A Note on Compressed Sensing and the Complexity of Matrix Multiplication

M. A. Iwen and C. V. Spencer

Institute for Mathematics and its Applications, and Institute for Advanced Study
iwen@ima.umn.edu, and cvspenc@math.ias.edu

Abstract

We consider the conjectured $O(N^{2+\epsilon})$ time complexity of multiplying any two $N \times N$ matrices A and B. Our main result is a deterministic Compressed Sensing (CS) algorithm that both rapidly and accurately computes $A \cdot B$ provided that the resulting matrix product is sparse/compressible. As a consequence of our main result we increase the class of matrices A, for any given $N \times N$ matrix B, which allows the exact computation of $A \cdot B$ to be carried out using the conjectured $O(N^{2+\epsilon})$ operations. Additionally, in the process of developing our matrix multiplication procedure, we present a modified version of Indyk's recently proposed extractor-based CS algorithm [12] which is resilient to noise.

Key words: algorithms, analysis of algorithms, approximation algorithms, computational complexity

1 Introduction

Multiplying two arbitrary $N \times N$ matrices requires $\Omega(N^2)$ operations (e.g., to read the input matrices). Naive multiplication of two $N \times N$ matrices uses $\Theta(N^3)$ operations. It is conjectured that for any $\epsilon > 0$, one can multiply two $N \times N$ matrices with $O(N^{2+\epsilon})$ operations, and this result would follow from various combinatorial and algebraic conjectures [4,7].

Recent approaches to matrix multiplication include the use of tensor product constructions to produce algorithms to multiply two large matrices. The current best algorithm for multiplying two $N \times N$ matrices [7] combines tensor product constructions with a result from additive combinatorics due to Salem and D. C. Spencer [17] to derive an algorithm requiring $O(N^{2.376})$ operations. For a survey of matrix multiplication complexity and related geometry results see [15]. In this paper, we generalize the following theorem of Coppersmith (see [6]).

Theorem 1 *Let $\beta = .29462...$ and $\epsilon > 0$. One can multiply matrices of size $N \times N$ and $N \times N^{\beta}$ with complexity $O(N^{2+\epsilon})$.*

Theorem 1 provides the current best result in terms of maximizing the number of rows, m, an $m \times N$ matrix may have while still being able to be multiplied by another $N \times N$ matrix with complexity $O(N^{2+\epsilon})$. In the next section we present an algorithm for computing the product of two $N \times N$ matrices using $O(N^{2+\epsilon})$ operations under the assumption that the product is sparse in each column. As a result, we generalize Theorem 1 with respect to the types of $N \times N$ matrices A we may multiply by any given $N \times N$ matrix B with the conjectured complexity.

2 Preliminaries

Throughout the remainder of this paper we will utilize the standard Frobenius matrix norm. Let A be an $N \times N$ complex-valued matrix. A's Frobenius norm, $\|A\|_{\mathrm{F}}$, is defined as

$$\|A\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} |A_{i,j}|^2}. \tag{1}$$

Here $A_{i,j}$ is A's i^{th} row's j^{th} entry. Similarly, A_i will denote A's i^{th} row and A^j will denote A's j^{th} column.

Our main result deals with compressible matrices (i.e., matrices which consist of a sparse representation contaminated with additional noise terms). We say that a complex-valued vector, $\mathbf{v} \in \mathbb{C}^N$, is (C, γ)-compressible for fixed $C, \gamma \in \mathbb{R}^+$, if there exists an ordering of $\mathbf{v}$'s elements by magnitude,

$$|\mathbf{v}_{j_1}| \geq \ldots \geq |\mathbf{v}_{j_m}| \geq \ldots \geq |\mathbf{v}_{j_N}|, \tag{2}$$

such that $|\mathbf{v}_{j_l}| \leq C \cdot 2^{-\gamma \cdot l}$ for all $1 \leq l \leq N$. Furthermore, we will say that a vector containing only k nonzero-elements, $\mathbf{u}_k^{\text{opt}}$, is k-optimal with respect to vector $\mathbf{v}$ if

$$\|\mathbf{v} - \mathbf{u}_k^{\text{opt}}\|_2^2 = \sum_{l=k+1}^{N} |\mathbf{v}_{j_l}|^2 = O\left(\frac{C^2}{\gamma} \cdot 4^{-\gamma \cdot k}\right). \tag{3}$$

Note that the k-optimal error

$$\|\mathbf{v} - \mathbf{u}_k^{\text{opt}}\|_2^2 \tag{4}$$

is unique for each $v \in \mathbb{C}^N$. Finally, we will say that an $N \times N$ complex-valued matrix A is *compressible*, or (C, γ)-compressible, if all of A's column (or row) vectors are (C, γ)-compressible. For a compressible $N \times N$ matrix A, we will let U_k^{opt} denote any $N \times N$ matrix minimizer of

$$\|A - U_k\|_{\mathrm{F}}^2 \tag{5}$$

over the class of matrices containing $\leq k$ non-zero entries per column (or row). Without loss of generality we will assume column compressibility from now on.

2.1 Compressed Sensing

Let $\mathbf{v} \in \mathbb{C}^N$ and Ψ be a complex-valued $N \times N$ matrix. Furthermore, suppose that $\Psi \cdot \mathbf{v}$ is sparse/compressible (e.g., (C, γ)-compressible). Compressed Sensing (CS) methods provide a $K \times N$ measurement matrix, $\mathcal{M}$, with K minimized such that the k most significant entries of $\Psi \cdot \mathbf{v}$ can be recovered from the K-element result of

$$\mathcal{M} \cdot \Psi \cdot \mathbf{v}. \tag{6}$$

Standard algorithms for recovering/approximating $\Psi \cdot \mathbf{v}$'s largest k entries in magnitude from the result of (6) include linear programming [9,3], orthogonal matching pursuit [18], and various faster algorithms [11,16,8,14,12] for particular types of measurement matrices $\mathcal{M}$. For the purposes of this paper we will utilize a variant of Theorem 2 (proved in [12]).

Theorem 2 *Suppose that the vector $\Psi \cdot \boldsymbol{v} \in \mathbb{C}^N$ contains $\leq k$ non-zero elements. There exists a $k \cdot 2^{O(\log^2 \log N)} \times N$ measurement matrix, $\mathcal{M}$, which enables the exact reconstruction of $\Psi \cdot \boldsymbol{v}$ from the $k \cdot 2^{O(\log^2 \log N)}$-element result of $\mathcal{M} \cdot \Psi \cdot \boldsymbol{v}$ in $k \cdot 2^{O(\log^2 \log N)}$ time.*

We concentrate on Theorem 2 for two reasons. First, the reconstruction method outlined in [12] has a runtime complexity that is both sublinear in N (the vector dimension) and linear in k (the sparsity level). All deterministic variants of [9,3,18,16,8,14] utilize reconstruction algorithms which are superlinear in either N, k, or both. Furthermore, unlike fast CS methods with uniform error guarantees (e.g., [11]), Indyk's method is both deterministic and explicit (i.e., there is no probability of failure). Although the uniformly random guarantees in [11] suffice to demonstrate the existence of deterministic matrix multiplication algorithms, verifying any such algorithm's correctness over all sparse signals is computationally intractable.

3 Approximating Matrix Products

In this section we discuss how the combination of compressed sensing methods with Coppersmith's work (i.e., Theorem 1) can be used to (approximately) multiply two $N \times N$ matrices with $O(N^{2+\epsilon})$ operations when the product of the two matrices is known to be sparse/compressible. However, in order to state our simple CS based matrix multiplication method we must utilize a noise tolerant version of

Theorem 2. By modifying Indyk's recovery algorithm and measurement construction the following result can be obtained.

Theorem 3 *Suppose that* $\boldsymbol{v} \in \mathbb{C}^N$, Ψ *is a complex-valued* $N \times N$ *matrix, and* $\Psi \cdot \boldsymbol{v}$ *is* (C, γ)*-compressible. Then, we may construct a* $\left(m + \frac{1}{\gamma}\right) \cdot 2^{O(\log^2 \log N)} \times N$ *measurement matrix,* $\mathcal{M}$*, which allows a* $\left(m + \frac{1}{\gamma}\right) \cdot 2^{O(\log^2 \log N)}$*-time reconstruction algorithm to use the result of* $\mathcal{M} \cdot \Psi \cdot \boldsymbol{v}$ *and return a vector* $\boldsymbol{u}_m$ *such that*

$$\| \Psi \cdot \boldsymbol{v} - \boldsymbol{u}_m \|_2^2 \quad \leq \quad \| \Psi \cdot \boldsymbol{v} - \boldsymbol{u}_m^{\text{opt}} \|_2^2 + \frac{\left| (\Psi \cdot \boldsymbol{v})_{j_{m+1}} \right|^2}{N}.$$

Here, $\left| (\Psi \cdot \boldsymbol{v})_{j_{m+1}} \right|$ *is the magnitude of the product's* $(m+1)^{\text{st}}$*-largest entry/entries.*

Theorem 3's proof is analogous to Theorem 2's proof, modulo complications due to the presence of 'noise' (i.e., the exponentially decaying smaller magnitude entries of $\Psi \cdot \mathbf{v}$). Due to the proof's similarity to the work in [12] we will only sketch it here.

Proof Sketch:

If we want to recover the m largest magnitude entries of $\Psi \cdot \mathbf{v}$ we will substitute

$$m + O\left(\frac{\log^2 N + \log\left(\frac{C}{\gamma}\right)}{\gamma}\right) \tag{7}$$

for r (i.e., the sparsity level) everywhere in [12]. Furthermore, instead of using [8]'s explicit CS construction we can just as easily use the related construction/theorems in [14]. Thus, complex values are easily handled and each non-overflowing H row can recover entries with enough accuracy to yield results along the lines of [14]'s Theorems 2 and 3 (exponential decay).

We will consider the vector we want to recover, $\Psi \cdot \mathbf{v}$, to consist of an exact r-sparse vector (containing a few more than the m largest magnitude entries we ultimately want to recover — see Equation 7) plus a noise vector containing all the remaining entries (i.e., the exponentially decaying 'noise'). As long as the sum of all the noise terms is small enough, Indyk's algorithm will work as before after it is modified as follows:

First, we must modify [12]'s Reduce procedure by replacing the line

"IF $votes[j]$ CONTAINS $> d_A/2$ COPIES OF val THEN $y_j = val$"

with

"IF $|votes[j]| > 2d_A/3$ THEN $\mathbb{Re}(y_j)$ = MEDIAN OF $\mathbb{Re}(votes[j])$ AND $\mathbb{Im}(y_j)$ = MEDIAN OF $\mathbb{Im}(votes[j])$".

This changes the proof of [12]'s Lemma 1 only in that now $d_A/3$ vote changes are needed to make any entry y_j have a value more than the current cumulative noise level from the true value (e.g., more than $O(2^{-\gamma \cdot m} \cdot N^{-2})$ from the correct value in the final iterative call of Reduce). Thus, if we set $\epsilon < 1/24$, more than half of the r-sparse portion of our input vector will be replaced by bounded noise after each iteration.

Second, we note that the iterative nature of Indyk's Recover procedure won't degrade our final accuracy. Each iteration of Reduce can multiply the additive noise for every recovered entry by no more than N, resulting in Recover returning an estimate y_{j_l} for each largest magnitude entry $(\Psi \cdot \mathbf{v})_{j_l}$, $1 \leq l \leq m$, with

$$|y_{j_l} - (\Psi \cdot \mathbf{v})_{j_l}| = N^{O(\log N)} \cdot \left(\sum_{n=r+1}^{N} \left| (\Psi \cdot \mathbf{v})_{j_n} \right| \right) = N^{O(\log N)} \cdot \frac{C \cdot 2^{-\gamma \cdot r}}{\gamma}. \tag{8}$$

If r is replaced with Equation 7 we can maintain the additive error bounds needed by [14]'s recovery algorithm to maintain its required accuracy during all $O(\log N)$ iterative calls of the Reduce procedure.

Finally, after we collect the output from the Recover procedure, we sort the output entries by their magnitude and return the largest m of them as our sparse representation $\mathbf{u}_m$. Because we are able to maintain the required accuracy of Recover's output (see preceding paragraph), an argument analogous to the proof of [14]'s Theorem 2 will give us our final result. □

With Theorem 3 in hand we are ready to consider matrix multiplication. Let A and B denote two $N \times N$ matrices with complex entries. Furthermore, we suppose that $A \cdot B$ is (C, γ)-compressible. To construct an approximate product matrix U_m with

$$||A \cdot B - U_m||_{\mathrm{F}} = O\left(||A \cdot B - U_m^{\mathrm{opt}}||_{\mathrm{F}} \right) \tag{9}$$

we proceed as follows:

(1) Use a $\left(m + \frac{1}{\gamma}\right) \cdot 2^{O(\log^2 \log N)} \times N$ measurement matrix, $\mathcal{M}$, as per Theorem 3 to compute

$$P = (\mathcal{M} \cdot A) \cdot B \tag{10}$$

using Theorem 1. Provided that there exists some $\delta > 0$ so that both m and $\frac{1}{\gamma}$ are $O(N^{\beta - \delta})$ this can be accomplished in $O(N^{2+\epsilon})$ time.

(2) Apply Theorem 3 to P^j for all $1 \leq j \leq N$ to recover U_m.

The total recovery time will be $O(N^{1+\beta})$. We quickly obtain our main theorem.

Theorem 4 *Let $\beta^- < .29462...$, $\epsilon > 0$, and A, B be $N \times N$ matrices. If $A \cdot B$ is (C, γ)-compressible and both m and $\frac{1}{\gamma}$ are $O(N^{\beta^-})$, then one can obtain an $N \times N$ matrix U_m such that*

$$\|(A \cdot B) - U_m\|_{\mathrm{F}}^2 \leq \|(A \cdot B) - U_m^{\mathrm{opt}}\|_{\mathrm{F}}^2 + \sum_{i=1}^{N} \frac{\left|(A \cdot B)^i_{j_{m+1}}\right|^2}{N}$$

in $O(N^{2+\epsilon})$ time.

Note that in the special case where $A \cdot B$ has $\leq m$ non-zero elements per column, we have

$$\sum_{i=1}^{N} \frac{\left|(A \cdot B)^i_{j_{m+1}}\right|^2}{N} = 0. \tag{11}$$

We obtain the following corollary.

Corollary 5 *Let $\beta^- < .29462...$ and $c, \epsilon \in \mathbb{R}^+$. Furthermore, let A and B denote $N \times N$ matrices. If the product $A \cdot B$ has at most cN^{β^-} non-zero elements in each column, then $A \cdot B$ can be computed using $O(N^{2+\epsilon})$ operations.*

Let A and B denote square $N \times N$ matrices, $\epsilon > 0$, and $c > 0$. If $A \cdot B$ is compressible in each column, we can use Theorem 4 to obtain a near-optimal best $cN^{.29462}$ element-per-column approximation to $A \cdot B$ using $O(N^{2+\epsilon})$ operations. More specifically, if each column of the product $A \cdot B$ has at most $cN^{.29462}$ non-zero elements, then we can use Corollary 5 to calculate the product $A \cdot B$ exactly using $O(N^{2+\epsilon})$ operations.

4 Discussion

In this paper we discussed how compressed sensing methods can be used to (approximately) multiply two square matrices quickly if the product is known to be sparse. In the process, we have increased the class of $N \times N$ matrices A, for any given $N \times N$ matrix B, which allow $A \cdot B$ to be calculated exactly using $O(N^{2+\epsilon})$ operations (see Corollary 5). Provided that $A \cdot B$ contains at most $O(N^{\beta^-})$ non-zero entries per column, it can be calculated exactly using $O(N^{2+\epsilon})$ operations. In contrast, previous results [5,6] required that A contain $O(N^{\beta})$ non-empty (i.e., non-sparse) rows to achieve the same bound.

Furthermore, we have also provided results concerning the approximation of the product of two (dense) $N \times N$ matrices in $O(N^{2+\epsilon})$ time. Any two matrices may be

approximately multiplied using our method, and the result will be accurate to the extent that the true product is compressible. The required measurement acquisition (i.e., Equation 10) can either be accomplished via traditional matrix multiplication or via lower complexity methods (e.g., Theorem 1). In the later case It is worth mentioning that any additional advances in rapid matrix multiplication similar to Theorem 1 will automatically strengthen our results. This is due to the reconstruction algorithm in Theorem 3 having $O(m \cdot N^{\epsilon})$ runtime.

We finish by noting that in practice we may not know when a matrix product is going to be column/row-sparse. Thus, although we have given a deterministic algorithm which is guaranteed to accurately approximate such products, we won't necessarily know *when* our answers are accurate. Existing streaming algorithm techniques [10,1] allow us to predict the sparsity (i.e., number of non-zero entries) of all the matrix product's columns/rows to within a small constant factor (e.g., 4) with probability $\left(1 - \frac{1}{N^{O(1)}}\right)$ in $O(N^2 \cdot \log N)$-time [13]. Thus, in the general case (where the matrix product's sparsity is unknown) a Monte-Carlo variant of Corollary 5 holds.

5 Acknowledgments

We would like to thank Anna Gilbert, Piotr Indyk, S. Muthukrishnan, and Martin Strauss for helpful comments. This work was supported in part by NSF DMS-0510203.

References

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

[2] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, *Combining Geometry and Combinatorics: A Unified Approach to Sparse Signal Recovery*, preprint, 2008.

[3] E. Candes, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52:489–509, 2006.

[4] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans, *Group-theoretic algorithms for matrix multiplication*, preprint.

[5] D. Coppersmith, *Rapid multiplication of rectangular matrices*, SIAM J. Comput. **11** (1982), 467-471.

[6] D. Coppersmith, *Rectangular matrix multiplication revisited*, J. Complexity **13** (1997), 42-49.

[7] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput. **9** (1990), 251-280.

[8] G. Cormode and S. Muthukrishnan, *Combinatorial Algorithms for Compressed Sensing*, CISS, March 2006.

[9] D. Donoho, *Compressed sensing*, IEEE Trans. on Information Theory, 52:1289 – 1306, 2006.

[10] P. Flajolet, and G. Martin, *Probabilistic counting algorithms for data base applications*, J. of Comput. and System Sci., 31:182-209, 1985.

[11] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, *Algorithmic Linear Dimension Reduction in the l_1 Norm for Sparse Vectors*, Submitted, 2006.

[12] P. Indyk, *Explicit Constructions for Compressed Sensing of Sparse Signals*, SODA, 2008.

[13] P. Indyk, Personal Correspondence, 2008.

[14] M. A. Iwen, *A deterministic sub-linear time sparse fourier algorithm via non-adaptive compressed sensing methods*, SODA, 2008.

[15] J. M. Landsberg, *Geometry and the complexity of matrix multiplication*, Bulletin of the American Mathematical Society, vol. 45, no. 2, April 2008.

[16] S. Muthukrishnan, *Some Algorithmic Problems and Results in Compressed Sensing*. Allerton Conference, 2006.

[17] R. Salem and D. C. Spencer, *On sets of integers which contain no three terms in arithmetical progression*, Proc. Nat. Acad. Sci. USA **28** (1942), 561-563.

[18] J. Tropp and A. Gilbert, *Signal recovery from partial information via orthogonal matching pursuit*, Submitted for Publication, 2005.