Available online at www.sciencedirect.com

ScienceDirect

Appl. Comput. Harmon. Anal. 25 (2008) 335–366

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

A fast randomized algorithm for the approximation of matrices ☆

Franco Woolfe, Edo Liberty, Vladimir Rokhlin, Mark Tygert *

Program in Applied Mathematics, Yale University, A.K. Watson Hall, 51 Prospect St., New Haven, CT 06511, USA

Received 21 August 2007; accepted 6 December 2007

Available online 23 December 2007

Communicated by Naoki Saito

Abstract

We introduce a randomized procedure that, given an $m \times n$ matrix A and a positive integer k, approximates A with a matrix Z of rank k. The algorithm relies on applying a structured $l \times m$ random matrix $\mathcal{R}$ to each column of A, where l is an integer near to, but greater than, k. The structure of $\mathcal{R}$ allows us to apply it to an arbitrary $m \times 1$ vector at a cost proportional to $m \log(l)$; the resulting procedure can construct a rank-k approximation Z from the entries of A at a cost proportional to $mn \log(k) + l^2(m+n)$. We prove several bounds on the accuracy of the algorithm; one such bound guarantees that the spectral norm $\|A - Z\|$ of the discrepancy between A and Z is of the same order as $\sqrt{\max\{m, n\}}$ times the $(k+1)$st greatest singular value σ_{k+1} of A, with small probability of large deviations.

In contrast, the classical pivoted "QR" decomposition algorithms (such as Gram–Schmidt or Householder) require at least kmn floating-point operations in order to compute a similarly accurate rank-k approximation. In practice, the algorithm of this paper runs faster than the classical algorithms, even when k is quite small or large. Furthermore, the algorithm operates reliably independently of the structure of the matrix A, can access each column of A independently and at most twice, and parallelizes naturally. Thus, the algorithm provides an efficient, reliable means for computing several of the greatest singular values and corresponding singular vectors of A. The results are illustrated via several numerical examples.

Keywords: Fast; Randomized; Algorithm; Low rank; Matrix; SVD; QR; Lanczos

1. Introduction

The construction of low-rank approximations to matrices is very important in many applications of numerical computation. Such approximations help to characterize the structure of linear operators, and to facilitate rapid calculations involving them. One classical form of these approximations is the singular value decomposition (SVD), which is known in the statistical literature as the principal component analysis (PCA). Another classical form is the approximation obtained via subset selection; we will refer to the matrix representation obtained via subset selection as an interpolative decomposition. These two types of matrix approximations are defined as follows.

☆ Partially supported by NGA Grants HM1582-06-1-2039 and HM1582-06-1-2037, DARPA Grant FA9550-07-1-0541, and AFOSR Grants FA9550-06-1-0197, FA9550-06-1-0239, and FA9550-05-C-0064.

* Corresponding author.

E-mail addresses: francis.woolfe@yale.edu (F. Woolfe), edo.liberty@yale.edu (E. Liberty), mark.tygert@yale.edu (M. Tygert).

1063-5203/$ – see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.acha.2007.12.002

An approximation to an SVD of a complex $m \times n$ matrix A consists of nonnegative real numbers $\sigma_1, \sigma_2, \ldots, \sigma_{k-1}, \sigma_k$ known as singular values, orthonormal complex $m \times 1$ column vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k-1)}, u^{(k)}$ known as left singular vectors, and orthonormal complex $n \times 1$ column vectors $v^{(1)}, v^{(2)}, \ldots, v^{(k-1)}, v^{(k)}$ known as right singular vectors, such that

$$\left\| A - \sum_{j=1}^{k} u^{(j)} \sigma_j \bigl(v^{(j)}\bigr)^* \right\| \leqslant \delta, \tag{1}$$

where k, m, and n are positive integers with $k \leqslant m$ and $k \leqslant n$, δ is a positive real number specifying the precision of the approximation, and, for any matrix B, $\|B\|$ denotes the spectral (l^2-operator) norm of B, that is, $\|B\|$ is the greatest singular value of B. An approximation to an SVD of A is often written in the equivalent form

$$A \approx U \Sigma V^*, \tag{2}$$

where U is a complex $m \times k$ matrix whose columns are orthonormal, V is a complex $n \times k$ matrix whose columns are orthonormal, and Σ is a real diagonal $k \times k$ matrix whose entries are all nonnegative. See, for example, [15] for a more detailed discussion of SVDs.

An interpolative decomposition of a complex $m \times n$ matrix A consists of a complex $m \times k$ matrix B whose columns constitute a subset of the columns of A, and a complex $k \times n$ matrix P, such that

1. some subset of the columns of P makes up the $k \times k$ identity matrix,
2. no entry of P has an absolute value greater than 2, and
3. $A = BP$.

See, for example, [5,9,12–14,19], or Sections 4 and 5 of [4] for a discussion of interpolative decompositions.

The present article introduces an algorithm for the computation of a low-rank approximation of either type to an arbitrary matrix. The algorithm is generally at least as efficient as pivoted Gram–Schmidt and the other classical pivoted "QR" decomposition algorithms, and often substantially more efficient. In order to construct a nearly optimal rank-k approximation to a complex $n \times n$ matrix, any of the standard schemes (such as Gram–Schmidt or Householder) requires at least

$$kn^2 \tag{3}$$

floating-point operations (see, for example, Chapter 5 in [8]). In contrast, the algorithm of Section 5.2 of the present paper requires

$$\mathcal{O}\bigl(n^2 \log(k) + nl^2\bigr) \tag{4}$$

floating-point operations, where l is an integer near to, but greater than, k.

In practice, the scheme of the present article is more efficient than pivoted "QR" decomposition algorithms whenever $l < n$ and k is not extremely small (see Section 6 below). Furthermore, the scheme of the present paper requires less storage whenever the input matrix is to be preserved, applies naturally to matrices whose entries are to be evaluated on-the-fly, rather than stored in memory, and parallelizes trivially. Thus, the algorithm described below would seem to be preferable to the classical algorithms for the construction of low-rank approximations to medium- and large-scale dense matrices, or (more or less equivalently) for the computation of a few of the greatest singular values of matrices and their corresponding singular vectors.

Unlike the classical algorithms, the scheme of the present paper is a randomized one, and fails with a small probability. However, one can determine rapidly whether the algorithm has succeeded, using a verification scheme such as that described in Section 3.4. If the algorithm were to fail, then one could run the algorithm again with an independent realization of the random variables involved, in effect boosting the probability of success at a reasonable additional expected cost. In fact, the randomized algorithm succeeded during every trial reported in the numerical experiments of Section 6, obviating the need to run the algorithm again.

The algorithm of [14] is similar to the algorithm of the present paper. The core steps of both algorithms involve the rapid computation of the product of a random matrix and the matrix to be approximated. The algorithm of [14] assumes that the matrix to be approximated (and its transpose) can be applied rapidly to arbitrary vectors, thus enabling the rapid computation of the product of the matrix to be approximated (or its transpose) and any matrix. In contrast, the algorithm of the present paper utilizes a random matrix $\mathcal{R}$ which can be applied rapidly to arbitrary vectors, thus enabling the rapid computation of the product of $\mathcal{R}$ and any matrix.

The matrix $\mathcal{R}$ employed in the present paper consists of several uniformly randomly selected rows of the product of a discrete Fourier transform matrix and a random diagonal matrix. The fast Fourier transform and similar algorithms allow the rapid application of $\mathcal{R}$ to arbitrary vectors (see, for example, [15] for a discussion of the fast Fourier transform algorithm and its applications). The idea of using a random matrix with such structure has been introduced in [1]. The idea of using such a matrix in numerical linear algebra (specifically, for the purpose of computing a solution in the least-squares sense to an overdetermined system of linear-algebraic equations) has been introduced in [16], utilizing both [1] and [7].

It should be observed that there is nothing magical about our choice of the matrix $\mathcal{R}$. In our experience, several other constructions work just as well; for example, the Fourier transform utilized in the present paper can be replaced with the Walsh–Hadamard transform (see [1] or [21]). We are investigating several possible alternatives (see [2]). We gave preliminary versions of the present paper in [20] and [21]. For simplicity, we discuss here only complex matrices; our preliminary report [21] discusses an early version of the algorithm tailored for real matrices.

The present paper has the following structure: Section 2 sets the notation. Section 3 collects together various known facts which later sections utilize. Section 4 provides the principal lemmas which Section 5 uses to construct algorithms. Section 5 describes the algorithm of the present paper, providing details about its accuracy and computational costs. Section 6 illustrates the performance of the algorithm via several numerical examples. Section 7 draws several conclusions and proposes directions for further work.

2. Notation

In this section, we set notational conventions employed throughout the present paper.

We denote an identity matrix by $\mathbf{1}$, and a matrix whose entries are all zeros by $\mathbf{0}$. For any matrix A, we define the norm $\|A\|$ of A to be the spectral (l^2-operator) norm of A, that is, $\|A\|$ is the greatest singular value of A. For any matrix A, we define A^* to be the adjoint of A. For any complex number z, we define $\bar{z}$ to be the conjugate of z. We use $i = \sqrt{-1}$ and $e = \exp(1)$. For any nonnegative integers n and m, we define $l = (n \bmod m)$ to be the integer l such that $n - l$ is a multiple of m and $0 \leqslant l \leqslant m - 1$, that is, $n \bmod m$ is the remainder after integer division of n by m. We use $\mathbf{P}$ to take the probability of an event, and $\mathbf{E}$ to take the expectation of a random variable. We abbreviate "independent, identically distributed" to "i.i.d."

For any positive integer m, we define the unnormalized discrete Fourier transform $F^{(m)}$ to be the complex $m \times m$ matrix with the entry

$$\left(F^{(m)}\right)_{j,k} = e^{-2\pi i(j-1)(k-1)/m} \tag{5}$$

for $j, k = 1, 2, \ldots, m-1, m$; if the size m is clear from the context, then we omit the superscript in $F^{(m)}$, denoting the unnormalized discrete Fourier transform by simply F.

We will frequently utilize the following subsampled randomized Fourier transform. For any positive integers l and m with $l < m$, we define the $l \times m$ SRFT to be the complex $l \times m$ random matrix

$$\mathcal{R} = SFD. \tag{6}$$

In (6), S is the $l \times m$ matrix whose entries are all zeros, aside from a single 1 in column s_j of row j for $j = 1, 2, \ldots, l-1, l$, where $s_1, s_2, \ldots, s_{l-1}, s_l$ are i.i.d. integer random variables, each distributed uniformly over $\{1, 2, \ldots, m-1, m\}$. Moreover, F is the $m \times m$ unnormalized discrete Fourier transform, and D is the diagonal $m \times m$ matrix whose diagonal entries $d_1, d_2, \ldots, d_{m-1}, d_m$ are i.i.d. complex random variables, each distributed uniformly over the unit circle. We call $\mathcal{R}$ an "SRFT" for lack of a better term.

3. Preliminaries

In this section, we summarize various facts from linear algebra, and describe efficient algorithms for computing an arbitrary subset of the outputs of a discrete Fourier transform, as well as for identifying matrices whose spectral norms are larger than desired.

3.1. General facts from linear algebra

In this subsection, we summarize various general facts from linear algebra.

The following lemma states that, for any $m \times n$ matrix A whose rank is k, there exist an $m \times k$ matrix B whose columns constitute a subset of the columns of A, and a $k \times n$ matrix P, such that

1. some subset of the columns of P makes up the $k \times k$ identity matrix,
2. P is not too large, and
3. $BP = A$.

Moreover, the lemma provides an analogous approximation BP to A when the exact rank of A is not k, but the $(k+1)$st singular value of A is nevertheless small. The lemma is a reformulation of Theorem 3.2 in [13] and Theorem 3 in [5].

Lemma 3.1. *Suppose that m and n are positive integers, and A is a complex $m \times n$ matrix.*

Then, for any positive integer k with $k \leqslant m$ and $k \leqslant n$, there exist a complex $k \times n$ matrix P, and a complex $m \times k$ matrix B whose columns constitute a subset of the columns of A, such that

1. *some subset of the columns of P makes up the $k \times k$ identity matrix,*
2. *no entry of P has an absolute value greater than* 1,
3. $\|P\| \leqslant \sqrt{k(n-k)+1}$,
4. *the least* (*that is, the kth greatest*) *singular value of P is at least* 1,
5. *$BP = A$ when $k = m$ or $k = n$, and*
6. $\|BP - A\| \leqslant \sqrt{k(n-k)+1}\sigma_{k+1}$ *when $k < m$ and $k < n$, where σ_{k+1} is the* $(k+1)$*st greatest singular value of A.*

Remark 3.2. Properties 1, 2, 3, and 4 in Lemma 3.1 ensure that the interpolative decomposition BP of A is numerically stable. Also, Property 3 follows directly from Properties 1 and 2, and Property 4 follows directly from Property 1.

Observation 3.3. Existing algorithms for computing the matrices B and P in Lemma 3.1 are computationally expensive. We use an algorithm to produce B and P which satisfy somewhat weaker conditions than those in Lemma 3.1. We compute B and P such that

1. some subset of the columns of P makes up the $k \times k$ identity matrix,
2. no entry of P has an absolute value greater than 2,
3. $\|P\| \leqslant \sqrt{4k(n-k)+1}$,
4. the least (that is, the kth greatest) singular value of P is at least 1,
5. $BP = A$ when $k = m$ or $k = n$, and
6. $\|BP - A\| \leqslant \sqrt{4k(n-k)+1}\sigma_{k+1}$ when $k < m$ and $k < n$, where σ_{k+1} is the $(k+1)$st greatest singular value of A.

For any positive real number ε, the algorithm can identify the least k such that $\|BP - A\| \approx \varepsilon$. Furthermore, there exists a real number C such that the algorithm computes both B and P using at most $Ckmn\log(n)$ floating-point operations and Cmn floating-point words of memory. The algorithm is based upon the Cramer rule and the ability to obtain the minimal-norm (or at least roughly minimal-norm) solutions to linear-algebraic systems of equations (see [5,10,13]).

The following lemma provides an approximation QZ to an $n \times l$ matrix Y via an $n \times k$ matrix Q whose columns are orthonormal, and a $k \times l$ matrix Z.

Lemma 3.4. *Suppose that k, l, and n are positive integers with $k < l \leqslant n$, and Y is a complex $n \times l$ matrix. Then, there exist a complex $n \times k$ matrix Q whose columns are orthonormal, and a complex $k \times l$ matrix Z, such that*

$$\|QZ - Y\| \leqslant \eta_{k+1}, \tag{7}$$

where η_{k+1} is the $(k+1)$st greatest singular value of Y.

Proof. We start by forming an SVD of Y,

$$Y = U\Sigma V^*, \tag{8}$$

where U is a complex $n \times l$ matrix whose columns are orthonormal, V is a complex $l \times l$ matrix whose columns are orthonormal, and Σ is a real diagonal $l \times l$ matrix whose entries are all nonnegative, such that

$$\Sigma_{j,j} = \eta_j \tag{9}$$

for $j = 1, 2, \ldots, l-1, l$, where $\Sigma_{j,j}$ is the entry in row j and column j of Σ, and η_j is the jth greatest singular value of Y. We define Q to be the leftmost $n \times k$ block of U, and P to be the rightmost $n \times (l-k)$ block of U, so that

$$U = (Q|P). \tag{10}$$

We define Z to be the uppermost $k \times l$ block of ΣV^*, and X to be the lowermost $(l-k) \times l$ block of ΣV^*, so that

$$\Sigma V^* = \left(\frac{Z}{X}\right). \tag{11}$$

Combining (8)–(11) and the fact that the columns of U are orthonormal, as are the columns of V, yields (7). □

Observation 3.5. In order to compute the matrices Q and Z in (7) from the matrix Y, we can construct (8), and then form Q and Z according to (10) and (11). (See, for example, Chapter 8 in [8] for details concerning the computation of the SVD.)

The following technical lemma will be needed in Section 4; Lemma 6 of [14] provides a proof.

Lemma 3.6. *Suppose that k and l are positive integers with $k \leqslant l$. Suppose further that G is a complex $l \times k$ matrix such that G^*G is invertible.*

Then,

$$\left\|(G^*G)^{-1}G^*\right\| = \frac{1}{\sigma_k}, \tag{12}$$

where σ_k is the least (that is, the kth greatest) singular value of G.

3.2. More specialized facts from linear algebra

In this subsection, we summarize various facts from linear algebra that are useful specifically for the randomized approximation of matrices.

The following lemma states that the product BP of matrices B and P is a good approximation to a matrix A, provided that there exists a matrix $\mathcal{R}$ such that

1. the columns of B constitute a subset of the columns of A,
2. $\|P\|$ is not too large,
3. $\mathcal{R}BP$ is a good approximation to $\mathcal{R}A$, and
4. there exists a matrix T such that $\|T\|$ is not too large, and $T\mathcal{R}A$ is a good approximation to A.

Lemma 3.7. *Suppose that k, l, m, and n are positive integers with $k \leqslant n$. Suppose further that A is a complex $m \times n$ matrix, B is a complex $m \times k$ matrix whose columns constitute a subset of the columns of A, P is a complex $k \times n$ matrix, T is a complex $m \times l$ matrix, and $\mathcal{R}$ is a complex $l \times m$ matrix.*

Then,

$$\|BP - A\| \leqslant \|T\mathcal{R}A - A\|\bigl(\|P\| + 1\bigr) + \|T\|\|\mathcal{R}BP - \mathcal{R}A\|. \tag{13}$$

Proof. We observe that

$$\|BP - A\| \leqslant \|BP - T\mathcal{R}BP\| + \|T\mathcal{R}BP - T\mathcal{R}A\| + \|T\mathcal{R}A - A\|, \tag{14}$$

$$\|BP - T\mathcal{R}BP\| \leqslant \|B - T\mathcal{R}B\|\|P\|, \tag{15}$$

and

$$\|T\mathcal{R}BP - T\mathcal{R}A\| \leqslant \|T\|\|\mathcal{R}BP - \mathcal{R}A\|. \tag{16}$$

Since the columns of B constitute a subset of the columns of A, it follows that the columns of $B - T\mathcal{R}B$ constitute a subset of the columns of $A - T\mathcal{R}A$, and therefore,

$$\|B - T\mathcal{R}B\| \leqslant \|A - T\mathcal{R}A\|. \tag{17}$$

Combining (14)–(17) yields (13). □

Remark 3.8. Since the columns of B constitute a subset of the columns of A in Lemma 3.7, it follows that the columns of $\mathcal{R}B$ constitute a subset of the columns of $\mathcal{R}A$. Conversely, whenever a matrix Z is formed by gathering distinct columns of $Y = \mathcal{R}A$ together into Z, then clearly $Z = \mathcal{R}B$ for some matrix B whose columns constitute a subset of the columns of A.

The following lemma states that the product AQQ^* of matrices A, Q, and Q^* is a good approximation to a matrix A, provided that there exist matrices $\mathcal{R}$ and Z such that

1. the columns of Q are orthonormal,
2. QZ is a good approximation to $(\mathcal{R}A)^*$, and
3. there exists a matrix T such that $\|T\|$ is not too large, and $T\mathcal{R}A$ is a good approximation to A.

Lemma 17 of [14] provides a proof for the following lemma.

Lemma 3.9. *Suppose that k, l, m, and n are positive integers with $k \leqslant n$. Suppose further that A is a complex $m \times n$ matrix, Q is a complex $n \times k$ matrix whose columns are orthonormal, Z is a complex $k \times l$ matrix, T is a complex $m \times l$ matrix, and $\mathcal{R}$ is a complex $l \times m$ matrix.*

Then,

$$\|AQQ^* - A\| \leqslant 2\|T\mathcal{R}A - A\| + 2\|T\|\bigl\|QZ - (\mathcal{R}A)^*\bigr\|. \tag{18}$$

The following lemma provides an efficient means of computing an SVD of a complex $m \times n$ matrix A, given a complex $m \times k$ matrix B and a complex $k \times n$ matrix P such that $A = BP$ and k is much less than both m and n. If, in addition, $\|B\| \leqslant \|A\|$ and P is well conditioned, then the scheme described by the lemma is numerically stable. Clearly, if B and P arise from an interpolative decomposition, then indeed $\|B\| \leqslant \|A\|$ and P is well conditioned, and so the scheme described by the lemma is numerically stable.

Lemma 3.10. *Suppose that k, m, and n are positive integers with $k \leqslant m$ and $k \leqslant n$. Suppose further that A is a complex $m \times n$ matrix, B is a complex $m \times k$ matrix, and P is a complex $k \times n$ matrix, such that*

$$A = BP. \tag{19}$$

Suppose in addition that L is a complex $k \times k$ matrix, and Q is a complex $n \times k$ matrix whose columns are orthonormal, such that

$$P = LQ^*. \tag{20}$$

Suppose finally that C is a complex $m \times k$ matrix, U is a complex $m \times k$ matrix whose columns are orthonormal, Σ is a real $k \times k$ matrix, and W is a complex $k \times k$ matrix whose columns are orthonormal, such that

$$C = BL \tag{21}$$

and

$$C = U\Sigma W^*. \tag{22}$$

Then,

$$A = U\Sigma V^*, \tag{23}$$

where V is the complex $n \times k$ matrix given by the formula

$$V = QW. \tag{24}$$

Moreover, the columns of V are orthonormal (as are the columns of U), and

$$\|L\| = \|P\|. \tag{25}$$

Proof. Combining (19)–(22) and (24) yields (23). Combining (24) and the facts that W is unitary and that the columns of Q are orthonormal yields that the columns of V are orthonormal. Combining (20) and the fact that the columns of Q are orthonormal yields (25). □

Remark 3.11. The matrices L and Q in (20) can be computed from P as follows. Using the algorithms described, for example, in Chapter 5 of [8], we construct an upper triangular complex $k \times k$ matrix R, and a complex $n \times k$ matrix Q whose columns are orthonormal, such that

$$P^* = QR. \tag{26}$$

We thus obtain Q. We then define L to be the adjoint of R, that is,

$$L = R^*. \tag{27}$$

3.3. An accelerated fast Fourier transform

In this subsection, we describe an efficient algorithm for computing an arbitrary subset of the outputs of a discrete Fourier transform, based on the fast Fourier transform (see, for example, [15] for a discussion of the fast Fourier transform algorithm and its applications). The algorithm requires $\mathcal{O}(n\log(l))$ floating-point operations in order to compute l samples of the discrete Fourier transform of a vector of length n. [18] discusses an extremely similar algorithm.

The following lemma is easily verified by identifying the summation indices k, k_1, and k_2 via the equation $k = m(k_1 - 1) + k_2$.

Lemma 3.12. *Suppose that l, m, and n are positive integers with $n = l \cdot m$, and v is a complex $n \times 1$ column vector. Then,*

$$\sum_{k=1}^{n} e^{-2\pi i(j-1)(k-1)/n} v_k = \sum_{k_2=1}^{m} e^{-2\pi i(j_1-1)(k_2-1)/m} \cdot e^{-2\pi i(j_2-1)(k_2-1)/n} \cdot \sum_{k_1=1}^{l} e^{-2\pi i(j_2-1)(k_1-1)/l} v_{m(k_1-1)+k_2} \tag{28}$$

for $j_1 = 1, 2, \ldots, m-1, m$ and $j_2 = 1, 2, \ldots, l-1, l$, where $j = l(j_1 - 1) + j_2$.

Suppose that l, m, and n are positive integers with $n = l \cdot m$, and v and z are complex $n \times 1$ column vectors, such that $z = F^{(n)}v$, where $F^{(n)}$ is the $n \times n$ unnormalized discrete Fourier transform. Then, it follows from (28) that the following procedure computes z from v:

1. Viewing the vector v as an $l \times m$ matrix V stored in row-major order, form the product W of the $l \times l$ unnormalized discrete Fourier transform $F^{(l)}$ and V, so that
$$W = F^{(l)} V. \tag{29}$$
2. Multiply the entry in row j and column k of W by $e^{-2\pi i(j-1)(k-1)/n}$ for $j = 1, 2, \ldots, l-1, l$ and $k = 1, 2, \ldots, m-1, m$, in order to obtain the $l \times m$ matrix X.
3. Transpose X to obtain an $m \times l$ matrix Y, so that
$$Y = X^{\mathrm{T}}. \tag{30}$$
4. Form the product Z of the $m \times m$ unnormalized discrete Fourier transform $F^{(m)}$ and Y, so that
$$Z = F^{(m)} Y. \tag{31}$$
View the $m \times l$ matrix Z as a vector z stored in row-major order.

If we only need to compute l entries of $z = F^{(n)} v$, then we can use Steps 1–3 above in their entirety to obtain Y, and then compute the desired entries of z directly from the entries of Y. Step 1 costs $\mathcal{O}(m \cdot l \log(l))$ using the fast Fourier transform, Step 2 costs $\mathcal{O}(m \cdot l)$, and Step 3 costs $\mathcal{O}(m \cdot l)$. It follows from (31) that each entry of z is a linear combination of m entries of Y. Therefore, computing the l desired entries of z directly from the entries of Y costs $\mathcal{O}(l \cdot m)$.

Summing up these costs and using the fact that $l \cdot m = n$, we find that computing any specified l entries of $z = F^{(n)} v$ from the entries of v costs
$$C_{l \text{ of } n} = \mathcal{O}\big(n \log(l)\big) \tag{32}$$
floating-point operations.

3.4. A randomized scheme for estimating the spectral norm of a matrix

In this subsection, we formalize the intuitively obvious statement that the product Ax of a matrix A and a random vector x has a small Euclidean norm whenever $\|A\|$ is small, and that $\|Ax\|$ is only rarely not small whenever $\|A\|$ is not small. By applying A to a short sequence of independent vectors $x^{(1)}, x^{(2)}, \ldots, x^{(k-1)}, x^{(k)}$ and looking at the results, we can estimate $\|A\|$ with very high probability and acceptable accuracy.

Needless to say, other estimates of this type have been constructed previously; those in [3] are some of the best-known ones. The estimates of [3] are different from ours in several respects, most notably in that we work in the Euclidean norm, while the authors of [3] work in the max norm; in addition, the entries of our vectors $x^{(1)}, x^{(2)}, \ldots, x^{(k-1)}, x^{(k)}$ are Gaussian random variables, while in [3] the entries are chosen uniformly at random from $\{-1, 1\}$.

Theorem 3.15 below is the principal purpose of this subsection; we start with two technical lemmas. The following lemma provides an expression for the probability that a certain trial will succeed several times, in terms of the probability that the trial will succeed once.

Lemma 3.13. *Suppose that μ is a positive real number, k, m, and n are positive integers, A is a complex $m \times n$ matrix, and x and $x^{(1)}, x^{(2)}, \ldots, x^{(k-1)}, x^{(k)}$ are $n \times 1$ i.i.d. random vectors with i.i.d. entries, each distributed as a complex Gaussian random variable of zero mean and unit variance.*

Then,
$$\mathbf{P}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\| \textit{ for all } j = 1, 2, \ldots, k-1, k\right\} = \left(\mathbf{P}\left\{\frac{\|Ax\|}{\|x\|} < \mu\|A\|\right\}\right)^k. \tag{33}$$

Proof. Clearly,
$$\mathbf{P}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\| \text{ for all } j = 1, 2, \ldots, k-1, k\right\} = \mathbf{P}\left(\bigcap_{j=1}^{k}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\|\right\}\right). \tag{34}$$

It follows from the independence of $x^{(1)}, x^{(2)}, \dots, x^{(k-1)}, x^{(k)}$ that

$$\mathbf{P}\Bigg(\bigcap_{j=1}^{k}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\|\right\}\Bigg) = \prod_{j=1}^{k}\mathbf{P}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\|\right\}. \tag{35}$$

It follows from the fact that $x^{(1)}, x^{(2)}, \dots, x^{(k-1)}, x^{(k)}$ are all distributed the same as x that

$$\mathbf{P}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\|\right\} = \mathbf{P}\left\{\frac{\|Ax\|}{\|x\|} < \mu\|A\|\right\} \tag{36}$$

for $j = 1, 2, \dots, k-1, k$. Combining (34)–(36) yields (33). □

Given a matrix A and a random vector x, the following lemma estimates the probability that $\|Ax\|$ is small compared to $\|A\| \cdot \|x\|$. Theorem 1 in [6] provides another formulation of this lemma.

Lemma 3.14. *Suppose that μ is a real number with $0 < \mu < 1$, m and n are positive integers, A is a complex $m \times n$ matrix, and x is an $n \times 1$ random vector with i.i.d. entries, each distributed as a complex Gaussian random variable of zero mean and unit variance.*

Then,

$$\mathbf{P}\left\{\frac{\|Ax\|}{\|x\|} < \mu\|A\|\right\} \leqslant 0.8\mu\sqrt{n}. \tag{37}$$

The following theorem provides an efficient means for testing whether the spectral norm of a matrix exceeds a user-specified threshold.

Theorem 3.15. *Suppose that μ is a real number with $0 < \mu < 1$, k, m, and n are positive integers, A is a complex $m \times n$ matrix, and $x^{(1)}, x^{(2)}, \dots, x^{(k-1)}, x^{(k)}$ are $n \times 1$ i.i.d. random vectors with i.i.d. entries, each distributed as a complex Gaussian random variable of zero mean and unit variance.*

Then,

$$\mathbf{P}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \mu\|A\| \textit{ for all } j = 1, 2, \dots, k-1, k\right\} \leqslant (0.8\mu\sqrt{n})^k. \tag{38}$$

Proof. Combining (33) and (37) yields (38). □

We now consider an application of Theorem 3.15.

On the one hand, suppose that ε is a positive real number, m and n are positive integers, and A is a complex $m \times n$ matrix, such that

$$\|A\| \geqslant 80\sqrt{n}\varepsilon. \tag{39}$$

To ascertain computationally that A has a spectral norm greater than ε, we apply A to half a dozen $n \times 1$ i.i.d. random vectors $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, $x^{(4)}$, $x^{(5)}$, $x^{(6)}$ with i.i.d. entries, each distributed as a complex Gaussian random variable of zero mean and unit variance. We then check whether at least one of the numbers

$$\frac{\|Ax^{(1)}\|}{\|x^{(1)}\|}, \quad \frac{\|Ax^{(2)}\|}{\|x^{(2)}\|}, \quad \frac{\|Ax^{(3)}\|}{\|x^{(3)}\|}, \quad \frac{\|Ax^{(4)}\|}{\|x^{(4)}\|}, \quad \frac{\|Ax^{(5)}\|}{\|x^{(5)}\|}, \quad \frac{\|Ax^{(6)}\|}{\|x^{(6)}\|} \tag{40}$$

is at least ε. Combining (39) and (38) with $\mu = \frac{1}{80\sqrt{n}}$ yields that

$$\mathbf{P}\left\{\frac{\|Ax^{(j)}\|}{\|x^{(j)}\|} < \varepsilon \text{ for all } j = 1, 2, 3, 4, 5, 6\right\} \leqslant 10^{-12}. \tag{41}$$

Thus, we are unlikely to find that all of the numbers (40) are less than ε. Obviously, if the spectral norm of A is even greater than $80\sqrt{n}\varepsilon$, then we are even less likely to find that all of the numbers (40) are less than ε.

On the other hand, suppose that ε is a positive real number, m and n are positive integers, and A is a complex $m \times n$ matrix, such that

$$\|A\| < \varepsilon. \tag{42}$$

Suppose we apply A to half a dozen $n \times 1$ i.i.d. random vectors $x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}$ with i.i.d. entries, each distributed as a complex Gaussian random variable of zero mean and unit variance. Then, (42) guarantees that all of the numbers (40) will be less than ε.

Hence, by applying matrices to a few random vectors, we can with very high probability filter out those matrices whose spectral norms are significantly greater than a user-specified precision ε, while always passing all of the matrices whose spectral norms are less than ε. Thus, we can efficiently and reliably identify matrices whose spectral norms are larger than desired, even when we cannot afford to form the individual entries of the matrices.

Remark 3.16. It is also possible to estimate the spectral norm of a matrix using the power method (or the Lanczos method when tighter bounds are desired) with a random starting vector. The analysis in [11] guarantees a better bound for both the power method and the Lanczos method as compared with the scheme of the present subsection when running-times are proportional to operation counts, and storage is not an issue. However, the power and Lanczos methods require successive applications of the matrix being tested (as well as its transpose) to a sequence of vectors generated on-the-fly, whereas the scheme of the present subsection requires only the application of the matrix being tested to a collection of independently generated vectors. Thus, in some circumstances the scheme of the present subsection parallelizes better than the power and Lanczos methods.

4. Mathematical apparatus

In this section, we describe the principal mathematical tools used in Section 5.

4.1. Several technical lemmas

In this subsection, we prove several lemmas needed for the proof of Lemma 4.4 in Section 4.2.

The following lemma evaluates the mean and variance of a certain random variable similar to a random walk.

Lemma 4.1. *Suppose that l and m are positive integers with $l \leqslant m$, $s_1, s_2, \ldots, s_{l-1}, s_l$ are i.i.d. integer random variables, each distributed uniformly over $\{1, 2, \ldots, m-1, m\}$, F is the $m \times m$ unnormalized discrete Fourier transform, G_{sqb} is the complex number*

$$G_{sqb} = \overline{F_{sq}} F_{sb} \tag{43}$$

for $s, q, b = 1, 2, \ldots, m-1, m$, and H_{qb} is the complex number

$$H_{qb} = \sum_{r=1}^{l} G_{s_r qb} \tag{44}$$

for $q, b = 1, 2, \ldots, m-1, m$.

Then,

$$H_{qq} = l \tag{45}$$

for $q = 1, 2, \ldots, m-1, m$,

$$\mathbf{E} H_{qb} = 0 \tag{46}$$

when $q \neq b$, and

$$\mathbf{E}|H_{qb}|^2 = l \tag{47}$$

when $q \neq b$.

Proof. First, we prove (45). It follows from (43) that

$$G_{sqq} = 1 \tag{48}$$

for $s, q = 1, 2, \ldots, m-1, m$. Combining (44) and (48) yields (45).

Next, we prove (46). It follows from (44) that

$$\mathbf{E}H_{qb} = \sum_{r=1}^{l} \mathbf{E}G_{s_r qb} \tag{49}$$

for $q, b = 1, 2, \ldots, m-1, m$. For any $r = 1, 2, \ldots, l-1, l$, it follows from the fact that s_r is distributed uniformly over $\{1, 2, \ldots, m-1, m\}$ that

$$\mathbf{E}G_{s_r qb} = \frac{1}{m} \sum_{s=1}^{m} G_{sqb} \tag{50}$$

for $q, b = 1, 2, \ldots, m-1, m$. Combining (43) and the fact that distinct columns of F are orthogonal yields that

$$\sum_{s=1}^{m} G_{sqb} = 0 \tag{51}$$

when $q \neq b$. Combining (49)–(51) yields (46).

Finally, we prove (47). It follows from (44) that

$$|H_{qb}|^2 = \sum_{r=1}^{l} |G_{s_r qb}|^2 + \sum_{r \neq t} \overline{G_{s_r qb}} G_{s_t qb} \tag{52}$$

for $q, b = 1, 2, \ldots, m-1, m$. It follows from (52) that

$$\mathbf{E}|H_{qb}|^2 = \sum_{r=1}^{l} \mathbf{E}|G_{s_r qb}|^2 + \sum_{r \neq t} \mathbf{E}\overline{G_{s_r qb}} G_{s_t qb} \tag{53}$$

for $q, b = 1, 2, \ldots, m-1, m$.

It follows from (43) that

$$|G_{sqb}| = |F_{sq}||F_{sb}| \tag{54}$$

for $s, q, b = 1, 2, \ldots, m-1, m$. However,

$$|F_{sq}| = |F_{sb}| = 1 \tag{55}$$

for $s, q, b = 1, 2, \ldots, m-1, m$. Combining (54) and (55) yields that

$$\mathbf{E}|G_{s_r qb}|^2 = 1 \tag{56}$$

for $r = 1, 2, \ldots, l-1, l$ and $q, b = 1, 2, \ldots, m-1, m$.

It follows from the independence of $s_1, s_2, \ldots, s_{l-1}, s_l$ that

$$\mathbf{E}\overline{G_{s_r qb}} G_{s_t qb} = (\overline{\mathbf{E}G_{s_r qb}})(\mathbf{E}G_{s_t qb}) \tag{57}$$

for $q, b = 1, 2, \ldots, m-1, m$, when $r \neq t$. Combining (57), (50), and (51) yields that

$$\mathbf{E}\overline{G_{s_r qb}} G_{s_t qb} = 0 \tag{58}$$

for $q, b = 1, 2, \ldots, m-1, m$, when $r \neq t$.

Combining (53), (56), and (58) yields (47). □

We will need the following technical lemma in order to prove Lemma 4.4 below.

Lemma 4.2. *Suppose that k, l, and m are positive integers, such that $k \leqslant l < m$. Suppose further that H_{qb} is the complex number defined in* (44) *and* (43), *for $q, b = 1, 2, \ldots, m-1, m$. Suppose finally that $\mathcal{R}$ is the $l \times m$ SRFT defined in Section* 2, *U is a complex $m \times k$ matrix whose columns are orthonormal, C is the complex $k \times k$ matrix defined via the formula*

$$C = (\mathcal{R}U)^*(\mathcal{R}U), \tag{59}$$

and E is the complex $k \times k$ matrix with the entry

$$E_{pa} = \sum_{q=1}^{m} \overline{d_q}\,\overline{U_{qp}} \sum_{b \neq q} d_b U_{ba} H_{qb} \tag{60}$$

for $p, a = 1, 2, \ldots, k-1, k$, where $d_1, d_2, \ldots, d_{m-1}, d_m$ are the i.i.d. complex random variables, each distributed uniformly over the unit circle, used in the construction of D in (6) *for the SRFT $\mathcal{R}$.*

Then,

$$C = l \cdot \mathbf{1} + E. \tag{61}$$

Proof. For any integers a and p with $1 \leqslant a \leqslant k$ and $1 \leqslant p \leqslant k$, combining (59) and (6) yields that

$$C_{pa} = \sum_{r=1}^{l} \sum_{q=1}^{m} \overline{F_{s_r q}}\,\overline{d_q}\,\overline{U_{qp}} \sum_{b=1}^{m} F_{s_r b} d_b U_{ba}. \tag{62}$$

Combining (62), (43), and (44) yields that

$$C_{pa} = \sum_{q=1}^{m} |d_q|^2 \overline{U_{qp}} U_{qa} H_{qq} + \sum_{q=1}^{m} \overline{d_q}\,\overline{U_{qp}} \sum_{b \neq q} d_b U_{ba} H_{qb}. \tag{63}$$

Combining (63), (45), the fact that $|d_q| = 1$, and the fact that the columns of U are orthonormal yields that

$$C_{pp} = l + \sum_{q=1}^{m} \overline{d_q}\,\overline{U_{qp}} \sum_{b \neq q} d_b U_{bp} H_{qb}, \tag{64}$$

and

$$C_{pa} = \sum_{q=1}^{m} \overline{d_q}\,\overline{U_{qp}} \sum_{b \neq q} d_b U_{ba} H_{qb} \tag{65}$$

when $p \neq a$.

Combining (64), (65), and (60) yields (61). □

The following lemma states that the spectral norm of the matrix E defined in (60) is reasonably small with high probability.

Lemma 4.3. *Suppose that α and β are real numbers greater than* 1, *and k, l, and m are positive integers, such that*

$$m > l \geqslant \frac{\alpha^2 \beta}{(\alpha - 1)^2} k^2. \tag{66}$$

Suppose further that $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2, *U is a complex $m \times k$ matrix whose columns are orthonormal, and E is the complex $k \times k$ matrix defined in* (60), *with H_{qb} being the complex number defined in* (44) *and* (43), *and with $d_1, d_2, \ldots, d_{m-1}, d_m$ being the i.i.d. complex random variables, each distributed uniformly over the unit circle, used in the construction of D in* (6) *for the SRFT $\mathcal{R}$.*

Then,

$$\|E\| \leqslant l\left(1 - \frac{1}{\alpha}\right) \tag{67}$$

with probability at least $1 - \frac{1}{\beta}$.

Proof. We first derive an upper bound on $\mathbf{E}|E_{pa}|^2$, for $p, a = 1, 2, \dots, k-1, k$, and then use this bound to prove (67). It follows from (60) that

$$\mathbf{E}|E_{pa}|^2 = \mathbf{E}\left(\sum_{q=1}^{m} d_q U_{qp} \sum_{b\neq q} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\right)\left(\sum_{r=1}^{m} \overline{d_r}\,\overline{U_{rp}} \sum_{c\neq r} d_c U_{ca} H_{rc}\right). \tag{68}$$

Performing the summation over q and r separately for the cases when $q = r$ and when $q \neq r$, and using the fact that $|d_q| = 1$, we obtain that

$$\begin{aligned}&\mathbf{E}\sum_{q,r=1}^{m}\left(d_q U_{qp} \sum_{b\neq q} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\right)\left(\overline{d_r}\,\overline{U_{rp}} \sum_{c\neq r} d_c U_{ca} H_{rc}\right)\\ &\quad= \mathbf{E}\sum_{q=1}^{m} |U_{qp}|^2 \left|\sum_{b\neq q} d_b U_{ba} H_{qb}\right|^2 + \mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp} \overline{U_{rp}} \sum_{b\neq q} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}} \sum_{c\neq r} d_c U_{ca} H_{rc}.\end{aligned} \tag{69}$$

To bound the first term in the right-hand side of (69), we observe that

$$\mathbf{E}\sum_{q=1}^{m} |U_{qp}|^2 \left|\sum_{b\neq q} d_b U_{ba} H_{qb}\right|^2 = \sum_{q=1}^{m} |U_{qp}|^2 \mathbf{E}\left|\sum_{b\neq q} d_b U_{ba} H_{qb}\right|^2. \tag{70}$$

But,

$$\mathbf{E}\left|\sum_{b\neq q} d_b U_{ba} H_{qb}\right|^2 = \mathbf{E}\sum_{b\neq q} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}} \sum_{c\neq q} d_c U_{ca} H_{qc}. \tag{71}$$

Moreover,

$$\mathbf{E}\sum_{b\neq q} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}} \sum_{c\neq q} d_c U_{ca} H_{qc} = \sum_{b,c\neq q} \overline{U_{ba}} U_{ca} \mathbf{E}\overline{d_b} d_c \overline{H_{qb}} H_{qc}. \tag{72}$$

Performing the summation over b and c separately for the cases when $b = c$ and when $b \neq c$, and using the fact that $|d_b| = 1$, we obtain that

$$\sum_{b,c\neq q} \overline{U_{ba}} U_{ca} \mathbf{E}\overline{d_b} d_c \overline{H_{qb}} H_{qc} = \sum_{b\neq q} |U_{ba}|^2 \mathbf{E}|H_{qb}|^2 + \sum_{b,c\neq q \text{ and } b\neq c} \overline{U_{ba}} U_{ca} \mathbf{E}\overline{d_b} d_c \overline{H_{qb}} H_{qc}. \tag{73}$$

It follows from (47) that

$$\sum_{b\neq q} |U_{ba}|^2 \mathbf{E}|H_{qb}|^2 = l \sum_{b\neq q} |U_{ba}|^2. \tag{74}$$

It follows from the fact that the columns of U are normalized that

$$\sum_{b\neq q} |U_{ba}|^2 \leqslant \sum_{b=1}^{m} |U_{ba}|^2 = 1. \tag{75}$$

Combining (74) and (75) yields that

$$\sum_{b\neq q} |U_{ba}|^2 \mathbf{E}|H_{qb}|^2 \leqslant l. \tag{76}$$

It follows from the independence of the random variables involved that

$$\sum_{b,c\neq q \text{ and } b\neq c} \overline{U_{ba}} U_{ca} \mathbf{E}\overline{d_b} d_c \overline{H_{qb}} H_{qc} = \sum_{b,c\neq q \text{ and } b\neq c} \overline{U_{ba}} U_{ca} (\overline{\mathbf{E}d_b})(\mathbf{E}d_c)(\mathbf{E}\overline{H_{qb}} H_{qc}). \tag{77}$$

Combining (77) and the fact that $\mathbf{E}d_b = 0$ (or $\mathbf{E}d_c = 0$) yields that

$$\sum_{b,c\neq q \text{ and } b\neq c} \overline{U_{ba}} U_{ca} \mathbf{E}\overline{d_b} d_c \overline{H_{qb}} H_{qc} = 0. \tag{78}$$

Combining (70)–(73), (76), and (78) yields that

$$\mathbf{E}\sum_{q=1}^{m}|U_{qp}|^2\left|\sum_{b\neq q} d_b U_{ba} H_{qb}\right|^2 \leqslant l\sum_{q=1}^{m}|U_{qp}|^2. \tag{79}$$

Combining (79) and the fact that the columns of U are normalized yields that

$$\mathbf{E}\sum_{q=1}^{m}|U_{qp}|^2\left|\sum_{b\neq q} d_b U_{ba} H_{qb}\right|^2 \leqslant l. \tag{80}$$

To bound the second term in the right-hand side of (69), we observe that

$$\begin{aligned}
&\mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\sum_{b\neq q}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\sum_{c\neq r} d_c U_{ca} H_{rc}\\
&\quad= \mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\left(\overline{d_r}\,\overline{U_{ra}}\,\overline{H_{qr}} + \sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\right)\left(d_q U_{qa} H_{rq} + \sum_{c\neq q,r} d_c U_{ca} H_{rc}\right).
\end{aligned} \tag{81}$$

Expanding the product, we obtain that

$$\begin{aligned}
&\mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\left(\overline{d_r}\,\overline{U_{ra}}\,\overline{H_{qr}} + \sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\right)\left(d_q U_{qa} H_{rq} + \sum_{c\neq q,r} d_c U_{ca} H_{rc}\right)\\
&\quad= \mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\,\overline{d_r} d_q \overline{U_{ra}} U_{qa}\overline{H_{qr}} H_{rq}\\
&\qquad+ \mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\sum_{c\neq q,r} d_c U_{ca} H_{rc}\\
&\qquad+ \mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\,\overline{d_r}\,\overline{U_{ra}}\,\overline{H_{qr}}\sum_{c\neq q,r} d_c U_{ca} H_{rc}\\
&\qquad+ \mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}} d_q U_{qa} H_{rq}\sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}.
\end{aligned} \tag{82}$$

To evaluate the first term in the right-hand side of (82), we observe that

$$\mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\,\overline{d_r} d_q \overline{U_{ra}} U_{qa}\overline{H_{qr}} H_{rq} = \sum_{q\neq r} U_{qp}\overline{U_{rp}}\,\overline{U_{ra}} U_{qa}\mathbf{E}(d_q)^2\overline{(d_r)^2}\,\overline{H_{qr}} H_{rq}. \tag{83}$$

It follows from the independence of the random variables involved that

$$\mathbf{E}(d_q)^2\overline{(d_r)^2}\,\overline{H_{qr}} H_{rq} = \big(\mathbf{E}(d_q)^2\big)\big(\overline{\mathbf{E}(d_r)^2}\big)(\mathbf{E}\overline{H_{qr}} H_{rq}) \tag{84}$$

when $q\neq r$. Combining (83), (84), and the fact that $\mathbf{E}(d_q)^2 = 0$ (or $\mathbf{E}(d_r)^2 = 0$) yields that

$$\mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\,\overline{d_r} d_q \overline{U_{ra}} U_{qa}\overline{H_{qr}} H_{rq} = 0. \tag{85}$$

To evaluate the second term in the right-hand side of (82), we note that the independence of the random variables involved implies that

$$\begin{aligned}
&\mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\sum_{c\neq q,r} d_c U_{ca} H_{rc}\\
&\quad= \sum_{q\neq r} U_{qp}\overline{U_{rp}}(\mathbf{E}d_q)(\overline{\mathbf{E}d_r})\left(\mathbf{E}\sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\sum_{c\neq q,r} d_c U_{ca} H_{rc}\right).
\end{aligned} \tag{86}$$

Combining (86) and the fact that $\mathbf{E}d_q = 0$ (or $\mathbf{E}d_r = 0$) yields that

$$\mathbf{E}\sum_{q\neq r} d_q \overline{d_r} U_{qp}\overline{U_{rp}}\sum_{b\neq q,r}\overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\sum_{c\neq q,r} d_c U_{ca} H_{rc} = 0. \tag{87}$$

To evaluate the third term in the right-hand side of (82), we observe that

$$\mathbf{E}\sum_{q\neq r} d_q\overline{d_r}U_{qp}\overline{U_{rp}}\overline{d_r}\,\overline{U_{ra}}\,\overline{H_{qr}}\sum_{c\neq q,r} d_cU_{ca}H_{rc} = \sum_{q\neq r} U_{qp}\overline{U_{rp}}\,\overline{U_{ra}}\mathbf{E}d_q\overline{(d_r)^2}\,\overline{H_{qr}}\sum_{c\neq q,r} d_cU_{ca}H_{rc}. \tag{88}$$

It follows from the independence of the random variables involved that

$$\mathbf{E}d_q\overline{(d_r)^2}\,\overline{H_{qr}}\sum_{c\neq q,r} d_cU_{ca}H_{rc} = (\mathbf{E}d_q)\big(\overline{\mathbf{E}(d_r)^2}\big)\left(\mathbf{E}\overline{H_{qr}}\sum_{c\neq q,r} d_cU_{ca}H_{rc}\right) \tag{89}$$

when $q \neq r$. It follows from the fact that $\mathbf{E}d_q = 0$ (or $\mathbf{E}(d_r)^2 = 0$) that

$$(\mathbf{E}d_q)\big(\overline{\mathbf{E}(d_r)^2}\big)\left(\mathbf{E}\overline{H_{qr}}\sum_{c\neq q,r} d_cU_{ca}H_{rc}\right) = 0. \tag{90}$$

Combining (88)–(90) yields that

$$\mathbf{E}\sum_{q\neq r} d_q\overline{d_r}U_{qp}\overline{U_{rp}}\overline{d_r}\,\overline{U_{ra}}\,\overline{H_{qr}}\sum_{c\neq q,r} d_cU_{ca}H_{rc} = 0. \tag{91}$$

Similarly,

$$\mathbf{E}\sum_{q\neq r} d_q\overline{d_r}U_{qp}\overline{U_{rp}}d_qU_{qa}H_{rq}\sum_{b\neq q,r} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}} = 0. \tag{92}$$

Combining (81), (82), (85), (87), (91), and (92) yields that

$$\mathbf{E}\sum_{q\neq r} d_q\overline{d_r}U_{qp}\overline{U_{rp}}\sum_{b\neq q} \overline{d_b}\,\overline{U_{ba}}\,\overline{H_{qb}}\sum_{c\neq r} d_cU_{ca}H_{rc} = 0. \tag{93}$$

Combining (68), (69), (80), and (93) yields that

$$\mathbf{E}|E_{pa}|^2 \leqslant l. \tag{94}$$

It follows from (94) that

$$\mathbf{E}\sum_{p,a=1}^{k} |E_{pa}|^2 \leqslant k^2 l. \tag{95}$$

However,

$$\|E\|^2 \leqslant \sum_{p,a=1}^{k} |E_{pa}|^2. \tag{96}$$

Combining (96) and (95) yields that

$$\mathbf{E}\|E\|^2 \leqslant k^2 l. \tag{97}$$

Combining (97) and the Markov inequality yields that

$$\|E\| \leqslant \sqrt{\beta k^2 l} \tag{98}$$

with probability at least $1 - \frac{1}{\beta}$. Combining (98) and (66) yields (67). □

4.2. *Spectral norms of various random matrices*

In this subsection, we derive bounds on the spectral norms of several random matrices.

With the choice $\alpha = 8$ and $\beta = 2$, the following lemma states that, with probability at least $\frac{1}{2}$, the least singular value of the complex $l \times k$ matrix $\mathcal{R}U$ is at least $\sqrt{\frac{l}{8}}$, and the greatest singular value is at most $\sqrt{\frac{15l}{8}}$, where $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2, U is a complex $m \times k$ matrix whose columns are orthonormal, and $m > l \geqslant 3k^2$. This lemma is similar to the subspace Johnson–Lindenstrauss lemma (Corollary 11) of [17].

Lemma 4.4. *Suppose that α and β are real numbers greater than* 1, *and k, l, and m are positive integers, such that*

$$m > l \geqslant \frac{\alpha^2\beta}{(\alpha-1)^2}k^2. \tag{99}$$

Suppose further that $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2, *U is a complex $m \times k$ matrix whose columns are orthonormal, and C is the complex $k \times k$ matrix defined via the formula*

$$C = (\mathcal{R}U)^*(\mathcal{R}U). \tag{100}$$

Then, the least (*that is, the kth greatest*) *singular value σ_k of $\mathcal{R}U$ satisfies*

$$\sigma_k = \frac{1}{\sqrt{\|C^{-1}\|}} \geqslant \sqrt{\frac{l}{\alpha}} \tag{101}$$

and (*simultaneously*) *the greatest singular value σ_1 of $\mathcal{R}U$ satisfies*

$$\sigma_1 = \sqrt{\|C\|} \leqslant \sqrt{l\left(2 - \frac{1}{\alpha}\right)} \tag{102}$$

with probability at least $1 - \frac{1}{\beta}$.

Proof. Combining (61) and (67) yields (102).

We now prove (101). It follows from (61) that

$$\|C^{-1}\| = \frac{1}{l}\left\|\left(\mathbf{1} + \frac{1}{l}\cdot E\right)^{-1}\right\|. \tag{103}$$

However,

$$\left\|\left(\mathbf{1} + \frac{1}{l}\cdot E\right)^{-1}\right\| \leqslant \sum_{j=0}^{\infty}\left\|-\frac{1}{l}\cdot E\right\|^j. \tag{104}$$

It follows from (67) that

$$\sum_{j=0}^{\infty}\left\|-\frac{1}{l}\cdot E\right\|^j \leqslant \alpha \tag{105}$$

with probability at least $1 - \frac{1}{\beta}$. Combining (103)–(105) yields (101). □

The following lemma states that the spectral norm of the $l \times m$ SRFT defined in Section 2 is at most $\sqrt{lm}$.

Lemma 4.5. *Suppose that l and m are positive integers with $l < m$, and $\mathcal{R}$ is the $l \times m$ SRFT defined in Section* 2. *Then,*

$$\|\mathcal{R}\| \leqslant \sqrt{lm}. \tag{106}$$

Proof. It follows from (6) that

$$\|\mathcal{R}\| \leqslant \|S\|\|F\|\|D\|. \tag{107}$$

However,

$$\|S\| \leqslant \sqrt{l}, \tag{108}$$

$$\|F\| \leqslant \sqrt{m}, \tag{109}$$

and

$$\|D\| = 1. \tag{110}$$

Combining (107)–(110) yields (106). □

The following lemma states that, for any matrix A, with high probability there exists a matrix T with a reasonably small spectral norm, such that $T\mathcal{R}A$ is a good approximation to A, where $\mathcal{R}$ is the SRFT defined in Section 2.

Lemma 4.6. *Suppose that k, l, m, and n are positive integers with $k \leqslant l$, such that $l < m$ and $l < n$. Suppose further that α and β are real numbers greater than 1, such that*

$$m > l \geqslant \frac{\alpha^2\beta}{(\alpha-1)^2}k^2. \tag{111}$$

Suppose finally that A is a complex $m \times n$ matrix, and $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2.

Then, there exists a complex $m \times l$ matrix T such that

$$\|T\mathcal{R}A - A\| \leqslant \sqrt{\alpha m + 1}\sigma_{k+1} \tag{112}$$

and

$$\|T\| \leqslant \sqrt{\frac{\alpha}{l}} \tag{113}$$

with probability at least $1 - \frac{1}{\beta}$, where σ_{k+1} is the $(k+1)$st greatest singular value of A.

Proof. We prove the existence of a matrix T satisfying (112) and (113) by constructing one.

We start by forming an SVD of A,

$$A = U\Sigma V^*, \tag{114}$$

where U is a complex unitary $m \times m$ matrix, Σ is a real $m \times n$ matrix whose entries are nonnegative everywhere and zero off of the main diagonal, and V is a complex unitary $n \times n$ matrix, such that

$$\Sigma_{i,i} = \sigma_i \tag{115}$$

for $i = 1, 2, \ldots, \min\{m,n\} - 1, \min\{m,n\}$, where $\Sigma_{i,i}$ is the entry in row i and column i of Σ, and σ_i is the ith greatest singular value of A.

Next, we define auxiliary matrices G and H. We define G to be the leftmost $l \times k$ block of the $l \times m$ matrix $\mathcal{R}U$, and H to be the rightmost $l \times (m-k)$ block of $\mathcal{R}U$, so that

$$\mathcal{R}U = (G|H). \tag{116}$$

We define $G^{(-1)}$ to be the complex $k \times l$ matrix given by the formula

$$G^{(-1)} = (G^*G)^{-1}G^*. \tag{117}$$

Finally, we define T to be the $m \times l$ matrix given by

$$T = U\left(\frac{G^{(-1)}}{\mathbf{0}}\right). \tag{118}$$

Combining (12), (117), (116), and (101) (using the leftmost k columns of the matrix U from the present proof as the matrix denoted U in (101) and (100)) yields that

$$\|G^{(-1)}\| \leqslant \sqrt{\frac{\alpha}{l}} \tag{119}$$

with probability at least $1 - \frac{1}{\beta}$. Combining (118), (119), and the fact that U is unitary yields (113).

We now show that T defined in (118) satisfies (112).

We define Φ to be the leftmost uppermost $k \times k$ block of Σ, and Ψ to be the rightmost lowermost $(m-k) \times (n-k)$ block of Σ, so that

$$\Sigma = \left(\begin{array}{c|c} \Phi & \mathbf{0} \\ \hline \mathbf{0} & \Psi \end{array}\right). \tag{120}$$

Combining (114), (116), and (118) yields that

$$T\mathcal{R}A - A = U\left(\left(\frac{G^{(-1)}}{\mathbf{0}}\right)(G|H) - \mathbf{1}\right)\Sigma V^{*}. \tag{121}$$

Combining (117) and (120) yields that

$$\left(\left(\frac{G^{(-1)}}{\mathbf{0}}\right)(G|H) - \mathbf{1}\right)\Sigma = \left(\begin{array}{c|c} \mathbf{0} & G^{(-1)}H\Psi \\ \mathbf{0} & -\Psi \end{array}\right). \tag{122}$$

Furthermore,

$$\left\|\left(\begin{array}{c|c} \mathbf{0} & G^{(-1)}H\Psi \\ \mathbf{0} & -\Psi \end{array}\right)\right\|^{2} \leqslant \left\|G^{(-1)}H\Psi\right\|^{2} + \|\Psi\|^{2}. \tag{123}$$

Moreover,

$$\left\|G^{(-1)}H\Psi\right\| \leqslant \left\|G^{(-1)}\right\| \|H\| \|\Psi\|. \tag{124}$$

Combining (120) and (115) yields that

$$\|\Psi\| \leqslant \sigma_{k+1}. \tag{125}$$

Combining (121)–(125) and the fact that U and V are unitary yields that

$$\|T\mathcal{R}A - A\| \leqslant \sqrt{\left\|G^{(-1)}\right\|^{2}\|H\|^{2} + 1}\,\sigma_{k+1}. \tag{126}$$

Clearly,

$$\|H\| \leqslant \left\|(G|H)\right\|. \tag{127}$$

Combining (116) and the fact that U is unitary yields that

$$\left\|(G|H)\right\| = \|\mathcal{R}\|. \tag{128}$$

Combining (127), (128), and (106) yields that

$$\|H\| \leqslant \sqrt{lm}. \tag{129}$$

Combining (126), (119), and (129) yields (112). □

4.3. Randomized linear least-squares regression

In this subsection, we derive bounds regarding randomized methods for the solution in the least-squares sense of overdetermined systems of linear-algebraic equations.

The following lemma states that, with high probability, there exists a matrix Θ with a reasonably small spectral norm, such that Θ is the inverse of the SRFT $\mathcal{R}$ (defined in Section 2) on the image under $\mathcal{R}$ of a certain subspace.

Lemma 4.7. *Suppose that α and β are real numbers greater than* 1, *and k, l, and m are positive integers, such that*

$$m > l \geqslant \frac{\alpha^{2}\beta}{(\alpha-1)^{2}}(2k)^{2}. \tag{130}$$

Suppose further that $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2, *A is a complex $m \times k$ matrix, and B is a complex $m \times k$ matrix.*

Then, there exists a complex $m \times l$ matrix Θ such that

$$\Theta\mathcal{R}A = A, \tag{131}$$

$$\Theta\mathcal{R}B = B, \tag{132}$$

and

$$\|\Theta\| \leqslant \sqrt{\frac{\alpha}{l}} \tag{133}$$

with probability at least $1 - \frac{1}{\beta}$.

Proof. We define U to be a complex matrix whose columns constitute an orthonormal basis of the subspace of $\mathbb{C}^m$ spanned by the columns of A and the columns of B. We define j to be the number of columns in U. Combining the facts that A has k columns and that B has k columns yields that

$$j \leqslant 2k. \tag{134}$$

Combining (130), (134), (101), and the fact that the columns of U are orthonormal yields that the least (that is, the jth greatest) singular value σ_j of $\mathcal{R}U$ satisfies

$$\sigma_j \geqslant \sqrt{\frac{l}{\alpha}} \tag{135}$$

with probability at least $1 - \frac{1}{\beta}$. It follows from (135) that $(\mathcal{R}U)^*(\mathcal{R}U)$ is invertible, so we can define Θ to be the complex $m \times l$ matrix

$$\Theta = U\big((\mathcal{R}U)^*(\mathcal{R}U)\big)^{-1}(\mathcal{R}U)^*. \tag{136}$$

It follows from (136) that

$$\Theta\mathcal{R}U = U. \tag{137}$$

Combining (137) and the fact that the columns of A and the columns of B are in the column space of U yields (131) and (132).

Combining (136) and the fact that the columns of U are orthonormal yields that

$$\|\Theta\| \leqslant \big\|\big((\mathcal{R}U)^*(\mathcal{R}U)\big)^{-1}(\mathcal{R}U)^*\big\|. \tag{138}$$

Combining (12) and (135) yields that

$$\big\|\big((\mathcal{R}U)^*(\mathcal{R}U)\big)^{-1}(\mathcal{R}U)^*\big\| \leqslant \sqrt{\frac{\alpha}{l}}. \tag{139}$$

Combining (138) and (139) yields (133). □

The following lemma states that, with high probability, a $k \times k$ matrix X minimizing $\|\mathcal{R}AX - \mathcal{R}B\|$ also minimizes $\|AX - B\|$ to within a small factor, where $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2, A is an $m \times k$ matrix, and B is an $m \times k$ matrix. Whereas solving $AX \approx B$ in the least-squares sense involves m simultaneous linear equations, solving $\mathcal{R}AX \approx \mathcal{R}B$ in the least-squares sense involves just l simultaneous linear equations.

Lemma 4.8. *Suppose that α and β are real numbers greater than* 1, *and k, l, and m are positive integers, such that*

$$m > l \geqslant \frac{\alpha^2\beta}{(\alpha - 1)^2}(2k)^2. \tag{140}$$

Suppose further that $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2, *A is a complex $m \times k$ matrix, B is a complex $m \times k$ matrix, X is a complex $k \times k$ matrix which minimizes the quantity*

$$\|\mathcal{R}AX - \mathcal{R}B\|, \tag{141}$$

and Y is a complex $k \times k$ matrix which minimizes the quantity

$$\|AY - B\|. \tag{142}$$

Then,

$$\|AX - B\| \leqslant \sqrt{2\alpha - 1}\|AY - B\| \tag{143}$$

with probability at least $1 - \frac{1}{\beta}$.

Proof. Combining (131) and (132) yields that

$$\|AX - B\| = \|\Theta\mathcal{R}AX - \Theta\mathcal{R}B\|. \tag{144}$$

However,

$$\|\Theta\mathcal{R}AX - \Theta\mathcal{R}B\| \leqslant \|\Theta\|\|\mathcal{R}AX - \mathcal{R}B\|. \tag{145}$$

It follows from the fact that X minimizes (141) that

$$\|\mathcal{R}AX - \mathcal{R}B\| \leqslant \|\mathcal{R}AY - \mathcal{R}B\|. \tag{146}$$

We next define U to be a matrix whose columns constitute an orthonormal basis for the subspace of $\mathbb{C}^m$ spanned by the columns of A and the columns of B, and define j to be the number of columns in U. Then, there exists a complex $j \times k$ matrix Z such that

$$AY = UZ, \tag{147}$$

and there exists a complex $j \times k$ matrix C such that

$$B = UC. \tag{148}$$

Combining (147) and (148) yields that

$$\|\mathcal{R}AY - \mathcal{R}B\| = \|\mathcal{R}UZ - \mathcal{R}UC\|. \tag{149}$$

Yet,

$$\|\mathcal{R}UZ - \mathcal{R}UC\| \leqslant \|\mathcal{R}U\|\|Z - C\|. \tag{150}$$

Combining the facts that A has k columns and that B has k columns yields that

$$j \leqslant 2k. \tag{151}$$

Combining (140), (151), (102), and the fact that the columns of U are orthonormal yields that

$$\|\mathcal{R}U\| \leqslant \sqrt{l\left(2 - \frac{1}{\alpha}\right)} \tag{152}$$

with probability at least $1 - \frac{1}{\beta}$.

It follows from the fact that the columns of U are orthonormal that

$$\|Z - C\| = \|UZ - UC\|. \tag{153}$$

Combining (147) and (148) yields that

$$\|UZ - UC\| = \|AY - B\|. \tag{154}$$

Combining (153) and (154) yields that

$$\|Z - C\| = \|AY - B\|. \tag{155}$$

Combining (144)–(146), (149), (150), (152), (155), (133), and the fact that the matrix U used in the proof of (133) is identical to the matrix U used in the present proof (so that (133) and (152) hold simultaneously with probability at least $1 - \frac{1}{\beta}$) yields (143). □

Remark 4.9. Theorem 12 in [17] motivated us to use Lemma 4.8. Lemma 4.8 and its proof are modeled after Theorem 12 in [17].

The following lemma states that, with high probability, the product PXQ^* of matrices P, X, and Q^* is a good approximation to a matrix A, provided that

1. A^*PP^* is a good approximation to A^*,
2. AQQ^* is a good approximation to A,

3. the columns of Q are orthonormal, and
4. X minimizes $\|\mathcal{R}PX - \mathcal{R}AQ\|$, where $\mathcal{R}$ is the SRFT defined in Section 2.

Lemma 4.10. *Suppose that α and β are real numbers greater than* 1*, and k, l, m, and n are positive integers, such that*

$$m > l \geqslant \frac{\alpha^2\beta}{(\alpha-1)^2}(2k)^2. \tag{156}$$

Suppose further that $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2*, A is a complex $m \times n$ matrix, P is a complex $m \times k$ matrix, Q is a complex $n \times k$ matrix whose columns are orthonormal, and X is a complex $k \times k$ matrix which minimizes the quantity*

$$\|\mathcal{R}PX - \mathcal{R}AQ\|. \tag{157}$$

Then,

$$\|PXQ^* - A\| \leqslant \sqrt{2\alpha - 1}\|A^*PP^* - A^*\| + \|AQQ^* - A\| \tag{158}$$

with probability at least $1 - \frac{1}{\beta}$.

Proof. It follows from the triangle inequality that

$$\|PXQ^* - A\| \leqslant \|PXQ^* - AQQ^*\| + \|AQQ^* - A\|. \tag{159}$$

To derive a bound on the first term in the right-hand side of (159), we observe that

$$\|PXQ^* - AQQ^*\| \leqslant \|PX - AQ\|\|Q\|. \tag{160}$$

Combining (156), (143), and the fact that X minimizes (157) yields that

$$\|PX - AQ\| \leqslant \sqrt{2\alpha - 1}\big\|P(P^*AQ) - AQ\big\| \tag{161}$$

with probability at least $1 - \frac{1}{\beta}$. However,

$$\|PP^*AQ - AQ\| \leqslant \|A^*PP^* - A^*\|\|Q\|. \tag{162}$$

It follows from the fact that the columns of Q are orthonormal that

$$\|Q\| \leqslant 1. \tag{163}$$

Combining (160)–(163) yields that

$$\|PXQ^* - AQQ^*\| \leqslant \sqrt{2\alpha - 1}\|A^*PP^* - A^*\| \tag{164}$$

with probability at least $1 - \frac{1}{\beta}$.

Combining (159) and (164) yields (158). □

5. Description of the algorithm

In this section, we describe the algorithm of the present paper. In Section 5.1, we discuss approximations to interpolative decompositions. In Section 5.2, we discuss approximations to SVDs. In Section 5.3, we discuss a usually more efficient alternative method for constructing approximations to SVDs. In Section 5.4, we tabulate the computational costs of various parts of the algorithm.

5.1. Interpolative decomposition

Suppose that k, m, and n are positive integers with $k < m$ and $k < n$, and A is a complex $m \times n$ matrix. In this subsection, we will collect together k appropriately chosen columns of A into a complex $m \times k$ matrix B, and construct a complex $k \times n$ matrix P, such that some subset of the columns of P makes up the $k \times k$ identity matrix,

$$\|P\| \leqslant \sqrt{4k(n-k)+1}, \tag{165}$$

and

$$\|BP - A\| \lesssim \sqrt{kmn}\sigma_{k+1}, \tag{166}$$

where σ_{k+1} is the $(k+1)$st greatest singular value of A. We may assume without loss of generality that m is the product of prime factors no greater than a small constant (say 2), if necessary by adjoining to A rows consisting entirely of zeros. Then, to construct matrices B and P satisfying (165) and (166), we choose an integer l near to, but greater than, k, such that $l < m$ and $l < n$ (for example, $l = k + 8$), and make the following three steps:

1. Using the algorithm of Section 3.3, compute the $l \times n$ product matrix
$$Y = \mathcal{R}A, \tag{167}$$
where $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2. (This step amounts to applying A^* to $\mathcal{R}^*$, in order to identify the range of A^*.)
2. Using Observation 3.3, form a complex $l \times k$ matrix Z whose columns constitute a subset of the columns of Y, and a complex $k \times n$ matrix P satisfying (165), such that some subset of the columns of P makes up the $k \times k$ identity matrix, and
$$\|ZP - Y\| \leqslant \sqrt{4k(n-k)+1}\eta_{k+1}, \tag{168}$$
where η_{k+1} is the $(k+1)$st greatest singular value of Y.
3. Due to Step 2, the columns of Z constitute a subset of the columns of Y. In other words, there exists a finite sequence $i_1, i_2, \ldots, i_{k-1}, i_k$ of integers such that, for any $j = 1, 2, \ldots, k-1, k$, the jth column of Z is the i_jth column of Y. Collect the corresponding columns of A into a real $m \times k$ matrix B, so that, for any $j = 1, 2, \ldots, k-1, k$, the jth column of B is the i_jth column of A.

It is easy to see that the matrices B and P satisfy (165) and (166). Indeed, Step 2 above guarantees (165) by construction. Moreover, combining (167), (168), and Remark 3.8 yields that

$$\|\mathcal{R}BP - \mathcal{R}A\| \leqslant \sqrt{4k(n-k)+1}\eta_{k+1}, \tag{169}$$

where η_{k+1} is the $(k+1)$st greatest singular value of Y. Combining (167) and the fact that η_{k+1} is the $(k+1)$st greatest singular value of Y yields that

$$\eta_{k+1} \leqslant \|\mathcal{R}\|\sigma_{k+1}, \tag{170}$$

where σ_{k+1} is the $(k+1)$st greatest singular value of A. Suppose that α and β are real numbers greater than 1, such that

$$m > l \geqslant \frac{\alpha^2\beta}{(\alpha-1)^2}k^2. \tag{171}$$

Then, combining (13), (112), (113), (165), (169), (170), and (106) yields that

$$\|BP - A\| \leqslant \Big(\big(\sqrt{4k(n-k)+1}+1\big)\sqrt{\alpha m+1} + \sqrt{4k(n-k)+1}\sqrt{\alpha m}\Big)\sigma_{k+1} \tag{172}$$

with probability at least $1 - \frac{1}{\beta}$, where σ_{k+1} is the $(k+1)$st greatest singular value of A. The bound (172) is a precise version of (166). We can use the verification scheme described in Section 3.4 to estimate $\|BP - A\|$ during each run of the algorithm.

Strictly speaking, we require that (171) hold in order to prove our theoretical bound (172). However, numerical experiments (some of which are reported in Section 6) indicate that in fact l need be only very slightly greater than k; $l = k + 8$ always worked in our experiments.

5.2. Singular value decomposition

Suppose that k, m, and n are positive integers with $k < m$ and $k < n$, such that m and n are products of prime factors no greater than a small constant (2, for example). Suppose further that A is a complex $m \times n$ matrix. In this subsection, we will construct an approximation to an SVD of A such that

$$\|U\Sigma V^* - A\| \lesssim \sqrt{\max\{m,n\}}\sigma_{k+1}, \tag{173}$$

where U is a complex $m \times k$ matrix whose columns are orthonormal, V is a complex $n \times k$ matrix whose columns are orthonormal, Σ is a real diagonal $k \times k$ matrix whose entries are all nonnegative, and σ_{k+1} is the $(k+1)$st greatest singular value of A. To do so, we choose an integer l near to, but greater than, k, such that $l < m$ and $l < n$, and make the following ten steps:

1. Using the algorithm of Section 3.3, compute the $l \times n$ product matrix
$$Y = \mathcal{R}A, \tag{174}$$
where $\mathcal{R}$ is the $l \times m$ SRFT defined in Section 2. (This step amounts to applying A^* to $\mathcal{R}^*$, in order to identify the range of A^*.)
2. Using the algorithm of Section 3.3, compute the $l \times m$ product matrix
$$\tilde{Y} = \tilde{\mathcal{R}}A^*, \tag{175}$$
where $\tilde{\mathcal{R}}$ is an $l \times n$ realization of the SRFT defined in Section 2. (This step amounts to applying A to $\tilde{\mathcal{R}}^*$, in order to identify the range of A.)
3. Using an SVD, form a complex $n \times k$ matrix Q whose columns are orthonormal, such that there exists a complex $k \times l$ matrix Z for which
$$\|QZ - Y^*\| \leqslant \eta_{k+1}, \tag{176}$$
where η_{k+1} is the $(k+1)$st greatest singular value of Y, and Y is defined in (174). (See Observation 3.5 for details concerning the construction of such a matrix Q.)
4. Using an SVD, form a complex $m \times k$ matrix P whose columns are orthonormal, such that there exists a complex $k \times l$ matrix $\tilde{Z}$ for which
$$\|P\tilde{Z} - \tilde{Y}^*\| \leqslant \tilde{\eta}_{k+1}, \tag{177}$$
where $\tilde{\eta}_{k+1}$ is the $(k+1)$st greatest singular value of $\tilde{Y}$, and $\tilde{Y}$ is defined in (175). (See Observation 3.5 for details concerning the construction of such a matrix P.)
5. Using the algorithm of Section 3.3, compute the $l \times k$ product matrix
$$W = \mathcal{R}P, \tag{178}$$
where $\mathcal{R}$ is the same realization of the $l \times m$ SRFT as in (174), and P is from (177).
6. Compute the $l \times k$ product matrix
$$B = YQ, \tag{179}$$
where Y is defined in (174), and Q is from (176).
7. Compute the complex $k \times k$ matrix X which minimizes the quantity
$$\|WX - B\|, \tag{180}$$
where W is defined in (178), and B is defined in (179). (See, for example, Section 5.3 in [8] for details concerning the construction of such a minimizing X.)
8. Construct an SVD of X from (180), that is,
$$X = U^X \Sigma \big(V^X\big)^*, \tag{181}$$
where U^X is a complex $k \times k$ matrix whose columns are orthonormal, V^X is a complex $k \times k$ matrix whose columns are orthonormal, and Σ is a real diagonal $k \times k$ matrix whose entries are all nonnegative. (See, for example, Chapter 8 in [8] for details concerning the construction of such an SVD.)
9. Compute the $m \times k$ product matrix
$$U = PU^X, \tag{182}$$
where P is from (177), and U^X is from (181).
10. Compute the $n \times k$ product matrix
$$V = QV^X, \tag{183}$$
where Q is from (176), and V^X is from (181).

It is clear that the columns of U are orthonormal, as are the columns of V, and that the entries of Σ are all nonnegative and are zero off of the main diagonal. It is easy to see that the matrices U, Σ, and V satisfy (173). Indeed, combining (180), (178), (179), and (174) yields that X minimizes the quantity (157). Suppose that α and β are real numbers greater than 1, such that

$$m > l \geqslant \frac{\alpha^2 \beta}{(\alpha - 1)^2}(2k)^2. \tag{184}$$

Then, combining (184) and the fact that X minimizes the quantity (157) yields (158), that is,

$$\|PXQ^* - A\| \leqslant \sqrt{2\alpha - 1}\|A^*PP^* - A^*\| + \|AQQ^* - A\| \tag{185}$$

with probability at least $1 - \frac{1}{\beta}$.

We bound $\|AQQ^* - A\|$ first, then $\|A^*PP^* - A^*\|$. It follows from (174) that

$$\eta_{k+1} \leqslant \|\mathcal{R}\|\sigma_{k+1}, \tag{186}$$

where η_{k+1} is the $(k+1)$st greatest singular value of Y, and σ_{k+1} is the $(k+1)$st greatest singular value of A. Combining (18), (112), (113), (176), (174), (186), and (106) yields that

$$\|AQQ^* - A\| \leqslant 2(\sqrt{\alpha m + 1} + \sqrt{\alpha m})\sigma_{k+1} \tag{187}$$

with probability at least $1 - \frac{1}{\beta}$.

Similarly,

$$\|A^*PP^* - A^*\| \leqslant 2(\sqrt{\alpha n + 1} + \sqrt{\alpha n})\sigma_{k+1} \tag{188}$$

with probability at least $1 - \frac{1}{\beta}$.

Combining (185), (187), and (188) yields that

$$\|PXQ^* - A\| \leqslant 2(\sqrt{2\alpha - 1} + 1)\big(\sqrt{\alpha \max\{m,n\} + 1} + \sqrt{\alpha \max\{m,n\}}\big)\sigma_{k+1} \tag{189}$$

with probability at least $1 - \frac{3}{\beta}$. Combining (189) and (181)–(183) yields that

$$\|U\Sigma V^* - A\| \leqslant 2(\sqrt{2\alpha - 1} + 1)\big(\sqrt{\alpha \max\{m,n\} + 1} + \sqrt{\alpha \max\{m,n\}}\big)\sigma_{k+1} \tag{190}$$

with probability at least $1 - \frac{3}{\beta}$. The bound (190) is a precise version of (173). We can use the verification scheme described in Section 3.4 to estimate $\|U\Sigma V^* - A\|$ during each run of the algorithm.

Remark 5.1. Step 7 is motivated by an idea from [1,7,16] of using the SRFT defined in Section 2 for the purpose of computing a solution in the least-squares sense to an overdetermined system of linear-algebraic equations.

Remark 5.2. It is possible to replace the $l \times n$ matrix Y defined in (174) with a $k \times n$ matrix, by applying the algorithm of Section 5.1 to Y^{T} and using the transpose of the obtained $n \times k$ matrix in place of Y. Similarly, it is possible to replace the $l \times m$ matrix $\tilde{Y}$ defined in (175) with a $k \times m$ matrix, by applying the algorithm of Section 5.1 to $\tilde{Y}^{\mathrm{T}}$ and using the transpose of the obtained $m \times k$ matrix in place of $\tilde{Y}$. Furthermore, it is possible to obtain replacements for Y and $\tilde{Y}$ by using "QR" decompositions or SVDs in place of the interpolative decompositions employed by the algorithm of Section 5.1.

5.3. Singular value decomposition by means of the interpolative decomposition

In this subsection, we provide an alternative to the algorithm of Section 5.2 for computing an approximation to a singular value decomposition. This alternative is usually somewhat more efficient.

Suppose that k, m, and n are positive integers with $k < m$ and $k < n$, and A is a complex $m \times n$ matrix. We will compute an approximation to an SVD of A such that

$$\|U\Sigma V^* - A\| \lesssim \sqrt{kmn}\sigma_{k+1}, \tag{191}$$

where U is a complex $m \times k$ matrix whose columns are orthonormal, V is a complex $n \times k$ matrix whose columns are orthonormal, Σ is a real diagonal $k \times k$ matrix whose entries are all nonnegative, and σ_{k+1} is the $(k+1)$st greatest singular value of A. To do so, we choose an integer l near to, but greater than, k, such that $l < m$ and $l < n$ (for example, $l = k + 8$), and use the algorithm of Section 5.1 to construct the matrices B and P in (165) and (166). Then, we make the following four steps:

1. Construct a lower triangular complex $k \times k$ matrix L, and a complex $n \times k$ matrix Q whose columns are orthonormal, such that
$$P = LQ^*. \tag{192}$$
(See Remark 3.11 for details concerning the construction of such matrices L and Q.)
2. Compute the $m \times k$ product matrix
$$C = BL. \tag{193}$$
3. Construct an SVD of C, that is,
$$C = U\Sigma W^*, \tag{194}$$
where U is a complex $m \times k$ matrix whose columns are orthonormal, Σ is a real diagonal $k \times k$ matrix whose entries are all nonnegative, and W is a complex $k \times k$ matrix whose columns are orthonormal. (See, for example, Chapter 8 in [8] for details concerning the construction of such an SVD.)
4. Compute the $n \times k$ product matrix
$$V = QW. \tag{195}$$

It is clear that the columns of U are orthonormal, as are the columns of V, and that the entries of Σ are all nonnegative and are zero off of the main diagonal. It is easy to see that the matrices U, Σ, and V satisfy (191). Indeed, suppose that α and β are real numbers greater than 1, such that
$$m > l \geqslant \frac{\alpha^2\beta}{(\alpha-1)^2}k^2. \tag{196}$$
Then, combining (172) and Lemma 3.10 yields that
$$\|U\Sigma V^* - A\| \leqslant \left(\left(\sqrt{4k(n-k)+1}+1\right)\sqrt{\alpha m+1} + \sqrt{4k(n-k)+1}\sqrt{\alpha m}\right)\sigma_{k+1} \tag{197}$$
with probability at least $1 - \frac{1}{\beta}$, where σ_{k+1} is the $(k+1)$st greatest singular value of A. The bound (197) is a precise version of (191). We can use the verification scheme described in Section 3.4 to estimate $\|U\Sigma V^* - A\|$ during each run of the algorithm.

Strictly speaking, we require that (196) hold in order to prove our theoretical bound (197). However, numerical experiments (some of which are reported in Section 6) indicate that in fact l need be only very slightly greater than k; $l = k + 8$ always worked in our experiments.

Remark 5.3. Steps 2 and 4 in the procedure of the present subsection are somewhat subtle numerically. Both Steps 2 and 4 involve constructing products of matrices, and in general constructing the product $\Xi\Omega$ of matrices Ξ and Ω can be numerically unstable. Indeed, in general some entries of Ξ or Ω can have unmanageably large absolute values, while in exact arithmetic no entry of the product $\Xi\Omega$ has an unmanageably large absolute value; in such circumstances, constructing the product $\Xi\Omega$ can be unstable in finite-precision arithmetic. However, this problem does not arise in Steps 2 and 4 above, due to (165), (25), the fact that the columns of B constitute a subset of the columns of A (so that $\|B\| \leqslant \|A\|$), and the fact that the columns of Q are orthonormal, as are the columns of W.

5.4. Costs

In this subsection, we tabulate the numbers of floating-point operations required by the algorithm described in Sections 5.1–5.3, as applied once to a complex $m \times n$ matrix A.

5.4.1. Interpolative decomposition

The algorithm of Section 5.1 incurs the following costs in order to compute an approximation to an interpolative decomposition of A:

1. Computing Y in (167) costs $\mathcal{O}(mn\log(l))$.
2. Computing Z and P in (168) costs $\mathcal{O}(lkn\log(n))$.
3. Forming B in Step 3 requires retrieving k columns of the $m \times n$ matrix A, which costs $\mathcal{O}(km)$.

The verification scheme of Section 3.4 requires applying A, B, and P to a fixed number (say 6) of vectors, at costs of $\mathcal{O}(mn)$, $\mathcal{O}(km)$, and $\mathcal{O}(kn)$. Summing up the costs for Steps 1–3 above and for the verification scheme, we conclude that the algorithm of Section 5.1 costs

$$C_{\mathrm{ID}} = \mathcal{O}\big(mn\log(l) + lkn\log(n)\big). \tag{198}$$

Remark 5.4. When "QR" decompositions are used as in [5] to compute the matrices Z and P in (168), the cost of the algorithm of Section 5.1 is usually less than the cost of the algorithm of Section 5.2, typically

$$C'_{\mathrm{ID}} = \mathcal{O}\big(mn\log(l) + lkn\big). \tag{199}$$

5.4.2. Singular value decomposition

The algorithm of Section 5.2 incurs the following costs in order to compute an approximation to a singular value decomposition of A:

1. Computing Y in (174) costs $\mathcal{O}(mn\log(l))$.
2. Computing $\tilde{Y}$ in (175) costs $\mathcal{O}(mn\log(l))$.
3. Computing Q in (176) costs $\mathcal{O}(l^2 n)$.
4. Computing P in (177) costs $\mathcal{O}(l^2 m)$.
5. Computing W in (178) costs $\mathcal{O}(km\log(l))$.
6. Computing B in (179) costs $\mathcal{O}(lkn)$.
7. Computing X minimizing (180) costs $\mathcal{O}(k^2 l)$.
8. Computing the SVD (181) of X costs $\mathcal{O}(k^3)$.
9. Computing U in (182) costs $\mathcal{O}(k^2 m)$.
10. Computing V in (183) costs $\mathcal{O}(k^2 n)$.

The verification scheme of Section 3.4 requires applying A, U, Σ, and V^* to a fixed number (say 6) of vectors, at costs of $\mathcal{O}(mn)$, $\mathcal{O}(km)$, $\mathcal{O}(k)$, and $\mathcal{O}(kn)$. Summing up the costs for Steps 1–10 above and for the verification scheme, we conclude that the algorithm of Section 5.2 costs

$$C_{\mathrm{SVD}} = \mathcal{O}\big(mn\log(l) + l^2(m+n)\big). \tag{200}$$

Remark 5.5. With the modifications described in Remark 5.2, the scheme of Section 5.2 costs

$$C'_{\mathrm{SVD}} = \mathcal{O}\big(mn\log(l) + k^2(m+n) + kl^2\log(l)\big). \tag{201}$$

(201) can be less than (200) when l is large. However, while our current theoretical bounds require $l > k^2$ to guarantee good accuracy, our numerical experiments (some of which are reported in Section 6) indicate that $l \geqslant k+5$ suffices. For $l = k+5$, the estimate (200) is as tight as (201).

5.4.3. Singular value decomposition by means of the interpolative decomposition

The algorithm of Section 5.3 incurs the following costs in order to compute an approximation to a singular value decomposition of A, in addition to (198):

1. Computing L and Q in (192) costs $\mathcal{O}(k^2 n)$.
2. Computing C in (193) costs $\mathcal{O}(k^2 m)$.

Table 1
ID of the 4096 × 4096 matrix A defined in (204)

k	l	σ_{k+1}	δ_{direct}	δ_{fast}	$\delta_{\text{fast}}/\delta_{\text{est}}$	t_{direct}	t_{fast}	$t_{\text{direct}}/t_{\text{fast}}$
8	16	0.100E1	0.100E1	0.177E1	41.7	0.29E1	0.17E1	1.7
24	32	0.534E–7	0.755E–7	0.364E–6	41.2	0.79E1	0.19E1	4.1
56	64	0.588E–9	0.578E–7	0.997E–8	34.5	0.18E2	0.22E1	8.2
120	128	0.687E–11	0.178E–7	0.514E–9	39.0	0.37E2	0.29E1	13
248	256	0.622E–11	0.561E–6	0.407E–9	46.2	0.76E2	0.60E1	13
504	512	0.201E–11	0.162E–6	0.339E–9	29.9	0.16E3	0.28E2	5.7
1016	1024	0.150E–11	0.651E–6	0.285E–9	34.1	0.32E3	0.12E3	2.7

Table 2
ID of the 2048 × 2048 matrix A effecting convolution with γ defined in (213)

k	l	σ_{k+1}	δ_{direct}	δ_{fast}	$\delta_{\text{fast}}/\delta_{\text{est}}$	t_{direct}	t_{fast}	$t_{\text{direct}}/t_{\text{fast}}$
8	16	0.225E–5	0.319E–5	0.823E–5	5.06	0.72E0	0.42E0	1.7
24	32	0.187E–8	0.148E–7	0.184E–7	7.58	0.20E1	0.48E0	4.1
56	64	0.459E–10	0.119E–7	0.793E–9	9.59	0.44E1	0.58E0	7.6
120	128	0.687E–11	0.569E–7	0.118E–9	12.7	0.92E1	0.96E0	9.6
248	256	0.263E–11	0.181E–8	0.774E–10	10.8	0.18E2	0.24E1	7.4
504	512	0.162E–11	0.981E–11	0.759E–10	11.1	0.36E2	0.13E2	2.8
1016	1024	0.127E–11	0.851E–11	0.723E–10	11.9	0.72E2	0.46E2	1.6

Table 3
ID of the 1024 × 1024 matrix A defined in (216)

k	l	σ_{k+1}	δ_{direct}	δ_{fast}	$\delta_{\text{fast}}/\delta_{\text{est}}$	t_{direct}	t_{fast}	$t_{\text{direct}}/t_{\text{fast}}$
8	16	0.225E–5	0.342E–5	0.100E–4	14.2	0.17E0	0.10E0	1.7
24	32	0.187E–8	0.383E–8	0.163E–7	18.1	0.47E0	0.12E0	3.9
56	64	0.459E–10	0.127E–9	0.819E–9	14.5	0.11E1	0.16E0	6.5
120	128	0.687E–11	0.246E–10	0.213E–9	19.7	0.21E1	0.32E0	6.7
248	256	0.263E–11	0.133E–10	0.119E–9	14.9	0.46E1	0.97E0	4.7
504	512	0.162E–11	0.956E–11	0.117E–9	16.8	0.86E1	0.48E1	1.8

3. Computing the SVD of C in (194) costs $\mathcal{O}(k^2m)$.
4. Computing V in (195) costs $\mathcal{O}(k^2n)$.

The verification scheme of Section 3.4 requires applying A, U, Σ, and V^* to a fixed number (say 6) of vectors, at costs of $\mathcal{O}(mn)$, $\mathcal{O}(km)$, $\mathcal{O}(k)$, and $\mathcal{O}(kn)$. Summing up the costs for Steps 1–4 above and for the verification scheme, plus (198), we conclude that the algorithm of Section 5.3 costs

$$C_{\text{SVD(ID)}} = \mathcal{O}\big(mn\log(l) + lkn\log(n) + k^2m\big). \tag{202}$$

Remark 5.6. As in Remark 5.4, when "QR" decompositions are used as in [5] to compute the matrices Z and P in (168), the cost of the algorithm of Section 5.3 is usually less than the cost of the algorithm of Section 5.2, typically

$$C'_{\text{SVD(ID)}} = \mathcal{O}\big(mn\log(l) + lkn + k^2m\big). \tag{203}$$

6. Numerical examples

In this section, we describe the results of several numerical tests of the algorithm of the present paper. Tables 1–6 summarize the numerical output of applying the algorithm to the matrix A defined below for each of the examples.

In all of the tables, we set $l = k + 8$ for the user-specified parameter l, where k is the rank of the approximations constructed by the algorithms. As described below, many of the entries in the tables report the worst or average results over multiple trials of the randomized algorithms. We conducted 30 randomized trials for Tables 1 and 4, 100

Table 4
SVD of the 4096 × 4096 matrix A defined in (204)

k	l	σ_{k+1}	δ_{direct}	δ_{fast}	$\delta_{\text{fast}}/\delta_{\text{est}}$	t_{direct}	t_{fast}	$t_{\text{direct}}/t_{\text{fast}}$
8	16	0.100E1	0.100E1	0.177E1	41.7	0.29E1	0.17E1	1.7
24	32	0.534E–7	0.755E–7	0.364E–6	41.2	0.80E1	0.20E1	4.1
56	64	0.588E–9	0.575E–7	0.997E–8	34.5	0.19E2	0.28E1	6.7
120	128	0.687E–11	0.682E–8	0.514E–9	39.0	0.41E2	0.59E1	6.9
248	256	0.622E–11	0.127E–7	0.407E–9	46.2	0.87E2	0.19E2	4.7
504	512	0.201E–11	0.150E–7	0.339E–9	29.9	0.19E3	0.79E2	2.4
1016	1024	0.150E–11	0.689E–8	0.285E–9	34.1	0.46E3	0.35E3	1.3

Table 5
SVD of the 2048 × 2048 matrix A effecting convolution with γ defined in (213)

k	l	σ_{k+1}	δ_{direct}	δ_{fast}	$\delta_{\text{fast}}/\delta_{\text{est}}$	t_{direct}	t_{fast}	$t_{\text{direct}}/t_{\text{fast}}$
8	16	0.225E–5	0.319E–5	0.823E–5	5.06	0.73E0	0.41E0	1.8
24	32	0.187E–8	0.148E–7	0.184E–7	7.58	0.20E1	0.52E0	3.9
56	64	0.459E–10	0.956E–8	0.793E–9	9.59	0.47E1	0.85E0	5.5
120	128	0.687E–11	0.273E–7	0.178E–9	12.7	0.11E2	0.23E1	4.6
248	256	0.263E–11	0.159E–8	0.774E–10	10.8	0.24E2	0.86E1	2.8
504	512	0.162E–11	0.981E–11	0.759E–10	11.1	0.58E2	0.42E2	1.4
1016	1024	0.127E–11	0.851E–11	0.723E–10	11.9	0.16E3	0.18E3	0.9

Table 6
SVD of the 1024 × 1024 matrix A defined in (216)

k	l	σ_{k+1}	δ_{direct}	δ_{fast}	$\delta_{\text{fast}}/\delta_{\text{est}}$	t_{direct}	t_{fast}	$t_{\text{direct}}/t_{\text{fast}}$
8	16	0.225E–5	0.342E–5	0.100E–4	14.2	0.18E0	0.11E0	1.6
24	32	0.187E–8	0.383E–8	0.163E–7	18.1	0.51E0	0.15E0	3.4
56	64	0.459E–10	0.127E–9	0.819E–9	14.5	0.12E1	0.30E0	3.9
120	128	0.687E–11	0.246E–10	0.213E–9	19.7	0.27E1	0.90E0	3.0
248	256	0.263E–11	0.133E–10	0.119E–9	14.9	0.68E1	0.41E1	1.7
504	512	0.162E–11	0.956E–11	0.117E–9	16.8	0.20E2	0.20E2	1.0

randomized trials for Tables 2 and 5, and 500 randomized trials for Tables 3 and 6. For the verification scheme of Section 3.4, we ran 6 independent tests of each randomized approximation matrix, exactly as described in Section 3.4.

Tables 1–3 display the results of applying the interpolative decomposition algorithm of Section 5.1 once to the matrix A defined below for each example. Tables 4–6 display the results of applying the singular value decomposition algorithm of Section 5.3. The numerical experiments reported in [21] indicate that the algorithm of Section 5.2 is not competitive with the algorithm of Section 5.3 in terms of either accuracy or efficiency.

In all of the tables, k is the rank of the approximations constructed by the algorithms, and σ_{k+1} is the $(k+1)$st greatest singular value of A; σ_{k+1} is also the spectral norm of the difference between the original matrix A and its best rank-k approximation.

In Tables 1–3, δ_{direct} is the spectral norm of the difference between the original matrix A and the approximation BP to an interpolative decomposition obtained via the pivoted "QR" decomposition algorithm of [5] that is based upon plane (Householder) reflections. In Tables 4–6, δ_{direct} is the spectral norm of the difference between the original matrix A and the approximation $U\Sigma V^*$ to an SVD obtained via following up a pivoted "QR" decomposition algorithm based upon plane (Householder) reflections with a call to the LAPACK 3.1.1 divide-and-conquer SVD routine `dgesdd`.

In Tables 1–3, δ_{fast} is the maximum over multiple randomized trials of the spectral norm of the difference between the original matrix A and the approximation BP to an interpolative decomposition obtained via the randomized algorithm. In Tables 4–6, δ_{fast} is the maximum over multiple randomized trials of the spectral norm of the difference between the original matrix A and the approximation $U\Sigma V^*$ to an SVD obtained via the randomized algorithm.

In Tables 1–3, $\delta_{\text{fast}}/\delta_{\text{est}}$ is the maximum over multiple randomized trials of the factor by which the randomized verification scheme of Section 3.4 underestimated the spectral norm of the difference between A and the approximation BP to an interpolative decomposition obtained via the randomized algorithm. In Tables 4–6, $\delta_{\text{fast}}/\delta_{\text{est}}$ is the maximum over multiple randomized trials of the factor by which the randomized verification scheme underestimated the spectral norm of the difference between A and the approximation $U\Sigma V^*$ to an SVD obtained via the randomized algorithm.

In Tables 1–3, t_{direct} is the number of seconds of CPU time taken by the pivoted "QR" decomposition algorithm of [5] that is based upon plane (Householder) reflections. In Tables 4–6, t_{direct} is the number of seconds of CPU time taken by the combination of a pivoted "QR" decomposition algorithm based upon plane (Householder) reflections and the LAPACK 3.1.1 divide-and-conquer SVD routine `dgesdd`.

In all of the tables, t_{fast} is the average over multiple randomized trials of the number of seconds of CPU time taken by the randomized algorithm plus the number of seconds taken by the verification scheme of Section 3.4; every approximation produced by the randomized algorithm passed the verification test during our experiments (as well as during all of our experiments with $l \geqslant k+5$).

The values of δ_{direct} and δ_{fast} displayed in the tables are those obtained via the power method for estimating the spectral norm of a matrix.

Tables 1 and 4 report the results of applying the algorithms of Sections 5.1 and 5.3 to the 4096 × 4096 matrix defined via the formula

$$A = \sum_{k=1}^{l+2} u^{(k)} \sigma_k \big(v^{(k)}\big)^*, \tag{204}$$

where

$$\sigma_k = 10^{-120\cdot([k-1] \bmod 10)/(l+1)} \tag{205}$$

for $k = 1, 2, \ldots, l+1, l+2$,

$$\big(v^{(k)}\big)_j = \frac{1}{\sqrt{4096}} e^{2\pi ijk/4096} \tag{206}$$

for $j = 1, 2, \ldots, 4095, 4096$ and $k = 1, 2, \ldots, l+1, l+2$, and

$$\big(u^{(1)}\big)^{\mathrm{T}} = \frac{1}{\sqrt{n-1}} (1 \quad 1 \quad \ldots \quad 1 \quad 1 \quad 0), \tag{207}$$

$$\big(u^{(2)}\big)^{\mathrm{T}} = (0 \quad 0 \quad \ldots \quad 0 \quad 0 \quad 1), \tag{208}$$

$$\big(u^{(3)}\big)^{\mathrm{T}} = \frac{1}{\sqrt{n-2}} (1 \quad -1 \quad 1 \quad -1 \quad \ldots \quad 1 \quad -1 \quad 1 \quad -1 \quad 0 \quad 0), \tag{209}$$

$$\big(u^{(4)}\big)^{\mathrm{T}} = \frac{1}{\sqrt{2}} (1 \quad 0 \quad -1 \quad 0 \quad 0 \quad \ldots \quad 0 \quad 0), \tag{210}$$

$$\big(u^{(5)}\big)^{\mathrm{T}} = \frac{1}{\sqrt{2}} (0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad -1 \quad 0 \quad 0 \quad \ldots \quad 0 \quad 0), \tag{211}$$

$$\big(u^{(6)}\big)^{\mathrm{T}} = \frac{1}{\sqrt{2}} (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad -1 \quad 0 \quad 0 \quad \ldots \quad 0 \quad 0), \tag{212}$$

and so on. More precisely, for $k = 4, 5, \ldots, l+1, l+2$, the $(4k-15)$th and $(4k-13)$th entries of $u^{(k)}$ are $\frac{1}{\sqrt{2}}$ and $-\frac{1}{\sqrt{2}}$, and all other entries of $u^{(k)}$ are zero. Obviously, $\sigma_1, \sigma_2, \ldots, \sigma_{l+1}, \sigma_{l+2}$ are the nonzero singular values of A. We note that $\|A\| = \sigma_1 = 1$.

Tables 2 and 5 report the results of applying the algorithms of Sections 5.1 and 5.3 to the 2048 × 2048 matrix A effecting convolution with the complex 2048 × 1 column vector γ with the entry

$$\gamma_j = \frac{1}{2048} \sum_{k=1}^{2048} e^{-2\pi i(j-1)(k-1)/2048} \sigma_k \tag{213}$$

for $j = 1, 2, \ldots, 2047, 2048$, where

$$\sigma_k = 10^{-24\cdot([k-1] \bmod 2)/(l+1)} \tag{214}$$

for $k = 1, 2, \dots, l+1, l+2$, and

$$\sigma_k = 0 \tag{215}$$

otherwise (for $k = l+3, l+4, \dots, 2047, 2048$). Combining the facts that A effects convolution with γ, and that γ is a discrete Fourier transform of $\sigma_1, \sigma_2, \dots, \sigma_{2047}, \sigma_{2048}$, yields that $\sigma_1, \sigma_2, \dots, \sigma_{2047}, \sigma_{2048}$ are the singular values of A. We note that $\|A\| = \sigma_1 = 1$.

Tables 3 and 6 report the results of applying the algorithms of Sections 5.1 and 5.3 to the 1024 × 1024 matrix A defined via the formula

$$A = \sum_{k=1}^{l+2} u^{(k)} \sigma_k \big(v^{(k)}\big)^*, \tag{216}$$

where

$$\sigma_k = 10^{-12\cdot(k-1)/(l+1)} \tag{217}$$

for $k = 1, 2, \dots, l+1, l+2$, and $u^{(1)}, u^{(2)}, \dots, u^{(l+1)}, u^{(l+2)}$ and $v^{(1)}, v^{(2)}, \dots, v^{(l+1)}, v^{(l+2)}$ are two independent sets of orthonormal vectors obtained by applying the Gram–Schmidt process to vectors whose entries are drawn i.i.d. from a pseudorandom number generator with a complex Gaussian distribution of zero mean and unit variance. Obviously, $\sigma_1, \sigma_2, \dots, \sigma_{l+1}, \sigma_{l+2}$ are the nonzero singular values of A. We note that $\|A\| = \sigma_1 = 1$.

We performed all computations using IEEE standard double-precision variables, whose mantissas have approximately one bit of precision less than 16 digits (so that the relative precision of the variables is approximately 0.2E–15). We ran all computations on one core of a 1.86 GHz Intel Centrino Core Duo microprocessor with 2 MB of L2 cache and 1 GB of RAM. We compiled the Fortran 77 code using the Lahey/Fujitsu Linux Express v6.2 compiler, with the optimization flag `--o2` enabled. We used a double-precision version of P.N. Swarztrauber's FFTPACK library for the fast Fourier transforms required by Step 1 in the algorithm of Section 3.3. We used the LAPACK 3.1.1 divide-and-conquer SVD routine `dgesdd` to compute the SVDs in Steps 3, 4, and 8 in the algorithm of Section 5.2 and in Step 3 in the algorithm of Section 5.3. For the numbers $s_1, s_2, \dots, s_{l-1}, s_l$ used to construct the matrix S in (6), we drew numbers uniformly at random without replacement from $\{1, 2, \dots, m-1, m\}$; while deriving our theoretical bounds, we assumed that the numbers were drawn with replacement.

Remark 6.1. Tables 1–6 indicate that the algorithms of Sections 5.1 and 5.3 are generally more efficient than the classical pivoted "QR" decomposition algorithm based on plane (Householder) reflections, followed by either the algorithm of [5] or the LAPACK 3.1.1 divide-and-conquer SVD routine.

Remark 6.2. The numerical experiments reported in [21] indicate that the algorithm of Section 5.2 is not competitive with the algorithm of Section 5.3 in terms of either accuracy or efficiency. However, the algorithm of Section 5.2 is of theoretical interest.

Remark 6.3. The entries in the tables for $\delta_{\text{fast}}/\delta_{\text{est}}$ are all less than $8\sqrt{n}$, in accord with (38) for 6 verification trials, where A is $n \times n$ (in fact, the values are all less than $\sqrt{n}$).

7. Conclusions and generalizations

This paper provides an algorithm for the low-rank approximation of arbitrary matrices. Given the entries of a matrix A, the algorithm provides a means for computing several of the greatest singular values and corresponding singular vectors of A; it is generally faster than existing schemes, can access each column of A independently and at most twice, and parallelizes easily.

The theoretical bounds derived in the present paper should be considered preliminary. Our numerical experiments indicate that the algorithm performs better than our estimates guarantee. Comfortingly, the verification scheme of Section 3.4 provides an inexpensive means for determining the precision of the approximation obtained during every run. If the algorithm were to produce an approximation that were less accurate than desired, then one could run the algorithm again with an independent realization of the random variables involved, in effect boosting the probability of success at a reasonable additional expected cost.

Nevertheless, the randomized algorithm produced an approximation accurate to within 3 digits of the best possible during every trial reported in the numerical experiments of Section 6, obviating the need to run the algorithm again (assuming that k was chosen sufficiently large that the accuracy of the best possible rank-k approximation was sufficiently high). We are currently investigating our empirical observation that the algorithm produces a nearly optimal approximation whenever the user-specified parameter l is only very slightly greater than the rank k of the approximation. Our current theoretical bounds require $l > k^2$ in order to ensure good accuracy.

The algorithm of the present article admits several generalizations along the lines discussed in [14], namely:

1. If the singular values of the matrix being approximated decay sufficiently fast, then the factors of $\sqrt{m}$ in (166) and (191), and of $\sqrt{\max\{m,n\}}$ in (173), would appear to be superfluous, both in theory and in practice.
2. In the present article, the rank k of the approximation to be constructed and the user-specified parameter l are fixed. In practice, one adjusts k and l during the course of the algorithm in order to guarantee that the approximation attains a prescribed accuracy, preferably using as small a number l as possible.
3. The present article constructs approximations to interpolative decompositions and to singular value decompositions. We have constructed a similar algorithm for approximating the Schur decomposition (see, for example, Theorem 7.1.3 and the surrounding discussion in [8] for a description of the Schur decomposition).
4. The present paper uses complex arithmetic. When processing real matrices, one should use only real arithmetic.

Acknowledgments

We would like to thank Nir Ailon, Tamás Sarlós, Yoel Shkolnisky, Amit Singer, Lexing Ying, and Steven Zucker for useful discussions.

References

[1] N. Ailon, B. Chazelle, Approximate nearest neighbors and the fast Johnson–Lindenstrauss transform, SIAM J. Comput. (2007), in press.

[2] N. Ailon, E. Liberty, Fast dimension reduction using Rademacher series on dual BCH codes, Tech. rep. 1385, Yale University, Department of Computer Science, July 2007.

[3] S. Ar, M. Blum, B. Codenotti, P. Gemmell, Checking approximate computations over the reals, in: Proc. 25th Annual ACM Symposium on the Theory of Computing, 1993, pp. 786–795.

[4] T.F. Chan, P.C. Hansen, Some applications of the rank-revealing QR factorization, SIAM J. Sci. Statist. Comput. 13 (1992) 727–741.

[5] H. Cheng, Z. Gimbutas, P.-G. Martinsson, V. Rokhlin, On the compression of low rank matrices, SIAM J. Sci. Comput. 26 (2005) 1389–1404.

[6] J.D. Dixon, Estimating extremal eigenvalues and condition numbers of matrices, SIAM J. Numer. Anal. 20 (1983) 812–814.

[7] P. Drineas, M. Mahoney, S. Muthukrishnan, Polynomial time algorithm for column-row-based relative-error low-rank matrix approximation, Tech. rep. 2006-04, DIMACS, March 2006.

[8] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[9] S.A. Goreinov, E.E. Tyrtyshnikov, The maximal-volume concept in approximation by low-rank matrices, in: V. Olshevsky (Ed.), Structured Matrices in Mathematics, Computer Science, and Engineering I, Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference, University of Colorado, Boulder, June 27–July 1, 1999, in: Contemporary Mathematics, vol. 280, AMS, Providence, RI, 2001.

[10] M. Gu, S.C. Eisenstat, Efficient algorithms for computing a strong rank-revealing QR factorization, SIAM J. Sci. Comput. 17 (1996) 848–869.

[11] J. Kuczyński, H. Woźniakowski, Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start, SIAM J. Matrix Anal. Appl. 13 (1992) 1094–1122.

[12] S.L. Lee, G. Strang, Row reduction of a matrix and $A = CaB$, Amer. Math. Monthly 107 (2000) 681–688.

[13] P.-G. Martinsson, V. Rokhlin, M. Tygert, On interpolation and integration in finite-dimensional spaces of bounded functions, Comm. Appl. Math. Comput. Sci. 1 (2006) 133–142.

[14] P.-G. Martinsson, V. Rokhlin, M. Tygert, A randomized algorithm for the approximation of matrices, Tech. rep. 1361, Yale University, Department of Computer Science, June 2006.

[15] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes, second ed., Cambridge University Press, Cambridge, UK, 1992.

[16] T. Sarlós, Improved approximation algorithms for large matrices via random projections, in: Proc. FOCS 2006, 47th Annual IEEE Symposium on Foundations of Computer Science, October 2006, pp. 143–152.

[17] T. Sarlós, Improved approximation algorithms for large matrices via random projections, revised, extended long form, Manuscript in preparation, 2006, currently available at: http://www.ilab.sztaki.hu/~stamas/publications/rp-long.pdf.

[18] H.V. Sorensen, C.S. Burrus, Efficient computation of the DFT with only a subset of input or output points, IEEE Trans. Signal Process. 41 (1993) 1184–1200.

[19] E.E. Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, Computing 64 (2000) 367–380.

[20] F. Woolfe, E. Liberty, V. Rokhlin, M. Tygert, A fast randomized algorithm for the approximation of matrices, Tech. rep. 1386, Yale University, Department of Computer Science, July 2007.

[21] F. Woolfe, E. Liberty, V. Rokhlin, M. Tygert, A fast randomized algorithm for the approximation of matrices—Preliminary report, Tech. rep. 1380, Yale University, Department of Computer Science, April 2007.