-approximation of f on a stream using $g(\epsilon, \delta)$ bits of space and performing $h(\epsilon, \delta)$*

[1] A c-approximation to $x \in \mathbb{R}$ is $y \in \mathbb{R}$ such that $x \leq y \leq cx$ (if y is a random variable, then it should hold with high probability).

operations per stream item, then there exists an algorithm Λ' that maintains $(\alpha+\epsilon, \delta)$-approximation of f on sliding-windows using $O(\frac{1}{\beta}(g(\epsilon, \frac{\delta\beta}{\log n}) + \log n)\log n)$ bits of space and $O(\frac{1}{\beta}h(\epsilon, \frac{\delta\beta}{\log n})\log n)$ operations per item.[2]

We can now state our main technical result, that Schatten p-norms are smooth. Our proof generalizes the proof of [BO07, Lemma 4] that ℓ_p-norms are smooth, using matrix analysis tools that are based on pinching and monotonicity of the trace function.

Proposition 3.1.3. *For all $p \geq 2$ and $0 < \epsilon < 1$, the Schatten p-norm is $(\epsilon, \frac{\epsilon^{p/2}}{p/2})$-smooth.*

Proposition 3.1.4. *For all $0 < p \leq 2$ and $0 < \epsilon < 1$, the Schatten p-norm is (ϵ, ϵ)-smooth.*

These two propositions provide the same smoothness bound for $p = 2$, which is just the Frobenius norm. In the row-order model, computing the Frobenius norm is equivalent to computing the sum of positive numbers, which in turn is known to be (ϵ, ϵ)-smooth [BO07].

Our smoothness bounds match the known bounds for $\ell_{p/2}$-norms of vectors under insertions (no deletions) [BO07, Lemma 4], i.e. where the input is a stream of items $(i, \delta) \in [m] \times \mathbb{R}_+$, representing updates of the form $x_i \leftarrow x_i + \delta$ to a vector $x \in \mathbb{R}^m_+$. Further, our bounds are more general, since $\ell_{p/2}$-norms in this model can be simulated by Schatten p-norms in row-order streams, as follows. For an initial vector $x \in \mathbb{R}^m_+$, let $A \in \mathbb{R}^{m \times m}$ be a diagonal matrix with entries $\{\sqrt{x_i}\}_i$ on its diagonal. Then, $A^\top A$ is a diagonal matrix having the entries of x on its diagonal and $\|A\|^2_{S_p} = \|A^\top A\|_{S_{p/2}} = \|x\|_{p/2}$. Maintain this property of $A^\top A$, as follows. For an item (i, δ) with the update $x_i \leftarrow x_i + \delta$, where $\delta > 0$, update $A \leftarrow \begin{bmatrix} A \\ \sqrt{\delta} e_i^\top \end{bmatrix}$, where e_i is the i-th standard basis vector. Hence, the update to $A^\top A$ is $A^\top A \leftarrow A^\top A + \delta e_i e_i^\top$. Thus, the update to the diagonal of $A^\top A$ is the same as the update to x. This reduction shows that our smoothness bounds generalize the bounds of [BO07, Lemma 4] for $\ell_{p/2}$-norms (they proved that ℓ_q-norm is $(\epsilon, \epsilon^q/q)$-smooth for $q \geq 1$, and (ϵ, ϵ)-smooth for $0 < q < 1$).

3.1.2 Main Results

Our main result follows immediately from Propositions 3.1.3 and 3.1.4 using the smooth-histograms technique (Theorem 3.1.2). It shows that any one-pass algorithm to $(1+\epsilon)$-approximate the Schatten p-norm in a row-order stream, implies a sliding-window algorithm with almost the same space complexity.

Corollary 3.1.5. *Let Λ be a one-pass algorithm that maintains (ϵ, δ)-approximation of Schatten p-norm in row-order stream using $g(\epsilon, \delta)$ bits of space and performing $h(\epsilon, \delta)$ operations per stream item (i.e. a matrix row). Then there exists an algorithm Λ' that maintains $(2\epsilon, \delta)$-approximation of Schatten p-norm on sliding-windows such that:*

1. *if $p \geq 2$, it uses $O(\frac{p/2}{\epsilon^{p/2}}(g(\epsilon, \frac{2\delta\epsilon^{p/2}}{p\log n}) + \log n)\log n)$ bits of space and $O(\frac{p/2}{\epsilon^{p/2}}h(\epsilon, \frac{2\delta\epsilon^{p/2}}{p\log n})\log n)$ operations per stream item.*

2. *if $p < 2$, it uses $O(\frac{1}{\epsilon}(g(\epsilon, \frac{\delta\epsilon}{\log n}) + \log n)\log n)$ bits of space and $O(\frac{1}{\epsilon}h(\epsilon, \frac{\delta\epsilon}{\log n})\log n)$ operations per stream item.*

[2] (ϵ, δ)-approximation of f is $(1+\epsilon)$-approximation of f with probability at least $1 - \delta$.

Thus, every one-pass $(1+\epsilon)$-approximation algorithm for Schatten p-norm in row-order stream implies a sliding-window $(1+\epsilon)$-approximation algorithm with similar space requirement. Unfortunately, for $p<2$ no non-trivial algorithm is known; in fact, every one-pass $(1+\epsilon)$-approximation of Schatten p-norm of $n\times n$ matrices in row-order streams must use at least $\Omega(n^{1-g(\epsilon)})$ bits of space, where $g(\epsilon)\to 0$ as $\epsilon\to 0$ [BCK+18, Theorem 5.3 in arXiv version], and this lower bound extends immediately to the more restricted sliding-window model.

For $p>2$, there are two known one-pass (ϵ,δ)-approximation algorithms for Schatten p-norm of $n\times n$ matrices in row-order streams. One is for *Schatten* 4*-norm*, using $O(\epsilon^{-2}\log n)$ bits of space [BKKS20b, Theorem 7.2], and the other is for *sparse matrices, even* $p\ge 6$,[3] using $O_p(n^{1-4/\lceil p\rceil_4}\operatorname{poly}(\epsilon^{-1},\log n))$ bits of space [BCK+18, Theorem 6.1 and Section 1.3 in arXiv version].[4] These algorithms are stated for $n\times n$ matrices, but they immediately generalize to $n\times m$ matrices with $n>m$, and the polynomial dependence in the space bound is with respect to m (and not n). Plugging these algorithms into Corollary 3.1.5, we get the following.

Corollary 3.1.6. *There are algorithms that maintain (ϵ,δ)-approximation of Schatten p-norm in sliding-window row-order streams with the following space requirements.*

1. *for $p=4$: using $O(\epsilon^{-4}\log^2 n\log\frac{\log n}{\delta\epsilon})$ bits of space.*
2. *for even $p\ge 6$ and matrices with $O(1)$ non-zero entries in every row and column: using $O_p(n^{1-4/\lceil p\rceil_4}\operatorname{poly}(\epsilon^{-1},\log n,\log\delta))$ bits of space.*

Previously, only $(\sqrt{2}+\epsilon)$-approximation was known (with similar space requirement) [KR19].

3.2 Smoothness of Schatten Norms for $p\ge 2$

In this section, we prove Proposition 3.1.3, that Schatten p-norms for $p\ge 2$ in row-order streams are $(\epsilon,\frac{\epsilon^{p/2}}{p/2})$-smooth. [KR19] showed that Schatten norms are non-negative, non-decreasing and bounded. We complete this observation, and analyze the non-trivial property of Definition 3.1.1. Our proof is based on the proof for ℓ_q frequency moments for $q>1$ [BO07, Lemma 4]. We will need two auxiliary lemmas, as follows.

The first lemma has a simple proof using a pinching technique. It is given as an exercise by [Bha97], and we provide its proof for completeness.

Lemma 3.2.1. *[Bha97, Problem II.5.4.] For all $p\ge 2$, real matrices X,Y with the same row length and $A=\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right]$,*

$$\|A\|_{S_p}^p=\|X^TX+Y^TY\|_{S_{p/2}}^{p/2}\ge\|X^TX\|_{S_{p/2}}^{p/2}+\|Y^TY\|_{S_{p/2}}^{p/2}.$$

Proof. Note that

$$\|A\|_{S_p}^p=\|A^TA\|_{S_{p/2}}^{p/2}=\|AA^T\|_{S_{p/2}}^{p/2}=\left\|\begin{bmatrix}XX^T & XY^T\\ YX^T & YY^T\end{bmatrix}\right\|_{S_{p/2}}^{p/2}.$$

[3] A matrix is said to be sparse if there are $O(1)$ non-zero entries in every row/column.

[4] $\lceil p\rceil_4$ is the smallest multiple of 4 that is larger or equal to p.

Now, denote $Z = \begin{bmatrix} XX^T & XY^T \\ YX^T & YY^T \end{bmatrix}$ and $U = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$. Then, $\|A\|_{S_p}^p = \|Z\|_{S_{p/2}}^{p/2}$ and

$$\begin{bmatrix} XX^T & 0 \\ 0 & YY^T \end{bmatrix} = \frac{1}{2}(Z + UZU^T).$$

Thus, by the triangle inequality,

$$\begin{aligned} \|XX^T\|_{S_{p/2}}^{p/2} + \|YY^T\|_{S_{p/2}}^{p/2} &= \left\| \begin{bmatrix} XX^T & 0 \\ 0 & YY^T \end{bmatrix} \right\|_{S_{p/2}}^{p/2} \\ &\le \left(\tfrac{1}{2}(\|Z\|_{S_{p/2}} + \|UZU^T\|_{S_{p/2}})\right)^{p/2} = \|Z\|_{S_{p/2}}^{p/2}, \end{aligned}$$

where the last step holds since Schatten norms are unitarily invariant. □

The second lemma is a technical bound.

Lemma 3.2.2. *For all $0 < \epsilon \le 1$ and $p > 2$, it holds that $\sqrt{1 - (2\epsilon^{p/2} - \epsilon^p)^{2/p}} \ge 1 - \epsilon$.*

Proof. We begin by analyzing the function $f(\epsilon) = (2-\epsilon)^{p/2} + \epsilon^{p/2}$. Its derivative is

$$f'(\epsilon) = \tfrac{p}{2}(-(2-\epsilon)^{p/2-1} + \epsilon^{p/2-1}) < 0,$$

for the given range of ϵ. Hence it is decreasing, and its minimum is at $\epsilon = 1$, i.e. $f(\epsilon) \ge f(1) = 2$. Hence,

$$\begin{aligned} (2-\epsilon)^{p/2} \ge 2 - \epsilon^{p/2} &\implies (2\epsilon - \epsilon^2)^{p/2} \ge 2\epsilon^{p/2} - \epsilon^p \\ \implies 2\epsilon - \epsilon^2 \ge (2\epsilon^{p/2} - \epsilon^p)^{2/p} &\implies \sqrt{1 - (2\epsilon^{p/2} - \epsilon^p)^{2/p}} \ge 1 - \epsilon. \end{aligned}$$

□

We are ready to prove Proposition 3.1.3.

Proof of Proposition 3.1.3. Schatten norms are non-negative, non-decreasing and bounded [KR19, Corollary 3.9]. Let $p \ge 2$ and X, Y, C real matrices with rows of length m, as in Definition 3.1.1, such that $A = \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ satisfies $(1 - \frac{\epsilon^{p/2}}{p/2})\|A\|_{S_p} \le \|Y\|_{S_p}$. Since $\|A\|_{S_p}^p = \|A^\top A\|_{S_{p/2}}^{p/2}$, our goal is to prove that $(1-\epsilon)\|A^\top A + C^\top C\|_{S_{p/2}}^{1/2} \le \|Y^\top Y + C^\top C\|_{S_{p/2}}^{1/2}$. We have

$$\|Y\|_{S_p}^p \ge \left(1 - \tfrac{\epsilon^{p/2}}{p/2}\right)^p \|A\|_{S_p}^p \ge (1 - \epsilon^{p/2})^2 \|A\|_{S_p}^p = (1 - 2\epsilon^{p/2} + \epsilon^p)\|A\|_{S_p}^p.$$

By Lemma 3.2.1,

$$\|A^T A\|_{S_{p/2}}^{p/2} = \|X^T X + Y^T Y\|_{S_{p/2}}^{p/2} \ge \|X^T X\|_{S_{p/2}}^{p/2} + \|Y^T Y\|_{S_{p/2}}^{p/2}.$$

Hence

$$\|X\|_{S_p}^p \le (2\epsilon^{p/2} - \epsilon^p)\|A\|_{S_p}^p \le (2\epsilon^{p/2} - \epsilon^p)\|A^\top A + C^\top C\|_{S_{p/2}}^{p/2} \tag{3.1}$$

for any real matrix C. By the definition of A, triangle inequality and equation 3.1,

$$\begin{aligned}\|A^\top A + C^\top C\|_{S_{p/2}} &= \|X^\top X + Y^\top Y + C^\top C\|_{S_{p/2}} \\ &\leq \|Y^\top Y + C^\top C\|_{S_{p/2}} + \|X^\top X\|_{S_{p/2}} \\ &\leq \|Y^\top Y + C^\top C\|_{S_{p/2}} + (2\epsilon^{p/2} - \epsilon^p)^{2/p}\|A^\top A + C^\top C\|_{S_{p/2}}.\end{aligned}$$

Hence,

$$\|Y^\top Y + C^\top C\|_{S_{p/2}} \geq \big(1 - (2\epsilon^{p/2} - \epsilon^p)^{2/p}\big)\|A^\top A + C^\top C\|_{S_{p/2}}.$$

By Lemma 3.2.2, $\sqrt{1 - (2\epsilon^{p/2} - \epsilon^p)^{2/p}} \geq 1 - \epsilon$, which concludes the proof of Proposition 3.1.3. □

3.3 Smoothness of Schatten Norms for $p < 2$

In this section, we prove Proposition 3.1.4, that Schatten p-norms for $p \leq 2$ in row-order streams are (ϵ, ϵ)-smooth. As in the $p \geq 2$ case, recall that [KR19] showed that Schatten norms are non-negative, non-decreasing and bounded. The remaining part of the proof is based on the proof for ℓ_q frequency moments for $q \leq 1$ [BO07, Lemma 4].

To prove Proposition 3.1.4, we will need an auxiliary lemma. It is well known that if a function $f : \mathbb{R} \to \mathbb{R}$ is monotonically decreasing, then every two positive semidefinite (PSD) matrices $A \succeq B$ satisfy $\mathrm{Tr}[f(A)] \leq \mathrm{Tr}[f(B)]$. We need an analogous statement for the matrix function $f : X \mapsto (X + C^\top C)^q - X^q$ defined for PSD matrices X, where $C^\top C$ is a fixed PSD matrix. While the monotonicity of the trace function does not directly apply here, the desired monotonicity still holds, as summarized in the next lemma.

Lemma 3.3.1. *Let $C \in \mathbb{R}^{n\times m}$. For all $0 < q < 1$, if $A \succeq B \succeq 0$ then* $\mathrm{Tr}[(A + C^\top C)^q - A^q] \leq \mathrm{Tr}[(B + C^\top C)^q - B^q]$.

Proof. Denote the operator function $f(X) = (X + C^\top C)^q - X^q$ for $X \succeq 0$. Define

$$g(t) = \mathrm{Tr}\left[f\big(B + t(A - B)\big)\right] = \mathrm{Tr}\left[\big(B + t(A - B) + C^\top C\big)^q - \big(B + t(A - B)\big)^q\right].$$

Its derivative for $t \in [0, 1]$ is

$$\begin{aligned}g'(t) &= q\,\mathrm{Tr}\left[\big(B + t(A - B) + C^\top C\big)^{q-1}(A - B) - \big(B + t(A - B)\big)^{q-1}(A - B)\right] \\ &= q\,\mathrm{Tr}\left[(A - B)^{1/2}\Big(\big(B + t(A - B) + C^\top C\big)^{q-1} - \big(B + t(A - B)\big)^{q-1}\Big)(A - B)^{1/2}\right] \leq 0,\end{aligned}$$

where the last step is since $(B + t(A - B) + C^\top C)^{q-1} \prec (B + t(A - B))^{q-1}$ for $q < 1$, so the matrix inside the trace is negative semidefinite. Hence, $\mathrm{Tr}[f(A)] - \mathrm{Tr}[f(B)] = \int_0^1 g'(t)dt \leq 0$. □

Proof of Proposition 3.1.4. Schatten norms are non-negative, non-decreasing and bounded [KR19, Corollary 3.9]. Let $p \leq 2$ and let X, Y, C be real matrices as in Definition 3.1.1, such that $A = \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ satisfies

$$(1 - \epsilon)\|A\|_{S_p} \leq \|Y\|_{S_p}. \tag{3.2}$$

Note that

$$\left\| \begin{bmatrix} Y \\ C \end{bmatrix} \right\|_{S_p}^{p} - \|Y\|_{S_p}^{p} = \operatorname{Tr}\left[(Y^\top Y + C^\top C)^{p/2} - (Y^\top Y)^{p/2}\right]. \tag{3.3}$$

Now, by Lemma 3.3.1, and since $A^\top A = X^\top X + Y^\top Y \succeq Y^\top Y$, then

$$\operatorname{Tr}\left[(Y^\top Y + C^\top C)^{p/2} - (Y^\top Y)^{p/2}\right] \geq \operatorname{Tr}\left[(A^\top A + C^\top C)^{p/2} - (A^\top A)^{p/2}\right].$$

Combining this with equations 3.2 and 3.3, we get the desired result,

$$\left\| \begin{bmatrix} Y \\ C \end{bmatrix} \right\|_{S_p}^{p} \geq (1-\epsilon)^p \|A\|_{S_p}^{p} + (1-\epsilon)^p \operatorname{Tr}[(A^\top A + C^\top C)^{p/2} - (A^\top A)^{p/2}] = (1-\epsilon)^p \left\| \begin{bmatrix} A \\ C \end{bmatrix} \right\|_{S_p}^{p}.$$

□

Chapter 4

Bibliography

[AHK05] 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.

[AHK06] 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.

[AKL13] 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.

[AM07] Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9–es, 2007.

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, February 1999.

[AW21] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 522–539. SIAM, 2021.

[BCK$^+$18] Vladimir Braverman, Stephen Chestnut, Robert Krauthgamer, Yi Li, David Woodruff, and Lin Yang. Matrix norms in data streams: Faster, multi-pass and row-order. In *International Conference on Machine Learning*, pages 649–658. PMLR, 2018.

[BDM$^+$20] Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 517–528. IEEE, 2020.

[Bha97] Rajendra Bhatia. *Matrix analysis*, volume 169 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1997.

[BKKS20a] Vladimir Braverman, Robert Krauthgamer, Aditya Krishnan, and Shay Sapir. Near-optimal entrywise sampling of numerically sparse matrices. *arXiv preprint arXiv:2011.01777*, 2020.

[BKKS20b] Vladimir Braverman, Robert Krauthgamer, Aditya Krishnan, and Roi Sinoff. Schatten norms in matrix streams: Hello sparsity, goodbye dimension. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 1100–1110. PMLR, 2020.

[BO07] Vladimir Braverman and Rafail Ostrovsky. Smooth histograms for sliding windows. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 283–293. IEEE Computer Society, 2007.

[CCF02] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Automata, Languages and Programming, 29th International Colloquium, ICALP*, volume 2380, pages 693–703. Springer, 2002.

[CJST20] Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Coordinate methods for matrix games. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 283–293. IEEE, 2020.

[CNW16] 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.

[CW90] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3):251–280, 1990.

[CW09] 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.

[d'A11] Alexandre d'Aspremont. Subsampling algorithms for semidefinite programming. *Stochastic Systems*, 1(2):274–305, 2011.

[DGIM02] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining stream statistics over sliding windows. *SIAM Journal on Computing*, 31(6):1794–1813, 2002.

[DKM06] 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.

[DMM08] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, 2008.

[DZ11] 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.

[FGKS15] 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.

[FKV04] 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.

[GLP16] 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.

[GS18] 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.

[GT09] Alex Gittens and Joel A. Tropp. Error bounds for random matrix approximation schemes. *arXiv preprint arXiv:0911.4108*, 2009.

[GW18] Sumit Ganguly and David P. Woodruff. High probability frequency moment sketches. In *45th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 107 of *LIPIcs*, pages 58:1–58:15, 2018.

[Hoy04] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(9), 2004.

[HR09] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[Hua18] Zengfeng Huang. Near optimal frequent directions for sketching dense and sparse matrices. volume 80 of *Proceedings of Machine Learning Research*, pages 2048–2057, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, May 2006.

[IW05] Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208, 2005.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[KD14] Abhisek Kundu and Petros Drineas. A note on randomized element-wise matrix sparsification. *arXiv preprint arXiv:1404.0320*, 2014.

[KDMI17] 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.

[KNW10] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1161–1178, 2010.

[KO19] Ashish Khetan and Sewoong Oh. Spectrum estimation from a few entries. *The Journal of Machine Learning Research*, 20:21:1–21:55, 2019.

[KR19] Robert Krauthgamer and David Reitblat. Almost-smooth histograms and sliding-window graph algorithms. *arXiv preprint arXiv:1904.07957*, 2019.

[KS21] Robert Krauthgamer and Shay Sapir. Smoothness of Schatten norms and sliding-window matrix streams. *arXiv preprint arXiv:2103.08211*, 2021.

[KV17] Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *The Annals of Statistics*, 45(5):2218 – 2247, 2017.

[Lib13] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 581–588, 2013.

[LNW14] Yi Li, Huy L. Nguyen, and David P. Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1562–1581. SIAM, 2014.

[Lop13] Miles Lopes. Estimating unknown sparsity in compressed sensing. In *International Conference on Machine Learning*, pages 217–225, 2013.

[LW16a] Yi Li and David P. Woodruff. On approximating functions of the singular values in a stream. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 726–739. ACM, 2016.

[LW16b] Yi Li and David P. Woodruff. Tight bounds for sketching the operator norm, Schatten norms, and subspace embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 39:1–39:11, 2016.

[MD09] Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proc. Natl. Acad. Sci. USA*, 106(3):697–702, 2009.

[MMG17] Youssef Mroueh, Etienne Marcheret, and Vaibahava Goel. Co-occurring directions sketching for approximate matrix multiply. In *Artificial Intelligence and Statistics*, pages 567–575. PMLR, 2017.

[MNS$^+$18] Cameron Musco, Praneeth Netrapalli, Aaron Sidford, Shashanka Ubaru, and David P. Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPICS)*, pages 8:1–8:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[MZ11] 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.

[NDT15] 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.

[NHD12] Feiping Nie, Heng Huang, and Chris Ding. Low-rank matrix recovery via efficient Schatten p-norm minimization. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, page 655–661. AAAI Press, 2012.

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, December 2011.

[Tro12] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[WLL+16] Zhewei Wei, Xuancheng Liu, Feifei Li, Shuo Shang, Xiaoyong Du, and Ji-Rong Wen. Matrix sketching over sliding windows. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1465–1480. ACM, 2016.

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1–2):1–157, October 2014.

[YLZ16] 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.