Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix

Petros Drineas [*] Ravi Kannan [†‡] Michael W. Mahoney [§]

Technical Report, YALEU/DCS/TR-1270, February 2004.

Abstract

In many applications, the data consist of (or may be naturally formulated as) an $m \times n$ matrix A. It is often of interest to find a low-rank approximation to A, i.e., an approximation D to the matrix A of rank not greater than a specified rank k, where k is much smaller than m and n. Methods such as the Singular Value Decomposition (SVD) may be used to find an approximation to A which is the best in a well defined sense. These methods require memory and time which are superlinear in m and n; for many applications in which the data sets are very large this is prohibitive. Two simple and intuitive algorithms are presented which, when given an $m \times n$ matrix A, compute a description of a low-rank approximation D^* to A, and which are qualitatively faster than the SVD. Both algorithms have provable bounds for the error matrix $A - D^*$. For any matrix X, let $\|X\|_F$ and $\|X\|_2$ denote its Frobenius norm and its spectral norm, respectively. In the first algorithm, $c = O(1)$ columns of A are randomly chosen. If the $m \times c$ matrix C consists of those c columns of A (after appropriate rescaling) then it is shown that from $C^T C$ approximations to the top singular values and corresponding singular vectors may be computed. From the computed singular vectors a description D^* of the matrix A may be computed such that $\text{rank}(D^*) \le k$ and such that

$$\|A - D^*\|_\xi^2 \le \min_{D:\text{rank}(D)\le k} \|A - D\|_\xi^2 + poly(k, 1/c) \, \|A\|_F^2$$

holds with high probability for both $\xi = 2, F$. This algorithm may be implemented without storing the matrix A in Random Access Memory (RAM), provided it can make two passes over the matrix stored in external memory and use $O(m+n)$ additional RAM memory. The second algorithm is similar except that it further approximates the matrix C by randomly sampling $r = O(1)$ rows of C to form a $r \times c$ matrix W. Thus, it has additional error, but it can be implemented in three passes over the matrix using only constant additional RAM memory. To achieve an additional error (beyond the best rank k approximation) that is at most $\epsilon \|A\|_F^2$, both algorithms take time which is polynomial in k, $1/\epsilon$, and $\log(1/\delta)$, where $\delta > 0$ is a failure probability; the first takes time linear in $\max(m, n)$ and the second takes time independent of m and n. Our bounds improve previously published results with respect to the rank parameter k for both the Frobenius and spectral norms. In addition, the proofs for the error bounds use a novel method that makes important use of matrix perturbation theory. The probability distribution over columns of A and the rescaling are crucial features of the algorithms which must be chosen judiciously.

[*] Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180, drinep@cs.rpi.edu

[†] Department of Computer Science, Yale University, New Haven, Connecticut, USA 06520, kannan@cs.yale.edu

[‡] Supported in part by a grant from the NSF.

[§] Department of Mathematics, Yale University, New Haven, Connecticut, USA 06520, mahoney@cs.yale.edu

1 Introduction

We are interested in developing and analyzing fast Monte Carlo algorithms for performing useful computations on large matrices. In this paper we consider the Singular Value Decomposition (SVD); in two related papers we consider matrix multiplication and a new method for computing a compressed approximate decomposition of a large matrix [15, 16]. Since such computations generally require time which is superlinear in the number of nonzero elements of the matrix, we expect our algorithms to be useful in many applications where data sets are modeled by matrices and are extremely large. In all these cases, we assume that the input matrices are prohibitively large to store in Random Access Memory (RAM) and thus that only external memory storage is possible. Our algorithms will be allowed to read the matrices a few, e.g., one or two or three, times and keep a small randomly-chosen and rapidly-computable "sketch" of the matrices in RAM; computations will then be performed on this "sketch". We will work within the framework of the Pass-Efficient computational model, in which the scarce computational resources are the number of passes over the data, the additional RAM space required, and the additional time required [15, 13].

In many applications, the data consist of (or may be naturally formulated as) an $m \times n$ matrix A which is either low-rank or is well approximated by a low-rank matrix [7, 9, 26, 3, 27, 29, 30, 25, 23]. In these application areas, e.g., latent semantic indexing, DNA microarray analysis, facial and object recognition, and web search models, the data may consist of n points $a_i \in \mathbb{R}^m$, $i = 1, \ldots, n$. Let $A \in \mathbb{R}^{m \times n}$ be the matrix with columns a_i. Two methods for dealing with such high-dimensional data are the SVD (and the related Principle Component Analysis) and Multidimensional Scaling [19, 24]. Thus, it is often of interest to find a low-rank approximation to A, i.e., an approximation D, of rank no greater than a specified rank k, to the matrix A, where k is much smaller than m and n. For example, this rank reduction is used in many applications of linear algebra and statistics as well as in image processing, lossy data compression, text analysis and cryptography [6]. The SVD may be used to find an approximation to A which is the best in a well defined sense [19, 20] but it requires a superlinear (in m and n) polynomial time dependence that is prohibitive for many applications in which the data sets are very large. Another method that has attracted interest recently is the traditional "Random Projection" method where one projects the problem into a randomly chosen low-dimensional subspace [22, 31, 21]. This dimensional reduction requires performing an operation that amounts to premultiplying the given $m \times n$ matrix A by an $s \times m$ matrix which takes time dependent in a superlinear manner on $m + n$.

In this paper we present two simple and intuitive algorithms which, when given an $m \times n$ matrix A, compute a description of a low-rank approximation D^* to A, and which are qualitatively faster than the SVD. Both algorithms have provable bounds for the error matrix $A - D^*$. For any matrix X, let $\|X\|_F$ and $\|X\|_2$ denote its Frobenius norm and its spectral norm (as defined in Section 3.1), respectively. In the first algorithm, the LinearTimeSVD algorithm of Section 4, $c = O(1)$ columns of A are randomly chosen. If the $m \times c$ matrix C consists of those c columns of A (after appropriate rescaling) then it is shown that from $C^T C$ approximations to the top singular values and corresponding singular vectors of A may be computed. From the computed singular vectors a description D^* of the matrix A may be computed such that $\text{rank}(D^*) \leq k$ and such that

$$\|A - D^*\|_\xi^2 \leq \min_{D:\text{rank}(D) \leq k} \|A - D\|_\xi^2 + poly(k, 1/c)\,\|A\|_F^2 \tag{1}$$

holds with high probability for each of $\xi = 2, F$. This algorithm may be implemented without storing the matrix A in RAM, provided it can make two passes over the matrix stored in external memory and use $O(m + n)$ additional RAM memory. The second algorithm, the Constant-

addl. error for:	LinearTimeSVD	ConstantTimeSVD	Ref. [18, 17]
$\|A - D^*\|_2^2$	$1/\epsilon^2$	$1/\epsilon^4$	k^4/ϵ^3
$\|A - D^*\|_F^2$	k/ϵ^2	k^2/ϵ^4	k^4/ϵ^3

Figure 1: Summary of sampling complexity

TimeSVD algorithm of Section 5, is similar except that it further approximates the matrix C by randomly sampling $r = O(1)$ rows of C to form a $r \times c$ matrix W. Thus, it has additional error but it can be implemented in three passes over the matrix using only constant additional RAM memory. To achieve an additional error that is at most $\epsilon \|A\|_F^2$, both algorithms take time which is polynomial in k, $1/\epsilon$, and $\log(1/\delta)$, where $\delta > 0$ is an failure probability; see Figure 1 for a summary of the dependence of the sampling complexity on k and ϵ. The first algorithm takes time linear in $\max(m, n)$ and the other takes time independent of m and n. Our bounds improve previously published results with respect to the rank parameter k for both the Frobenius and spectral norms. In addition, the proofs for the error bounds use a novel method that makes important use of matrix perturbation theory. The probability distribution over columns of A and the rescaling are crucial features of the algorithms which must be chosen judiciously.

It is worth emphasizing how this work fits into recent work on computing low-rank matrix approximations. In the original work of Frieze, Kannan, and Vempala [18] (see also [17]) it was shown that by working with a randomly-chosen and constant-sized submatrix of A one could obtain bounds of the form (1) for the Frobenius norm (and thus indirectly for the spectral norm). To achieve an additional error that is at most $\epsilon \|A\|_F^2$ the size of the submatrix was a constant with respect to m and n but depended polynomially on k and $1/\epsilon$; although the submatrix was constant-sized, its construction (in particular, the construction of the sampling probabilities) required space and thus time that was linear in $m + n$. In this work, we modify the algorithm of [18] so that both the construction of *and* the computation on the constant-sized submatrix requires only constant additional space and time; thus, it fits within the framework of the Pass-Efficient model of data-streaming computation [13, 15]. In addition, we provide a different proof of the main result of [18] for the Frobenius norm and improve the polynomial dependence on k. Our proof method is quite different than that of [18]; it relies heavily on the approximate matrix multiplication result of [15] and [12] and it uses the Hoffman-Wielandt inequality. In addition, we provide a proof of a direct and significantly improved bound with respect to the spectral norm. Since these results are technically quite complex, we also present the corresponding proofs for both norms in the linear additional space and time framework [13, 15]. These latter results have been presented in the context of clustering applications [11, 10], but are included here for completeness and to provide motivation and clarity for the more complex constant time results. Figure 1 provides a summary of our results, for both the linear and constant time models, and shows the number of rows and columns to be sampled sufficient to ensure, with high probability, an additional error of $\epsilon \|A\|_F^2$ in (1); see Section 6 for more discussion.

In other related work, Achlioptas and McSherry have also computed low-rank approximations using somewhat different sampling techniques [2, 1]. The primary focus of their work was in introducing methods to accelerate Orthogonal Iteration and Lanczos Iteration, which are two commonly used methods for computing low-rank approximations to a matrix. Also included in [2, 1] is a comparison of their methods with those of [11, 13, 18] and thus with the results we present here. Our algorithms and those of [18] and [2, 1] come with mathematically rigorous guarantees of the running time and of the quality of the approximation produced. As far as we know, so-called incremental SVD algorithms which bring as much of the data as possible into memory, compute the SVD, and then update this SVD in an incremental fashion with the remaining data, do not come with such guarantees.

In Section 2 several applications areas that deal with large matrices are discussed and in Section 3 we provide a review of relevant linear algebra, the Pass-Efficient model, and an approximate matrix multiplication result that will be used extensively. Then, in Section 4 our linear additional space and time approximation algorithm, the LinearTimeSVD algorithm, is presented and analyzed; in Section 5 our constant additional space and time approximation algorithm, the ConstantTimeSVD algorithm, is presented and analyzed. Finally, in Section 6 a discussion and conclusion are presented.

2 Some Applications

There are numerous applications in which the data are well approximated by a low-rank matrix. In this section we discuss several such applications to provide a motivation for our algorithms.

2.1 Latent Semantic Indexing

Latent semantic indexing is a general technique for analyzing a collection of documents which are assumed to be related [7, 9, 26]. Approaches to retrieving textual information from databases that depend on a lexical match between words in the query and words in the document can be inaccurate, both since often users want to retrieve information on the basis of conceptual content and since individual words do not in general provide reliable evidence about the conceptual topic of a document. Latent semantic indexing (LSI) is an alternative matching method that attempts to overcome problems associated with lexical matching; it does so by assuming that there is some underlying or latent semantic structure that is partially obscured by variability in word choice and then using techniques such as SVD to remove the noise and estimate this latent structure.

Suppose that there are m documents and n terms which occur in the documents. Latent semantic structure analysis starts with a term-document matrix, e.g., a matrix $A \in \mathbb{R}^{m \times n}$, where A_{ij} is frequency of the j-th term in the i-th document. A topic is modeled as an n-vector of non-negative reals summing to 1, where the j-th component of a topic vector is interpreted as the frequency with which the j-th term occurs in a discussion of the topic. By assumption, the number of topics that the documents are about is small relative to the number of unique terms n. It can be argued that, for a given k, finding a set of k topics which best describe the documents corresponds to keeping only the top k singular vectors of A; most of the important underlying structure in the association of terms and documents will then be kept and most of the noise or variability in word usage will be removed.

2.2 DNA Microarray Data

DNA microarray technology has been used to study a variety of biological processes since it permits the monitoring of the expression levels of thousands of genes under a range of experimental conditions [3, 27, 29]. Depending on the particular technology, either the absolute or the relative expression levels of n genes, which for model organisms may constitute nearly the entire genome, are probed simultaneously by a single microarray. A series of m arrays probe genome-wide expression levels in m different samples, i.e., under m different experimental conditions. The data from microarray experiments may thus be represented as a matrix $A \in \mathbb{R}^{m \times n}$, where A_{ij} represents the expression level of gene i under experimental condition j. From this matrix, both the relative expression level of the i-th gene under every condition and also the relative expression level of every gene under the j-th condition may be easily extracted.

This matrix is low-rank and thus a small number of eigengenes and corresponding eigenarrays (left and right singular vectors) are sufficient to capture most of the gene expression information. Removing the rest, which correspond to noise or experimental artifacts, enables meaningful comparison of the expression levels of different genes. When processing and modeling genome wide expression data, the SVD and its low-rank approximation provides a framework such that the mathematical variables and operations suggest assigned biological meaning, e.g., in terms of cellular regulatory processes and cellular states, that may be hidden in the original data due to experimental noise or hidden dependencies. Expression data has been used for inference tasks such as to identify genes based on co-expression, predict regulatory elements, and reverse engineer transcription networks, but this inference is difficult with noise or dependencies.

2.3 Eigenfaces and Facial Recognition

Applications of SVD and low-rank approximations in computer vision include pattern estimation, image compression and restoration and facial and object recognition, where the concept of eigenfaces has been useful [30, 25].

The goal of facial recognition is to recognize a certain face given a database of photographs of human faces under variations in lighting conditions and pose viewpoints. A common approach is to represent the database as a matrix in which the rows of the matrix are the images represented as vectors. Thus, if there are m images, each of which is of size $n \times n$, the matrix $A \in \mathbb{R}^{m \times n^2}$ represents the database of images, where A_{ij} is the j-th pixel value in the i-th image. Typically, $m \ll n^2$ and, although many of the singular vectors are needed for very accurate reconstruction of an image, often only a few of the singular vectors are needed to extract the major appearance characteristics of an image. The right singular vectors of the matrix A are known as eigenfaces since they are the principal components or eigenvectors of the associated correlation matrix of the set of face images. The eigenfaces are computed and they are used to project the database of photographs to a lower dimensional space that spans the significant variations among known facial images. Then, given a new image, it is projected in to the same low dimensional space and its position is then compared to the images in the database.

2.4 Web Search Model

The problem of how to extract information from the network structure of a hyperlinked environment such as the world wide web was considered by Kleinberg [23]. This is of interest, for example, if one wants to find web pages that are relevant to a given query and one is using a keyword-based web search program, since there is no obvious endogenous measure of an authoritative page that would favor it under a text-based ranking system.

Starting with a set of pages returned by a text-based search engine, a document is defined to be an authority if many other documents returned by the search point to it, i.e., have a hypertext link to it. A document is defined to be a hub if it points to many other documents. More generally, suppose n documents are returned by the search engine. Then, a matrix $A \in \mathbb{R}^{m \times n}$ is defined, where A_{ij} is 1 or 0 depending on whether the i-th document points to the j-th document. Kleinberg attempts to find two n-vectors, x and y, where x_i is the hub weight of document i and y_j is the authority weight of document j. He then argues that it is desirable to find $\max_{|x|=|y|=1} x^T A y$, where $| \cdot |$ denotes the Euclidean length, since in maximizing x, y one expects the hub weights and authority weights to be mutually consistent. This is simply the problem of finding the singular vectors of A. Since A is large, he judiciously chooses a submatrix of A and computes only the singular vectors of it. In the case when the key word has multiple meanings not only the top but some of the other singular vectors with large singular values are interesting. Thus, it is of interest to find the k largest singular vectors form some small k. This is the problem we are considering and we also find the singular vectors of a submatrix, but a randomly chosen one.

3 Review of Relevant Background

This section contains a review of linear algebra that will be useful throughout the paper; for more detail, see [19, 20, 28, 8] and references therein. This section also contains a review of the Pass-Efficient model of data-streaming computation (which provides a framework within which our SVD results may be viewed) and a matrix multiplication result that will be used extensively in our proofs; see [12, 13, 15] for more details.

3.1 Review of Linear Algebra

For a vector $x \in \mathbb{R}^n$ we let x_i, $i = 1, \ldots, n$, denote the i-th element of x and we let $|x| = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$. For a matrix $A \in \mathbb{R}^{m\times n}$ we let $A^{(j)}$, $j = 1, \ldots, n$, denote the j-th column of A as a column vector and $A_{(i)}$, $i = 1, \ldots, m$, denote the i-th row of A as a row vector; thus, if A_{ij} denotes the (i,j)-th element of A, $A_{ij} = \left(A^{(j)}\right)_i = \left(A_{(i)}\right)_j$. The range of an $A \in \mathbb{R}^{m\times n}$ is

$$\text{range}(A) = \{y \in \mathbb{R}^m : \ y = Ax \ \text{ for some } \ x \in \mathbb{R}^n\} = \textbf{span}\left(A^{(1)}, \ldots, A^{(n)}\right).$$

The rank of A, $\text{rank}(A)$, is the dimension of $\text{range}(A)$ and is equal to the number of linearly independent columns of A; since this is equal to $\text{rank}(A^T)$ it also equals the number of linearly independent rows of A. The null space of A is

$$\text{null}(A) = \{x \in \mathbb{R}^n : Ax = 0\}.$$

For a matrix $A \in \mathbb{R}^{m\times n}$ we denote matrix norms by $\|A\|_\xi$, using subscripts to distinguish between various norms. Of particular interest will be the Frobenius norm which is defined by:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}. \tag{2}$$

If $\textbf{Tr}\,(A)$ is the matrix trace which is the sum of the diagonal elements of A, then $\|A\|_F^2 = \textbf{Tr}\left(A^TA\right) = \textbf{Tr}\left(AA^T\right)$. Also of interest is the spectral norm which is defined by:

$$\|A\|_2 = \sup_{x\in\mathbb{R}^n,\ x\neq 0} \frac{|Ax|}{|x|}. \tag{3}$$

Both of these norms are submultiplicative and unitarily invariant and they are related to each other as:

$$\|A\|_2 \le \|A\|_F \le \sqrt{n}\,\|A\|_2\,.$$

Both of these norms provide a measure of the "size" of the matrix A. Note that if $A \in \mathbb{R}^{m\times n}$ then there exists an $x \in \mathbb{R}^n$ such that $|x| = 1$ and $A^TAx = \|A\|_2^2\,x$ and that if $\{x^1, x^2, \ldots, x^n\}$ is any basis of $\mathbb{R}^n$ and if $A \in \mathbb{R}^{m\times n}$, then $\|A\|_F^2 = \sum_{i=1}^{n} \left|Ax^i\right|^2$.

If $A \in \mathbb{R}^{m\times n}$, then there exist orthogonal matrices $U = [u^1u^2\ldots u^m] \in \mathbb{R}^{m\times m}$ and $V = [v^1v^2\ldots v^n] \in \mathbb{R}^{n\times n}$ where $\left\{u^t\right\}_{t=1}^{m} \in \mathbb{R}^m$ and $\left\{v^t\right\}_{t=1}^{n} \in \mathbb{R}^n$ are such that

$$U^TAV = \Sigma = \textbf{diag}(\sigma_1, \ldots, \sigma_\rho), \tag{4}$$

where $\Sigma \in \mathbb{R}^{m \times n}$, $\rho = \min\{m, n\}$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_\rho \geq 0$. Equivalently,

$$A = U \Sigma V^T.$$

The three matrices U, V, and Σ constitute the Singular Value Decomposition (SVD) of A. The σ_i are the singular values of A and the vectors u^i, v^i are the i-th left and the i-th right singular vectors, respectively. The columns of U and V satisfy the relations $Av^i = \sigma_i u^i$ and $A^T u^i = \sigma_i v^i$. For symmetric matrices the left and right singular vectors are the same. The singular values of A are the non-negative square roots of the eigenvalues of $A^T A$ and of AA^T; furthermore, the columns of U, i.e., the left singular vectors, are eigenvectors of AA^T and the columns of V, i.e., the right singular vectors, are eigenvectors of $A^T A$.

The SVD can reveal important information about the structure of a matrix. If we define r by $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_\rho = 0$, then $\text{rank}(A) = r$, $\text{null}(A) = \textbf{span}\left(v^{r+1}, \ldots, v^\rho\right)$, and $\text{range}(A) = \textbf{span}\left(u^1, \ldots, u^r\right)$. If we let $U_r \in \mathbb{R}^{m \times r}$ denote the matrix consisting of the first r columns of U, $V_r \in \mathbb{R}^{r \times n}$ denote the matrix consisting of the first r columns of V, and $\Sigma_r \in \mathbb{R}^{r \times r}$ denote the principal $r \times r$ sub-matrix of Σ, then

$$A = U_r \Sigma_r V_r^T = \sum_{t=1}^{r} \sigma_t u^t {v^t}^T. \tag{5}$$

Note that this dyadic decomposition property provides a canonical description of a matrix as a sum of r rank one matrices of decreasing importance. If $k \leq r$ and we define

$$A_k = U_k \Sigma_k V_k^T = \sum_{t=1}^{k} \sigma_t u^t {v^t}^T \tag{6}$$

then $A_k = U_k U_k^T A = \left(\sum_{t=1}^{k} u^t {u^t}^T\right) A$ and $A_k = A V_k V_k^T = A \left(\sum_{t=1}^{k} v^t {v^t}^T\right)$, i.e., A_k is the projection of A onto the space spanned by the top k singular vectors of A. Furthermore, the distance (as measured by both $\|\cdot\|_2$ and $\|\cdot\|_F$) between A and any rank k approximation to A is minimized by A_k, i.e.,

$$\min_{D \in \mathbb{R}^{m \times n} : \text{rank}(D) \leq k} \|A - D\|_2 = \|A - A_k\|_2 = \sigma_{k+1}(A) \tag{7}$$

and

$$\min_{D \in \mathbb{R}^{m \times n} : \text{rank}(D) \leq k} \|A - D\|_F^2 = \|A - A_k\|_F^2 = \sum_{t=k+1}^{r} \sigma_t^2(A). \tag{8}$$

Thus, A_k constructed from the k largest singular triplets of A is the optimal rank k approximation to A with respect to both $\|\cdot\|_F$ and $\|\cdot\|_2$. More generally, one can also show that $\|A\|_2 = \sigma_1$ and that $\|A\|_F^2 = \sum_{i=1}^{r} \sigma_i^2$.

From the perturbation theory of matrices it is known that the size of the difference between two matrices can be used to bound the difference between the singular value spectrum of the two matrices [28, 8]. In particular, if $A, E \in \mathbb{R}^{m \times n}, m \geq n$, then:

$$\max_{t : 1 \leq t \leq n} |\sigma_t(A + E) - \sigma_t(A)| \leq \|E\|_2 \tag{9}$$

and

$$\sum_{k=1}^{n} (\sigma_k(A + E) - \sigma_k(A))^2 \leq \|E\|_F^2. \tag{10}$$

The latter inequality is known as the Hoffman-Wielandt inequality.

3.2 Review of the Pass-Efficient Model

The Pass-Efficient model of data-streaming computation is a computational model that is motivated by the observation that in modern computers the amount of disk storage, i.e., sequential access memory, has increased very rapidly, while RAM and computing speeds have increased at a substantially slower pace [15, 13]. In the Pass-Efficient model the three scarce computational resources are number of passes over the data and the additional RAM space and additional time required by the algorithm. The data are assumed to be stored on a disk, to consist of elements whose size is bounded by a constant, and to be presented to an algorithm on a read-only tape. See [15] for more details.

3.3 Review of Matrix Multiplication

The BasicMatrixMultiplication algorithm to approximate the product of two matrices is presented and analyzed in [15]. When this algorithm is given as input two matrices, $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$, a probability distribution $\{p_i\}_{i=1}^{n}$, and a number $c \leq n$, it returns as output two matrices, C and R, such that $CR \approx AB$; $C \in \mathbb{R}^{m\times c}$ is a matrix whose columns are c randomly chosen columns of A (suitably rescaled) and $R \in \mathbb{R}^{c\times p}$ is a matrix whose rows are the c corresponding rows of B (also suitably rescaled). An important aspect of this algorithm is the probability distribution $\{p_i\}_{i=1}^{n}$ used to choose column-row pairs. Although one could always use a uniform distribution, superior results are obtained if the probabilities are chosen judiciously. In particular, a set of sampling probabilities $\{p_i\}_{i=1}^{n}$ are *nearly optimal probabilities* if they are of the form (11) and are the *optimal probabilities* (with respect to approximating the product AB) if they are of the form (11) with $\beta = 1$. In [15] we prove the following theorem.

Theorem 1 *Suppose $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ such that $1 \leq c \leq n$, and $\{p_i\}_{i=1}^{n}$ are such that $p_i \geq 0$, $\sum_{i=1}^{n} p_i = 1$ and such that for some positive constant $\beta \leq 1$*

$$p_k \geq \frac{\beta \left|A^{(k)}\right| \left|B_{(k)}\right|}{\sum_{k'=1}^{n} \left|A^{(k')}\right| \left|B_{(k')}\right|}. \tag{11}$$

Construct C and R with the BasicMatrixMultiplication *algorithm of [15] and let CR be an approximation to AB. Then,*

$$\mathbf{E}\left[\|AB - CR\|_F^2\right] \leq \frac{1}{\beta c} \|A\|_F^2 \, \|B\|_F^2 . \tag{12}$$

Furthermore, let $\delta \in (0,1)$ and $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. Then, with probability at least $1 - \delta$,

$$\|AB - CR\|_F^2 \leq \frac{\eta^2}{\beta c} \|A\|_F^2 \, \|B\|_F^2 . \tag{13}$$

In [15] it is shown that after one pass over the matrices nearly optimal probabilities can be constructed. In the present paper, we will be particularly interested in the case that $B = A^T$. In this case, using the Select algorithm of [15] random samples can be drawn according to nearly optimal probabilities using $O(1)$ additional space and time.

4 Linear Time SVD Approximation Algorithm

4.1 The Algorithm

Given a matrix $A \in \mathbb{R}^{m\times n}$ we wish to approximate its top k singular values and the corresponding singular vectors in a constant number of passes through the data and additional space and time

```
LinearTimeSVD Algorithm
Input: $A \in \mathbb{R}^{m \times n}$, $c, k \in \mathbb{Z}^+$ s.t. $1 \le k \le c \le n$, $\{p_i\}_{i=1}^n$ s.t. $p_i \ge 0$ and $\sum_{i=1}^n p_i = 1$.
Output: $H_k \in \mathbb{R}^{m \times k}$ and $\sigma_t(C), t = 1, \ldots, k$.

• For $t = 1$ to $c$,
    – Pick $i_t \in 1, \ldots, n$ with $\mathbf{Pr}[i_t = \alpha] = p_\alpha$, $\alpha = 1, \ldots, n$.
    – Set $C^{(t)} = A^{(i_t)} / \sqrt{c p_{i_t}}$.
• Compute $C^T C$ and its singular value decomposition; say $C^T C = \sum_{t=1}^c \sigma_t^2(C) y^t {y^t}^T$.
• Compute $h^t = C y^t / \sigma_t(C)$ for $t = 1, \ldots, k$.
• Return $H_k$, where $H_k^{(t)} = h^t$, and $\sigma_t(C), t = 1, \ldots, k$.
```

Figure 2: The LinearTimeSVD Algorithm

$$\begin{tikzcd}
\mathbb{R}^n \arrow[r, "\Sigma_A"] \arrow[d, "V"', "\{v^i\}" near end] & \mathbb{R}^m \arrow[d, "U", "\{u^i\}"' near end] & \\
\mathbb{R}^n \arrow[r, "A"] & \mathbb{R}^m & \mathbb{R}^m \arrow[l, "H(=U_C)"', "\{h^i\}" near end] \\
& \mathbb{R}^c \arrow[u, "C", "\{h^i\}"' near end] \arrow[loop left, "C^T C"] & \mathbb{R}^c \arrow[u, "\Sigma_C"'] \arrow[l, "Y(=V_C)", "\{y^i\}"' near end]
\end{tikzcd}$$

Figure 3: Diagram for the LinearTimeSVD Algorithm

that is $O(m+n)$. The strategy behind the LinearTimeSVD algorithm is to pick c columns of the matrix A, rescale each by an appropriate factor to form a matrix $C \in \mathbb{R}^{m \times c}$, and then compute the singular values and corresponding left singular vectors of the matrix C, which will be approximations to the singular values and left singular vectors of A, in a sense we make precise later. These are calculated by performing an SVD of the matrix $C^T C$ to compute the right singular vectors of C and from them calculating the left singular vectors of C.

The LinearTimeSVD algorithm is described in Figure 2; it takes as input a matrix A and returns as output an approximation to the top k left singular values and the corresponding singular vectors. Note that by construction the SVD of C is $C = H \Sigma_C Y^T$. A diagram illustrating the action of the LinearTimeSVD algorithm is presented in Figure 3. The transformation represented by the matrix A is shown along with its SVD and the transformation represented by the matrix C is also shown along with its SVD. It will be shown that, if the probabilities $\{p_i\}_{i=1}^n$ are chosen judiciously, then the left singular vectors of C are with high probability approximations to the left singular vectors of A.

In Section 4.2 we will be show that this algorithm takes $O(m+n)$, i.e., linear, additional space and time and in Section 4.3 we will prove the correctness of the algorithm.

4.2 Analysis of the Implementation and Running Time

Assuming that nearly optimal sampling probabilities (as defined in Section 3.3) are used, then in the LinearTimeSVD algorithm the sampling probabilities p_k can be used to select columns to be sampled in one pass and $O(c)$ additional space and time using the Select algorithm of [15]. Given the elements to be sampled, the matrix C can then be constructed in one additional pass; this requires additional space and time that is $O(mc)$. Given $C \in \mathbb{R}^{m\times c}$ computing C^TC requires $O(mc^2)$ additional space and time and computing the SVD of C^TC requires $O(c^3)$ additional space and time. Then, computing H_k requires k matrix-vector multiplications for a total of $O(mck)$ additional space and time. Thus, since c and k are assumed to be a constant, overall $O(m+n)$ additional space and time are required by the LinearTimeSVD algorithm. Note that the "description" of the solution that is computable in the allotted additional space and time is the explicit approximation to the top k singular values and corresponding left singular vectors.

4.3 Analysis of the Sampling Step

Approximating A by $A_k = U_kU_k^TA$ incurs an error equal to $\|A - A_k\|_F^2 = \sum_{t=k+1}^{r} \sigma_t^2(A)$ and $\|A - A_k\|_2 = \sigma_{k+1}(A)$, since A_k is the "optimal" rank k approximation to A with respect to both $\|\cdot\|_F$ and $\|\cdot\|_2$. We will show that in addition to this error the matrix $H_kH_k^TA$ has an error that depends on $\left\|AA^T - CC^T\right\|_F$. Then, using the results of Theorem 1 we will show that this additional error depends on $\|A\|_F^2$. We first consider obtaining a bound with respect to the Frobenius norm.

Theorem 2 *Suppose $A \in \mathbb{R}^{m\times n}$ and let H_k be constructed from the* LinearTimeSVD *algorithm. Then,*

$$\left\|A - H_kH_k^TA\right\|_F^2 \leq \|A - A_k\|_F^2 + 2\sqrt{k}\left\|AA^T - CC^T\right\|_F .$$

Proof: Recall that for matrices X and Y, $\|X\|_F^2 = \mathbf{Tr}(X^TX)$, $\mathbf{Tr}\,(X+Y) = \mathbf{Tr}\,(X) + \mathbf{Tr}\,(Y)$, and also that $H_k^TH_k = I_k$. Thus, we may express $\left\|A - H_kH_k^TA\right\|_F^2$ as:

$$\begin{aligned} \left\|A - H_kH_k^TA\right\|_F^2 &= \mathbf{Tr}\left((A - H_kH_k^TA)^T(A - H_kH_k^TA)\right) \\ &= \mathbf{Tr}\left(A^TA - 2A^TH_kH_k^TA + A^TH_kH_k^TH_kH_k^TA\right) \\ &= \mathbf{Tr}\left(A^TA\right) - \mathbf{Tr}\left(A^TH_kH_k^TA\right) \\ &= \|A\|_F^2 - \left\|A^TH_k\right\|_F^2 . \end{aligned} \tag{14}$$

We may relate $\left\|A^TH^k\right\|_F^2$ and $\sum_{t=1}^{k}\sigma_t^2(C)$ by the following:

$$\begin{aligned} \left| \left\|A^TH_k\right\|_F^2 - \sum_{t=1}^{k}\sigma_t^2(C) \right| &\leq \sqrt{k}\left(\sum_{t=1}^{k}\left(\left|A^Th^t\right|^2 - \sigma_t^2(C)\right)^2\right)^{1/2} \\ &= \sqrt{k}\left(\sum_{t=1}^{k}\left(\left|A^Th^t\right|^2 - \left|C^Th^t\right|^2\right)^2\right)^{1/2} \\ &= \sqrt{k}\left(\sum_{t=1}^{k}\left(h^{t^T}(AA^T - CC^T)h^t\right)^2\right)^{1/2} \\ &\leq \sqrt{k}\left\|AA^T - CC^T\right\|_F . \end{aligned} \tag{15}$$

The first inequality follows by applying the Cauchy-Schwartz inequality; the last inequality follows by writing AA^T and CC^T with respect to a basis containing $\{h^t\}_{t=1}^k$. By again applying the Cauchy-Schwartz inequality, noting that $\sigma_t^2(X) = \sigma_t(XX^T)$ for a matrix X, and applying the Hoffman-Wielandt inequality, (10), we may also relate $\sum_{k=1}^k \sigma_t^2(C)$ and $\sum_{k=1}^k \sigma_t^2(A)$ by the following:

$$\begin{aligned}
\left|\sum_{t=1}^k \sigma_t^2(C) - \sum_{t=1}^k \sigma_t^2(A)\right| &\le \sqrt{k}\left(\sum_{t=1}^k \left(\sigma_t^2(C) - \sigma_t^2(A)\right)^2\right)^{1/2} \\
&= \sqrt{k}\left(\sum_{t=1}^k \left(\sigma_t(CC^T) - \sigma_t(AA^T)\right)^2\right)^{1/2} \\
&\le \sqrt{k}\left(\sum_{t=1}^m \left(\sigma_t(CC^T) - \sigma_t(AA^T)\right)^2\right)^{1/2} \\
&\le \sqrt{k}\left\|CC^T - AA^T\right\|_F . \qquad (16)
\end{aligned}$$

Combining the results of (15) and (16) allows us to relate $\left\|A^T H_k\right\|_F^2$ and $\sum_{t=1}^k \sigma_t^2(A)$ by the following:

$$\left|\left\|A^T H_k\right\|_F^2 - \sum_{t=1}^k \sigma_t^2(A)\right| \le 2\sqrt{k}\left\|AA^T - CC^T\right\|_F . \qquad (17)$$

Combining (17) with (14) yields the theorem.

◇

We next prove a similar result for the spectral norm; note that the factor $\sqrt{k}$ is not present.

Theorem 3 *Suppose $A \in \mathbb{R}^{m\times n}$ and let H_k be constructed from the* LinearTimeSVD *algorithm. Then,*

$$\left\|A - H_k H_k^T A\right\|_2^2 \le \left\|A - A_k\right\|_2^2 + 2\left\|AA^T - CC^T\right\|_2 .$$

Proof: Let $\mathcal{H}_k = \text{range}(H_k) = \textbf{span}\left(h^1, ..., h^k\right)$ and $\mathcal{H}_{m-k}$ be the orthogonal complement of $\mathcal{H}_k$. Let $x \in \mathbb{R}^m$ and let $x = \alpha y + \beta z$ where $y \in \mathcal{H}_k$, $z \in \mathcal{H}_{m-k}$, and $\alpha^2 + \beta^2 = 1$; then,

$$\begin{aligned}
\left\|A - H_k H_k^T A\right\|_2 &= \max_{x\in\mathbb{R}^m, |x|=1} \left|x^T(A - H_k H_k^T A)\right| \\
&= \max_{y\in\mathcal{H}_k, |y|=1, z\in\mathcal{H}_{m-k}, |z|=1, \alpha^2+\beta^2=1} \left|(\alpha y^T + \beta z^T)(A - H_k H_k^T A)\right| \\
&\le \max_{y\in\mathcal{H}_k, |y|=1} \left|y^T(A - H_k H_k^T A)\right| + \max_{z\in\mathcal{H}_{m-k}, |z|=1} \left|z^T(A - H_k H_k^T A)\right| \qquad (18) \\
&= \max_{z\in\mathcal{H}_{m-k}, |z|=1} \left|z^T A\right| . \qquad (19)
\end{aligned}$$

(18) follows since $\alpha, \beta \le 1$ and (19) follows since $y \in \mathcal{H}_k$ and $z \in \mathcal{H}_{m-k}$. We next bound (19):

$$\begin{aligned}
\left|z^T A\right|^2 &= z^T CC^T z + z^T\left(AA^T - CC^T\right) z \\
&\le \sigma_{k+1}^2(C) + \left\|AA^T - CC^T\right\|_2 \qquad (20) \\
&\le \sigma_{k+1}^2(A) + 2\left\|AA^T - CC^T\right\|_2 \qquad (21) \\
&\le \left\|A - A_k\right\|_2^2 + 2\left\|AA^T - CC^T\right\|_2 . \qquad (22)
\end{aligned}$$

(20) follows since $\max_{z\in\mathcal{H}_{m-k}} \left|z^T C\right|$ occurs when z is the $(k+1)$-st left singular vector, i.e., the maximum possible in the $\mathcal{H}_{m-k}$ subspace. (21) follows since $\sigma_{k+1}^2(C) = \sigma_{k+1}(CC^T)$ and since

by (9) we have that $\sigma_{k+1}^2(C) \leq \sigma_{k+1}(AA^T) + \left\|AA^T - CC^T\right\|_2$; (22) follows since $\|A - A_k\|_2 = \sigma_{k+1}(A)$. The theorem then follows by combining (19) and (22).

◇

Theorem 2 and Theorem 3 hold regardless of the sampling probabilities $\{p_i\}_{i=1}^n$. Since $\|A - A_k\|_\xi$, $\xi = 2, F$, is a property of the matrix A, the choice of sampling probabilities enters into the error of $\|A - H_k H_k^T A\|_\xi^2$ only through the term involving the additional error beyond the optimal rank k approximation, i.e., the term $\|AA^T - CC^T\|_\xi$. Although the additional error in Theorem 3 depends on $\left\|AA^T - CC^T\right\|_2$, we note that $\left\|AA^T - CC^T\right\|_2 \leq \left\|AA^T - CC^T\right\|_F$ and will use a bound for the latter quantity to bound the former in the following. Note that the prefactor of the additional error is $2\sqrt{k}$ for $\|\cdot\|_F^2$ while that for $\|\cdot\|_2^2$ is only 2.

In the following theorem we specialize the sampling probabilities to be those that are nearly optimal; by choosing enough columns, the error in the approximation of the SVD can be made arbitrarily small.

Theorem 4 *Suppose $A \in \mathbb{R}^{m\times n}$, let H_k be constructed from the* LinearTimeSVD *algorithm by sampling c columns of A with probabilities $\{p_i\}_{i=1}^n$ such that $p_i \geq \beta \left|A^{(i)}\right|^2 / \|A\|_F^2$ for some positive $\beta \leq 1$, and let $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. Let $\epsilon > 0$. If $c \geq 4k/\beta\epsilon^2$, then*

$$\mathbf{E}\left[\left\|A - H_k H_k^T A\right\|_F^2\right] \leq \|A - A_k\|_F^2 + \epsilon \|A\|_F^2 , \tag{23}$$

and if $c \geq 4k\eta^2/\beta\epsilon^2$ then with probability at least $1 - \delta$

$$\left\|A - H_k H_k^T A\right\|_F^2 \leq \|A - A_k\|_F^2 + \epsilon \|A\|_F^2 . \tag{24}$$

In addition, if $c \geq 4/\beta\epsilon^2$, then

$$\mathbf{E}\left[\left\|A - H_k H_k^T A\right\|_2^2\right] \leq \|A - A_k\|_2^2 + \epsilon \|A\|_F^2 , \tag{25}$$

and if $c \geq 4\eta^2/\beta\epsilon^2$ then with probability at least $1 - \delta$

$$\left\|A - H_k H_k^T A\right\|_2^2 \leq \|A - A_k\|_2^2 + \epsilon \|A\|_F^2 . \tag{26}$$

Proof: By combining Theorems 2 and 3 with Theorem 1 we have that

$$\mathbf{E}\left[\left\|A - H_k H_k^T A\right\|_F^2\right] \leq \|A - A_k\|_F^2 + \left(\frac{4k}{\beta c}\right)^{1/2} \|A\|_F^2 \tag{27}$$

$$\mathbf{E}\left[\left\|A - H_k H_k^T A\right\|_2^2\right] \leq \|A - A_k\|_2^2 + \left(\frac{4}{\beta c}\right)^{1/2} \|A\|_F^2 , \tag{28}$$

and that with probability at least $1 - \delta$,

$$\left\|A - H_k H_k^T A\right\|_F^2 \leq \|A - A_k\|_F^2 + \left(\frac{4\eta^2 k}{\beta c}\right)^{1/2} \|A\|_F^2 \tag{29}$$

$$\left\|A - H_k H_k^T A\right\|_2^2 \leq \|A - A_k\|_2^2 + \left(\frac{4\eta^2}{\beta c}\right)^{1/2} \|A\|_F^2 . \tag{30}$$

The theorem follows by using the appropriate value of c.

◇

Note that alternatively one could sample rows instead of columns of a matrix; in this case, a modified version of the LinearTimeSVD algorithm leads to results analogous to Theorem 2 through Theorem 4.

5 Constant Time SVD Approximation Algorithm

5.1 The Algorithm

Given a matrix $A \in \mathbb{R}^{m \times n}$ we now wish to approximate its top k singular values and the corresponding singular vectors in a constant number of passes through the data and additional space and time that are $O(1)$, independent of m and n. The strategy behind the ConstantTimeSVD algorithm is to pick c columns of the matrix A, rescale each by an appropriate factor to form a matrix $C \in \mathbb{R}^{m \times c}$, and then compute approximations to the singular values and left singular vectors of the matrix C which will then be approximations to the singular values and left singular vectors of A. In the LinearTimeSVD algorithm of Section 4, the left singular vectors of the matrix C are computed exactly; as the analysis of Section 4.2 showed, this computation takes $O(m+n)$ additional space and time. With the ConstantTimeSVD algorithm, in order to use only $O(1)$ additional space and time, sampling is performed again, drawing rows of C to construct a matrix $W \in \mathbb{R}^{w \times c}$. The SVD of W^TW is then computed; let $W^TW = Z\Sigma_{W^TW}Z^T = Z\Sigma_W^2 Z^T$. The singular values and corresponding singular vectors so obtained are with high probability approximations to the singular values and singular vectors of C^TC and thus to the singular values and right singular vectors of C.

The ConstantTimeSVD algorithm is described in Figure 4; it takes as input a matrix A and returns as output a "description" of an approximation to the top k left singular values and the corresponding singular vectors. This "description" of the approximations to the left singular vectors of A may, at the expense of one additional pass and linear additional space and time, be converted into an explicit approximation to the left singular vectors of A by using $C = \tilde{H}\Sigma_W Z^T$ to compute $\tilde{H}$, whose columns are approximations the left singular vectors of C. Note that γ in the ConstantTimeSVD algorithm is introduced to bound small singular values of C that may be perturbed by the second level of sampling; as indicated, the particular value of γ that is chosen depends on the norm bound which is desired. Note also that the probabilities $\{q_j\}_{j=1}^m$ used in the algorithm are optimal (in the sense of Section 3.3), as will be the probabilities $\{p_i\}_{i=1}^n$ which will enter into Theorem 5.

A diagram illustrating the action of the ConstantTimeSVD algorithm is presented in Figure 5. The transformation represented by the matrix A is represented along with its SVD and the transformation represented by the matrix C is also shown (but note that its SVD is not shown). The transformation represented by the matrix W, which is constructed from C with the second level of sampling, is also shown along with its SVD. In addition, approximations to the right singular vectors of C and to the left singular vectors of C calculated from $C = \tilde{H}\Sigma_W Z^T$ are shown.

In Section 5.2 we will show that this algorithm takes $O(1)$ additional space and time. In Section 5.3 we will state Theorem 5, which will establish the correctness of the algorithm; this theorem is the main result of this section and is the analogue of Theorem 4. Finally, in Section 5.4 we will prove Theorem 5.

5.2 Analysis of the Implementation and Running Time

Assuming that optimal sampling probabilities (as defined in Section 3.3) are used, then in the ConstantTimeSVD algorithm the sampling probabilities p_k can be used to select columns to be sampled in one pass and $O(c)$ additional space and time using the Select algorithm of [15]. Given the columns of A to be sampled, we do not explicitly construct the matrix C but instead perform a second level of sampling and select w rows of C with probabilities $\{q_i\}_{i=1}^m$ (as described in the ConstantTimeSVD algorithm) in order to construct the matrix W. We do this by performing a

```
ConstantTimeSVD Algorithm

Input: $A \in \mathbb{R}^{m\times n}$, $c, w, k \in \mathbb{Z}^+$ s.t. $1 \le w \le m$, $1 \le c \le n$, and $1 \le k \le \min(w,c)$, and $\{p_i\}_{i=1}^n$ s.t. $p_i \ge 0$ and $\sum_{i=1}^n p_i = 1$.

Output: $\sigma_t(W), t = 1,\dots,\ell$ and a "description" of $\tilde{H}_\ell \in \mathbb{R}^{m\times\ell}$.

• For $t = 1$ to $c$,
    – Pick $i_t \in 1,\dots,n$ with $\mathbf{Pr}[i_t = \alpha] = p_\alpha$, $\alpha = 1,\dots,n$ and save $\{(i_t, p_{j_t}) : t = 1,\dots,c\}$.
    – Set $C^{(t)} = A^{(i_t)}/\sqrt{cp_{i_t}}$. (Note that $C$ is not explicitly constructed in RAM.)
• Choose $\{q_j\}_{j=1}^m$ s.t. $q_j = \left|C_{(j)}\right|^2 / \|C\|_F^2$.
• For $t = 1$ to $w$,
    – Pick $j_t \in 1,\dots,m$ with $\mathbf{Pr}[j_t = \alpha] = q_\alpha$, $\alpha = 1,\dots,m$.
    – Set $W_{(t)} = C_{(j_t)}/\sqrt{wq_{j_t}}$.
• Compute $W^TW$ and its singular value decomposition. Say $W^TW = \sum_{t=1}^c \sigma_t^2(W) z^t {z^t}^T$.
• If a $\|\cdot\|_F$ bound is desired, set $\gamma = \epsilon/100k$,
  Else if a $\|\cdot\|_2$ bound is desired, set $\gamma = \epsilon/100$.
• Let $\ell = \min\{k, \max\{t : \sigma_t^2(W) \ge \gamma \|W\|_F^2\}\}$.
• Return singular values $\{\sigma_t(W)\}_{t=1}^\ell$ and their corresponding singular vectors $\{z^t\}_{t=1}^\ell$.
```

Figure 4: The ConstantTimeSVD Algorithm

$$\begin{tikzcd}
\mathbb{R}^n \arrow[r, "\Sigma_A"] \arrow[d, "V"', "\{v^i\}"] & \mathbb{R}^m \arrow[d, "U", "\{u^i\}"'] & \\
\mathbb{R}^n \arrow[r, "A"] & \mathbb{R}^m & \mathbb{R}^m \arrow[l, "\tilde{H}"', "\{\tilde{h}^i\}"] \\
& \mathbb{R}^c \arrow[u, "C"] \arrow[loop left, "W^TW"] \arrow[dl, "W"'] & \mathbb{R}^c \arrow[u, "\Sigma_W"'] \arrow[l, "Z(=V_W)", "\{z^i\}"'] \arrow[dl, "\Sigma_W"] \\
\mathbb{R}^w & \mathbb{R}^w \arrow[l, "U_W"] &
\end{tikzcd}$$

Figure 5: Diagram for the ConstantTimeSVD Algorithm

second pass and using $O(w)$ additional space and time, again using the SELECT algorithm. Then in a third pass we explicitly construct W; this requires additional space and time that is $O(cw)$. Then, given W, computing W^TW requires $O(cw^2)$ additional space and time, and computing the SVD of W^TW requires $O(c^3)$ additional space and time. The singular values and corresponding singular vectors thus computed can then be returned as the "description" of the solution, and the total time for the CONSTANTTIMESVD algorithm is then $O(1)$ since c and w are assumed to be a constant. To explicitly compute $\tilde{H}_k$ would require k matrix-vector multiplications which would require another pass over the data and $O(mck) = O(m)$ additional space and time.

5.3 Statement of Theorem 5

This subsection and the next provide an analysis of the CONSTANTTIMESVD algorithm similar to the analysis of the LINEARTIMESVD algorithm found in Section 4.3. Recall that in Section 4 we were interested in bounding $\left\|A - H_kH_k^TA\right\|_\xi^2$, where $\xi = F, 2$. In that case, $H_k^TH_k = I_k$, $H_kH_k^T$ was an orthonormal projection, and $H_kH_k^TA$ was our rank at most k approximation to A. In the constant time model, we do not have access to H_k but instead to $\tilde{H}_\ell$, where the columns of $\tilde{H}_\ell$, i.e. $\tilde{h}^t = Cz^t/\sigma_t(W), t = 1, \ldots, \ell$, do *not* form an orthonormal set. However, by Lemma 2 of Section 5.4.1, if C and W are constructed by sampling with optimal probabilities, then with high probability the columns of $\tilde{H}_\ell$ are approximately orthonormal, $\tilde{H}_\ell^T\tilde{H}_\ell \approx I_\ell$, and $\tilde{H}_\ell\tilde{H}_\ell^T = \sum_{t=1}^{\ell} \tilde{h}^t\tilde{h}^{t^T}$ is approximately an orthonormal projection. Applying this to A, we will get our low-rank approximation. Note that in dealing with this nonorthonormality the original proof of [18] contained a small error which was corrected in the journal version [17].

In this section and the next we use the following notation. Recall that the SVD of $W^TW \in \mathbb{R}^{c\times c}$ is

$$W^TW = \sum_{t=1}^{c} \sigma_t^2(W)z^tz^{t^T} = Z\Sigma_W^2Z^T, \tag{31}$$

where $Z \in \mathbb{R}^{c\times c}$. Define $Z_{\alpha,\beta} \in \mathbb{R}^{c\times(\beta-\alpha+1)}$ to be the matrix whose columns are the α-th through the β-th singular vectors of W^TW. Then,

$$\tilde{H}_\ell = CZ_{1,\ell}T, \tag{32}$$

where $T \in \mathbb{R}^{\ell\times\ell}$ is the diagonal matrix with elements $T_{tt} = 1/\sigma_t(W)$. In addition, let the SVD of $\tilde{H}_\ell$ be

$$\tilde{H}_\ell = B_\ell\Sigma_{\tilde{H}_\ell}D_\ell^T, \tag{33}$$

and let us define the matrix $\Delta \in \mathbb{R}^{\ell\times\ell}$ to be

$$\Delta = TZ_{1,\ell}^T(C^TC - W^TW)Z_{1,\ell}T. \tag{34}$$

We will see that Δ is a measure of the degree to which the columns of $\tilde{H}_\ell$ are not orthonormal.

Theorem 5 is the constant time analogue of Theorem 4 and is the main result of this section. Note that since the results from sampling at the second step, i.e. sampling from the matrix C to form the matrix W, depend on the samples chosen in the first sampling step, we do not state the following results in expectation, but instead state them with high probability.

Theorem 5 *Suppose $A \in \mathbb{R}^{m\times n}$, let a description of $\tilde{H}_\ell$ be constructed from the* CONSTANTTIMESVD *algorithm by sampling c columns of A with probabilities $\{p_i\}_{i=1}^n$ and w rows of C with probabilities $\{q_j\}_{j=1}^m$ where $p_i = \left|A^{(i)}\right|^2 / \|A\|_F^2$ and $q_j = \left|C_{(j)}\right|^2 / \|C\|_F^2$. Let $\eta = 1 + \sqrt{8\log(2/\delta)}$ and $\epsilon > 0$.*

If a Frobenius norm bound is desired, and hence the ConstantTimeSVD *algorithm is run with $\gamma = \epsilon/100k$, then by choosing $c = \Omega(k^2\eta^2/\epsilon^4)$ columns of A and $w = \Omega(k^2\eta^2/\epsilon^4)$ rows of C we have that with probability at least $1-\delta$,*

$$\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2 \le \|A - A_k\|_F^2 + \epsilon \|A\|_F^2 . \tag{35}$$

If a spectral norm bound is desired, and hence the ConstantTimeSVD *algorithm is run with $\gamma = \epsilon/100$, then by choosing $c = \Omega(\eta^2/\epsilon^4)$ columns of A and $w = \Omega(\eta^2/\epsilon^4)$ rows of C we have that with probability at least $1-\delta$,*

$$\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_2^2 \le \|A - A_k\|_2^2 + \epsilon \|A\|_F^2 . \tag{36}$$

Proof: See Section 5.4.

◇

Recall that in Section 4 we first proved Theorems 2 and 3, which provided a bound on $\left\|A - H_k H_k^T A\right\|_F^2$ and $\left\|A - H_k H_k^T A\right\|_2^2$, respectively, for arbitrary probabilities, and then we proved Theorem 4 for the nearly optimal probabilities. Although a similar presentation strategy could be adopted in this section, in the interests of simplicity (due to the technically more complicated proofs in the constant time model) we instead immediately restrict ourselves in Theorem 5 to the case of optimal sampling probabilities and we defer the proofs of the supporting lemmas to Section 5.4.

5.4 Proof of Theorem 5

In this section, we prove Theorem 5. We start in Section 5.4.1 with several lemmas that are common to both the Frobenius and spectral norms. Then, in Section 5.4.2 we provide the proof of (35). Finally, in Section 5.4.3 we provide the proof of (36).

5.4.1 General Lemmas

In this section, we prove four lemmas that are used in the proofs of both the Frobenius and spectral norm results.

First, we relate $\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_\xi^2$, for $\xi = 2, F$, to $\left\|A - B_\ell B_\ell^T A\right\|_\xi^2$ plus an error term; we do so since the columns of B_ℓ are orthonormal which will allow us to bound $\left\|A - B_\ell B_\ell^T A\right\|_\xi^2$ using arguments similar to those used to bound $\left\|A - H_k H_k^T A\right\|_\xi^2$ in Theorems 2 and 3.

Lemma 1 *For $\xi = 2, F$ and for any $\epsilon > 0$:*

$$\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_\xi^2 \le \left(1 + \frac{\epsilon}{100}\right) \left\|A - B_\ell B_\ell^T A\right\|_\xi^2 + \left(1 + \frac{100}{\epsilon}\right) \left\|B_\ell B_\ell^T - \tilde{H}_\ell \tilde{H}_\ell^T\right\|_\xi^2 \|A\|_\xi^2$$

Proof: By subadditivity and submultiplicitivity,

$$\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_\xi^2 \le \left(\left\|A - B_\ell B_\ell^T A\right\|_\xi + \left\|B_\ell B_\ell^T - \tilde{H}_\ell \tilde{H}_\ell^T\right\|_\xi \|A\|_\xi \right)^2 .$$

The lemma follows since $(\alpha+\beta)^2 \le (1+\varepsilon)\alpha^2 + (1+1/\varepsilon)\beta^2$ for all $\varepsilon \ge 0$.

◇

Second, although the vectors $\tilde{h}^t = Cz^t/\sigma_t(W), t = 1, \ldots, \ell$ do not in general form an orthonormal set, one would expect from their construction that if the matrix $W^T W$ is close to the matrix

C^TC, then with high probability they will be approximately orthonormal. Lemma 2 establishes that Δ, defined in (34), characterizes how far $\tilde{H}_\ell$ is from having orthonormal columns and shows that the error introduced due to this nonorthonormality is bounded by a simple function of γ and the error introduced at the second level of sampling.

Lemma 2 *When written in the basis with respect to Z:*

$$\tilde{H}_\ell^T \tilde{H}_\ell = I_\ell + \Delta.$$

Furthermore, for $\xi = 2, F$

$$\|\Delta\|_\xi \le \frac{1}{\gamma \|W\|_F^2} \left\|C^TC - W^TW\right\|_\xi .$$

Proof: Recall that $\tilde{h}^t = Cz^t/\sigma_t(W)$ is the t-th column of $\tilde{H}_\ell$ and that $\left|{z^t}^T W^TWz^{t'}\right| = \sigma_t^2(W)\delta_{tt'}$, where $\delta_{tt'}$ is the Kronecker delta function. First note that

$$\tilde{h}^{t^T}\tilde{h}^t = \frac{{z^t}^T C^TCz^t}{\sigma_t^2(W)} = \frac{{z^t}^T W^TWz^t}{\sigma_t^2(W)} + \frac{{z^t}^T \left(C^TC - W^TW\right) z^t}{\sigma_t^2(W)} = 1 + \Delta_{tt}.$$

Next note that if $t \neq t'$ then

$$\tilde{h}^{t^T}\tilde{h}^{t'} = \frac{{z^t}^T C^TCz^{t'}}{\sigma_t(W)\sigma_{t'}(W)} = \frac{{z^t}^T W^TWz^{t'}}{\sigma_t(W)\sigma_{t'}(W)} + \frac{{z^t}^T \left(C^TC - W^TW\right) z^{t'}}{\sigma_t(W)\sigma_{t'}(W)} = \Delta_{tt'}$$

Finally, since $|\Delta_{tt'}| \le \frac{1}{\gamma\|W\|_F^2}\left|\left(Z_{1,\ell}^T(C^TC - W^TW)Z_{1,\ell}\right)_{tt'}\right|$ for every t and t', it follows that

$$\|\Delta\|_F \le \frac{1}{\gamma \|W\|_F^2} \left\|C^TC - W^TW\right\|_F .$$

Similarly, by submultiplicitivity,

$$\|\Delta\|_2 \le \|R\|_2^2 \, \|Z_{1,\ell}\|_2^2 \left\|C^TC - W^TW\right\|_2 \le \frac{1}{\gamma \|W\|_F^2} \left\|C^TC - W^TW\right\|_2 .$$

The lemma then follows.

◇

Third, we consider the second term in Lemma 1, $\left\|B_\ell B_\ell^T - \tilde{H}_\ell \tilde{H}_\ell^T\right\|_\xi^2$ and show that it can be related to $\|\Delta\|_\xi$.

Lemma 3 *For $\xi = 2, F$*

$$\left\|B_\ell B_\ell^T - \tilde{H}_\ell \tilde{H}_\ell^T\right\|_\xi = \|\Delta\|_\xi .$$

Proof: Since $\tilde{H}_\ell = B_\ell \Sigma_{\tilde{H}_\ell} D_\ell^T$, we have

$$\begin{aligned} \left\|B_\ell B_\ell^T - \tilde{H}_\ell \tilde{H}_\ell^T\right\|_\xi &= \left\|B_\ell \left(I_\ell - \Sigma_{\tilde{H}_\ell}^2\right) B_\ell^T\right\|_\xi \\ &= \left\|I_\ell - \Sigma_{\tilde{H}_\ell}^2\right\|_\xi \\ &= \left\|D_\ell \left(I_\ell - \Sigma_{\tilde{H}_\ell}^2\right) D_\ell^T\right\|_\xi \\ &= \left\|I_\ell - \tilde{H}_\ell^T \tilde{H}_\ell\right\|_\xi . \end{aligned}$$

◇

Fourth, Lemma 4 considers the special case in which the probabilities $\{p_i\}_{i=1}^n$ that are entered into the ConstantTimeSVD algorithm are optimal, as is the case for Theorem 5.

Lemma 4 *Let $A \in \mathbb{R}^{m \times n}$ and let $\tilde{H}_\ell$ be constructed from the* ConstantTimeSVD *algorithm by sampling c columns of A with probabilities $\{p_i\}_{i=1}^n$ and w rows of C with probabilities $\{q_j\}_{j=1}^m$, where $p_i = \mathbf{Pr}\left[i_t = i\right] = \left|A^{(i)}\right|^2 / \|A\|_F^2$ and $q_j = \mathbf{Pr}\left[j_t = j\right] = \left|C_{(j)}\right|^2 / \|C\|_F^2$. Then,*

$$\|W\|_F = \|C\|_F = \|A\|_F .$$

Proof: If $p_i = \left|A^{(i)}\right|^2 / \|A\|_F^2$ then we have that $\|C\|_F^2 = \sum_{t=1}^c \left|C^{(t)}\right|^2 = \sum_{t=1}^c \frac{\left|A^{(i_t)}\right|^2}{cp_{i_t}} = \|A\|_F^2$. Similarly, if $q_j = \left|C_{(j)}\right|^2 / \|C\|_F^2$ then we have that $\|W\|_F^2 = \sum_{t=1}^w \left|W_{(t)}\right|^2 = \sum_{t=1}^w \frac{\left|C_{(i_t)}\right|^2}{wq_{i_t}} = \|C\|_F^2$. The lemma follows.

◇

5.4.2 Lemmas for the Frobenius norm proof

In this section we prove (35). We do this by first proving lemmas sufficient to bound $\left\|A - B_\ell B_\ell^T A\right\|_F^2$; when this is combined with the lemmas of Section 5.4.1 we obtain a bound on $\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2$. The bound on $\left\|A - B_\ell B_\ell^T A\right\|_F^2$ depends on the error for the optimal rank k approximation to A, i.e. $\|A - A_k\|_F^2$, and additional errors that depend on the quality of the sampling approximations, i.e. on $\left\|AA^T - CC^T\right\|_F$ and $\left\|C^T C - W^T W\right\|_F$. This will be the analogue of Theorem 2 applied to the constant additional space and time model. The result and associated proof will have a similar structure to that of Theorem 2, but will be more complicated due to the nonorthonormality of the vectors $\tilde{h}^t, t = 1, \ldots, \ell$, and will involve additional error terms since two levels of approximation are involved.

We now prove several lemmas which will provide a bound for the first term in Lemma 1 when applied to the Frobenius norm. We first rewrite the $\left\|A - B_\ell B_\ell^T A\right\|_F^2$ term from Lemma 1. Note that Lemma 5 is the constant time analogue of (14).

Lemma 5

$$\left\|A - B_\ell B_\ell^T A\right\|_F^2 = \|A\|_F^2 - \left\|B_\ell^T A\right\|_F^2$$

Proof:

$$\begin{aligned} \left\|A - B_\ell B_\ell^T A\right\|_F^2 &= \mathbf{Tr}\left(\left(A - B_\ell B_\ell^T A\right)^T \left(A - B_\ell B_\ell^T A\right)\right) \\ &= \mathbf{Tr}\left(A^T A - A^T B_\ell B_\ell^T A\right) \end{aligned}$$

◇

Next, we want to provide a lower bound for $\left\|B_\ell^T A\right\|_F^2$ in terms of the singular values of W. We do so in several steps. First, we relate $\left\|B_\ell^T A\right\|_F^2$ to $\left\|\tilde{H}_\ell^T A\right\|_F^2$. We note that the assumption $\|\Delta\|_F < 1$ is made since in Theorem 5 optimal probabilities are used and sufficiently many columns and rows are drawn; if this assumption is dropped then bounds of the form in Theorem 5 may be obtained with slightly worse sampling complexity [14].

Lemma 6 *If* $\|\Delta\|_F < 1$ *then*

$$\left\|B_\ell^T A\right\|_F^2 \geq (1 - 4\|\Delta\|_F) \left\|\tilde{H}_\ell^T A\right\|_F^2$$

Proof: We first provide an upper bound on $\left\|\tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2$ in terms of $\left\|B_\ell^T A\right\|_F^2$. To that end, note that

$$\left\|\tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2 = \left\|B_\ell \Sigma_{\tilde{H}_\ell}^2 B_\ell^T A\right\|_F^2 = \left\|\Sigma_{\tilde{H}_\ell}^2 B_\ell^T A\right\|_F^2 \leq \left\|(\sigma_{\tilde{H}_\ell}^{max})^2 I_\ell B_\ell^T A\right\|_F^2,$$

where $\sigma_{\tilde{H}_\ell}^{max}$ is the largest singular value of $\tilde{H}_\ell$. Since

$$(\sigma_{\tilde{H}_\ell}^{max})^2 = \sigma_{\tilde{H}_\ell^T \tilde{H}_\ell}^{max} = \left\|\tilde{H}_\ell^T \tilde{H}_\ell\right\|_2$$

it follows that

$$\begin{aligned}
\left\|\tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2 &\leq \left\|\tilde{H}_\ell^T \tilde{H}_\ell\right\|_2^2 \left\|B_\ell^T A\right\|_F^2 \\
&\leq \left(\|I_\ell\|_2 + \left\|\tilde{H}_\ell^T \tilde{H}_\ell - I_\ell\right\|_2\right)^2 \left\|B_\ell^T A\right\|_F^2 \\
&= (1 + \|\Delta\|_2)^2 \left\|B_\ell^T A\right\|_F^2. \qquad (37)
\end{aligned}$$

We next provide an lower bound on $\left\|\tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2$ in terms of $\left\|\tilde{H}_\ell^T A\right\|_F^2$:

$$\begin{aligned}
\left\|\tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2 &= \mathbf{Tr}\left(A^T \tilde{H}_\ell \tilde{H}_\ell^T \tilde{H}_\ell \tilde{H}_\ell^T A\right) \\
&= \mathbf{Tr}\left(A^T \tilde{H}_\ell (I_\ell + \Delta) \tilde{H}_\ell^T A\right) \\
&= \mathbf{Tr}\left(A^T \tilde{H}_\ell \tilde{H}_\ell^T A\right) + \mathbf{Tr}\left(A^T \tilde{H}_\ell \Delta \tilde{H}_\ell^T A\right) \\
&\geq \left\|\tilde{H}_\ell^T A\right\|_F^2 - \|\Delta\|_2 \left\|\tilde{H}_\ell^T A\right\|_F^2. \qquad (38)
\end{aligned}$$

Denoting the t-th column and row of a matrix X as $(X)^{(t)}$ and $(X)_{(t)}$, respectively, (38) follows since

$$\begin{aligned}
\left|\mathbf{Tr}\left(A^T \tilde{H}_\ell \Delta \tilde{H}_\ell^T A\right)\right| &= \left|\sum_t (A^T \tilde{H}_\ell)_{(t)} \Delta (\tilde{H}_\ell^T A)^{(t)}\right| \\
&\leq \sum_t \left|(A^T \tilde{H}_\ell)_{(t)} \Delta (\tilde{H}_\ell^T A)^{(t)}\right| \\
&\leq \sum_t \left|(A^T \tilde{H}_\ell)_{(t)}\right| \|\Delta\|_2 \left|(\tilde{H}_\ell^T A)^{(t)}\right| \\
&= \|\Delta\|_2 \left\|A^T \tilde{H}_\ell\right\|_F^2.
\end{aligned}$$

By combining (37) and (38) and using that $\|\cdot\|_2 \leq \|\cdot\|_F$ we have that

$$\left\|B_\ell^T A\right\|_F^2 \geq \frac{1 - \|\Delta\|_F}{(1 + \|\Delta\|_F)^2} \left\|\tilde{H}_\ell^T A\right\|_F^2.$$

The lemma follows since $\frac{1-x}{(1+x)^2} \geq 1 - 4x$ if $x \leq 1$.

$\diamond$

Second, we relate $\left\|\tilde{H}_\ell^T A\right\|_F^2$ to $\left\|\tilde{H}_\ell^T C\right\|_F^2$.

Lemma 7

$$\left\|\tilde{H}_\ell^T A\right\|_F^2 \geq \left\|\tilde{H}_\ell^T C\right\|_F^2 - \left(k + \sqrt{k}\,\|\Delta\|_F\right)\,\left\|AA^T - CC^T\right\|_F$$

Proof: Since $\left\|\tilde{H}_\ell^T A\right\|_F^2 = \mathbf{Tr}\left(\tilde{H}_\ell^T AA^T \tilde{H}_\ell^T\right)$, we have that

$$\begin{aligned}\left\|\tilde{H}_\ell^T A\right\|_F^2 &= \mathbf{Tr}\left(\tilde{H}_\ell^T CC^T \tilde{H}_\ell^T\right) + \mathbf{Tr}\left(\tilde{H}_\ell^T (AA^T - CC^T)\tilde{H}_\ell^T\right) \\ &\geq \left\|\tilde{H}_\ell^T C\right\|_F^2 - \left\|AA^T - CC^T\right\|_2 \left\|\tilde{H}_\ell\right\|_F^2,\end{aligned}$$

where the inequality follows since

$$\left|\mathbf{Tr}\left(\tilde{H}_\ell^T (AA^T - CC^T)\tilde{H}_\ell^T\right)\right| \leq \sum_t \left|(\tilde{H}_\ell^T)_{(t)}(AA^T - CC^T)(\tilde{H}_\ell)^{(t)}\right| \leq \left\|AA^T - CC^T\right\|_2 \left\|\tilde{H}_\ell\right\|_F^2.$$

The lemma follows since $\|\cdot\|_2 \leq \|\cdot\|_F$ and since

$$\left\|\tilde{H}_\ell\right\|_F^2 = \sum_{t=1}^{\ell} \left|\tilde{h}^{t^T}\tilde{h}^t\right| = \sum_{t=1}^{\ell} 1 + \Delta_{tt} \leq k + \sqrt{k}\,\|\Delta\|_F.$$

◇

Third, we relate $\left\|\tilde{H}_\ell^T C\right\|_F^2$ to $\sum_{t=1}^{\ell}\sigma_t^2(W)$.

Lemma 8

$$\left\|\tilde{H}_\ell^T C\right\|_F^2 \geq \sum_{t=1}^{\ell}\sigma_t^2(W) - \frac{2}{\sqrt{\gamma}}\left\|C^TC - W^TW\right\|_F$$

Proof: Since $\left\|\tilde{H}_\ell^T C\right\|_F^2 = \left\|C^T\tilde{H}_\ell\right\|_F^2 = \left\|C^TCZ_{1,\ell}T\right\|_F^2$, we have

$$\begin{aligned}\left\|\tilde{H}_\ell^T C\right\|_F^2 &\geq \left(\left\|W^TWZ_{1,\ell}T\right\|_F - \left\|(C^TC - W^TW)Z_{1,\ell}T\right\|_F\right)^2 \\ &\geq \left(\left(\sum_{t=1}^{\ell}\sigma_t^2(W)\right)^{1/2} - \frac{1}{\sqrt{\gamma}\,\|W\|_F}\left\|(C^TC - W^TW)\right\|_F\right)^2,\end{aligned}$$

where the second inequality uses that $\|XZ\|_F \leq \|X\|_F$ for any matrix X if the matrix Z has orthonormal columns. By multiplying out the right hand side and ignoring terms that reinforce the inequality, the lemma follows since $\frac{\left(\sum_{t=1}^{\ell}\sigma_t^2(W)\right)^{1/2}}{\|W\|_F} \leq 1$.

◇

By combining Lemmas 6, 7, and 8, we have our desired bound on $\left\|B_\ell^T A\right\|_F^2$ in terms of the singular values of W. Finally, we use matrix perturbation theory to relate $\sum_{t=1}^{\ell}\sigma_t^2(W)$ to $\sum_{t=1}^{k}\sigma_t^2(A)$.

Lemma 9

$$\sum_{t=1}^{\ell}\sigma_t^2(W) \geq \sum_{t=1}^{k}\sigma_t^2(A) - \sqrt{k}\left\|AA^T - CC^T\right\|_F - \sqrt{k}\left\|C^TC - W^TW\right\|_F - (k-\ell)\gamma\,\|W\|_F^2$$

Proof: Recalling the Hoffman-Wielandt inequality, we see that

$$\begin{aligned} \left|\sum_{t=1}^{k} \left(\sigma_t^2(C)-\sigma_t^2(A)\right)\right| &\leq \sqrt{k}\left(\sum_{t=1}^{k} \left(\sigma_t^2(C)-\sigma_t^2(A)\right)^2\right)^{1/2} \\ &\leq \sqrt{k}\left(\sum_{t=1}^{k}\left(\sigma_t(CC^T)-\sigma_t(AA^T)\right)^2\right)^{1/2} \\ &\leq \sqrt{k}\left\|AA^T-CC^T\right\|_F , \end{aligned}\tag{39}$$

and, similarly, that

$$\begin{aligned} \left|\sum_{t=1}^{k} \left(\sigma_t^2(W)-\sigma_t^2(C)\right)\right| &\leq \sqrt{k}\left(\sum_{t=1}^{k} \left(\sigma_t^2(W)-\sigma_t^2(C)\right)^2\right)^{1/2} \\ &\leq \sqrt{k}\left(\sum_{t=1}^{k}\left(\sigma_t(WW^T)-\sigma_t(CC^T)\right)^2\right)^{1/2} \\ &\leq \sqrt{k}\left\|C^TC-W^TW\right\|_F . \end{aligned}\tag{40}$$

By combining (39) and (40) we see that

$$\left|\sum_{t=1}^{k}\sigma_t^2(W)-\sum_{t=1}^{k}\sigma_t^2(A)\right| \quad \leq \quad \sqrt{k}\left\|AA^T-CC^T\right\|_F+\sqrt{k}\left\|C^TC-W^TW\right\|_F . \tag{41}$$

Since $\sigma_t^2(W) < \gamma\left\|W\right\|_F^2$ for all $t=\ell+1,\ldots,k$ we have that $\sum_{t=\ell+1}^{k}\sigma_t^2(W)\leq(k-\ell)\gamma\left\|W\right\|_F^2$. Combining this with (41) allows us to relate $\sum_{t=1}^{\ell}\sigma_t^2(W)$ and $\sum_{t=1}^{k}\sigma_t^2(A)$, thus establishing the lemma.

◇

Now we combine these results in order to prove (35). Let $E_{AA^T}=AA^T-CC^T$ and $E_{C^TC}=C^TC-W^TW$. First, we establish a lower bound on $\left\|B_\ell^TA\right\|_F^2$. By combining Lemmas 6 and 7 and dropping terms that reinforce the inequality, we have that

$$\left\|B_\ell^TA\right\|_F^2 \geq \left\|\tilde{H}_\ell^TC\right\|_F^2 - 4\left\|\Delta\right\|_F\left\|\tilde{H}_\ell^TC\right\|_F^2-\left(k+\sqrt{k}\left\|\Delta\right\|_F\right)\left\|E_{AA^T}\right\|_F .$$

By combining this with Lemmas 8 and 9 and dropping terms that reinforce the inequality, we have that

$$\begin{aligned} \left\|B_\ell^TA\right\|_F^2 \geq \sum_{t=1}^{k}\sigma_t^2(A)-\left(k+\sqrt{k}\right)\left\|E_{AA^T}\right\|_F-\left(\sqrt{k}+\frac{2}{\sqrt{\gamma}}\right)\left\|E_{C^TC}\right\|_F \\ -4\left\|\Delta\right\|_F\sum_{t=1}^{k}\sigma_t^2(A)-\sqrt{k}\left\|\Delta\right\|_F\left\|E_{AA^T}\right\|_F-(k-\ell)\gamma\left\|W\right\|_F^2 . \end{aligned}\tag{42}$$

From Lemma 5 this immediately leads to the upper bound on $\left\|A-B_\ell B_\ell^TA\right\|_F^2$,

$$\begin{aligned} \left\|A-B_\ell B_\ell^TA\right\|_F^2 \leq \left\|A-A_k\right\|_F^2+\left(k+\sqrt{k}\right)\left\|E_{AA^T}\right\|_F+\left(\sqrt{k}+\frac{2}{\sqrt{\gamma}}\right)\left\|E_{C^TC}\right\|_F \\ +4\left\|\Delta\right\|_F\sum_{t=1}^{k}\sigma_t^2(A)+\sqrt{k}\left\|\Delta\right\|_F\left\|E_{AA^T}\right\|_F+(k-\ell)\gamma\left\|W\right\|_F^2 . \end{aligned}\tag{43}$$

From Lemmas 1 and 3,

$$\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_F^2 \leq \left(1 + \frac{\epsilon}{100}\right) \left\|A - B_\ell B_\ell^T A\right\|_F^2 + \left(1 + \frac{100}{\epsilon}\right) \|\Delta\|_F^2 \, \|A\|_F^2 \,. \tag{44}$$

Recall that $\gamma = \epsilon/100k$, that $\sum_{t=1}^k \sigma_t^2(A) \leq \|A\|_F^2$, that $\|\Delta\|_F \leq \|E_{C^TC}\|_F / \gamma \|W\|_F^2$ by Lemma 2, and that $\|W\|_F = \|C\|_F = \|A\|_F$ by Lemma 4; (35) then follows by combining (43) and (44), using the sampling probabilities indicated in the statement of the theorem, and by choosing $c, w = \Omega(k^2\eta^2/\epsilon^4)$.

5.4.3 Lemmas for the spectral norm proof

In this section we prove (36). We do this by first proving lemmas sufficient to bound $\left\|A - B_\ell B_\ell^T A\right\|_2^2$; when this is combined with the lemmas of Section 5.4.1, we obtain a bound on $\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_2^2$. The bound on $\left\|A - B_\ell B_\ell^T A\right\|_2^2$ depends on the error for the optimal rank k approximation to A, i.e. $\|A - A_k\|_2^2$, and additional errors that depend on the quality of the sampling approximations, i.e. on $\left\|AA^T - CC^T\right\|_2$ and $\left\|C^TC - W^TW\right\|_2$. This will be the analogue of Theorem 3 applied to the constant additional space and time model. The result and associated proof will have a similar structure to that of Theorem 3, but will be more complicated due to the nonorthonormality of the vectors $\tilde{h}^t, t = 1, \ldots, \ell$, and will involve additional error terms since two levels of approximation are involved.

We now prove three lemmas which will provide a bound for the first term in Lemma 1 when applied to the spectral norm. We first rewrite the $\left\|A - B_\ell B_\ell^T A\right\|_2^2$ term from Lemma 1.

Lemma 10

$$\left\|A - B_\ell B_\ell^T A\right\|_2^2 \leq \left\|Z_{k+1,c}^T C^T C Z_{k+1,c}\right\|_2 + \left\|Z_{\ell+1,k}^T C^T C Z_{\ell+1,k}\right\|_2 + \left\|AA^T - CC^T\right\|_2$$

Proof: In order to bound $\left\|A - B_\ell B_\ell^T A\right\|_2$ we will project onto the subspace spanned by B_ℓ and its orthogonal complement in a manner analogous to that used in the proof of Theorem 3. Let $\mathcal{B}_\ell = \mathrm{range}(B_\ell)$ and let $\mathcal{B}_{m-\ell}$ be the orthogonal complement of $\mathcal{B}_\ell$. Let $x = \alpha y + \beta z$ where $y \in \mathcal{B}_\ell$, $z \in \mathcal{B}_{m-\ell}$, and $\alpha^2 + \beta^2 = 1$. Then,

$$\begin{aligned}
\left\|A - B_\ell B_\ell^T A\right\|_2 &= \max_{x \in \mathbb{R}^m, |x|=1} \left|x^T(A - B_\ell B_\ell^T A)\right| \\
&= \max_{y \in \mathcal{B}_\ell, |y|=1, z \in \mathcal{B}_{m-\ell}, |z|=1, \alpha^2+\beta^2=1} \left|(\alpha y^T + \beta z^T)(A - B_\ell B_\ell^T A)\right| \\
&\leq \max_{y \in \mathcal{B}_\ell, |y|=1} \left|y^T(A - B_\ell B_\ell^T A)\right| + \max_{z \in \mathcal{B}_{m-\ell}, |z|=1} \left|z^T(A - B_\ell B_\ell^T A)\right| \qquad (45) \\
&= \max_{z \in \mathcal{B}_{m-\ell}, |z|=1} \left|z^T A\right|. \qquad (46)
\end{aligned}$$

(45) follows since $\alpha, \beta \leq 1$ and (46) follows since $y \in \mathcal{B}_\ell$ and $z \in \mathcal{B}_{m-\ell}$. To bound (46), let $z \in \mathcal{B}_{m-\ell}$, $|z| = 1$; then

$$\begin{aligned}
\left|z^T A\right|^2 &= z^T \left(AA^T\right) z \\
&= z^T \left(CC^T\right) z + z^T \left(AA^T - CC^T\right) z \\
&= z^T \left(CC^T - CZ_{1,k} Z_{1,k}^T C^T\right) z + z^T \left(CZ_{1,k} Z_{1,k}^T C^T\right) z + z^T \left(AA^T - CC^T\right) z \qquad (47) \\
&= z^T \left(CZ_{k+1,c} Z_{k+1,c}^T C^T\right) z + z^T \left(CZ_{\ell+1,k} Z_{\ell+1,k}^T C^T\right) z + z^T \left(AA^T - CC^T\right) z. \qquad (48)
\end{aligned}$$

(48) follows since $I_c = ZZ^T = Z_{1,k}Z_{1,k}^T + Z_{k+1,c}Z_{k+1,c}^T$ and since

$$CZ_{1,\ell}Z_{1,\ell}^T C^T = \sum_{t=1}^{\ell} Cz^t {z^t}^T C^T = \sum_{t=1}^{\ell} \sigma_t^2(W)\tilde{h}^t {\tilde{h}^t}^T,$$

implies that

$$z^T CZ_{1,\ell}Z_{1,\ell}^T C^T z = 0 \tag{49}$$

for $z \in \mathcal{B}_{m-\ell}$. Thus, by combining (46) and (48)

$$\left\|A - B_\ell B_\ell^T A\right\|_2^2 \leq \left\|CZ_{k+1,c}Z_{k+1,c}^T C^T\right\|_2 + \left\|CZ_{\ell+1,k}Z_{\ell+1,k}^T C^T\right\|_2 + \left\|AA^T - CC^T\right\|_2 .$$

The lemma follows since $\left\|X^T X\right\|_2 = \left\|XX^T\right\|_2$ for any matrix X.

◇

We next bound the $\left\|Z_{k+1,c}^T C^T CZ_{k+1,c}\right\|_2$ term from Lemma 10; note that matrix perturbation theory is used in (52).

Lemma 11

$$\left\|Z_{k+1,c}^T C^T CZ_{k+1,c}\right\|_2 \leq \|A - A_k\|_2^2 + \left\|AA^T - CC^T\right\|_2 + 2\left\|C^T C - W^T W\right\|_2$$

Proof: First note that

$$\left\|Z_{k+1,c}^T C^T CZ_{k+1,c}\right\|_2 \leq \left\|Z_{k+1,c}^T W^T WZ_{k+1,c}\right\|_2 + \left\|Z_{k+1,c}^T (C^T C - W^T W)Z_{k+1,c}\right\|_2 . \tag{50}$$

Since

$$\begin{aligned}\left\|Z_{k+1,c}^T (C^T C - W^T W)Z_{k+1,c}\right\|_2 &\leq \left\|C^T C - W^T W\right\|_2 \|Z_{k+1,c}\|_2^2 \\ &= \left\|C^T C - W^T W\right\|_2\end{aligned}$$

and

$$\left\|Z_{k+1,c}^T W^T WZ_{k+1,c}\right\|_2 = \sigma_{k+1}^2(W).$$

It follows from (50) that

$$\left\|Z_{k+1,c}^T C^T CZ_{k+1,c}\right\|_2 \leq \sigma_{k+1}^2(W) + \left\|C^T C - W^T W\right\|_2 . \tag{51}$$

By a double application of (9), we see that

$$\sigma_{k+1}^2(W) \leq \sigma_{k+1}^2(A) + \left\|AA^T - CC^T\right\|_2 + \left\|C^T C - W^T W\right\|_2 . \tag{52}$$

The lemma follows by combining (51) and (52) since $\|A - A_k\|_2 = \sigma_{k+1}(A)$.

◇

Finally, we bound the $\left\|Z_{\ell+1,k}^T C^T CZ_{\ell+1,k}\right\|_2$ term from Lemma 10; note that if $\ell = k$ it is unnecessary.

Lemma 12

$$\left\|Z_{\ell+1,k}^T C^T CZ_{\ell+1,k}\right\|_2 \leq \left\|C^T C - W^T W\right\|_2 + \gamma \|W\|_F^2$$

Proof: First note that

$$\left\|Z_{\ell+1,k}^T C^T C Z_{\ell+1,k}\right\|_2 \leq \left\|Z_{\ell+1,k}^T \left(C^T C - W^T W\right) Z_{\ell+1,k}\right\|_2 + \left\|Z_{\ell+1,k}^T W^T W Z_{\ell+1,k}\right\|_2 . \tag{53}$$

Since

$$\begin{aligned} \left\|Z_{\ell+1,k}^T \left(C^T C - W^T W\right) Z_{\ell+1,k}\right\|_2 &\leq \left\|C^T C - W^T W\right\|_2 \left\|Z_{\ell+1,k}\right\|_2^2 \\ &= \left\|C^T C - W^T W\right\|_2 , \end{aligned}$$

and

$$\left\|Z_{\ell+1,k}^T W^T W Z_{\ell+1,k}\right\|_2 = \sigma_{\ell+1}^2(W),$$

It follows from (53) that

$$\left\|Z_{\ell+1,k}^T C^T C Z_{\ell+1,k}\right\|_2 \leq \left\|C^T C - W^T W\right\|_2 + \sigma_{\ell+1}^2(W). \tag{54}$$

The lemma follows since $\sigma_t^2(W) < \gamma \left\|W\right\|_F^2$ for all $t = \ell+1, \ldots, k$.

◇

Now we combine these results in order to prove (36). Recall that $E_{AA^T} = AA^T - CC^T$ and $E_{C^TC} = C^T C - W^T W$. By combining Lemmas 10, 11, and 12, we have that

$$\left\|A - B_\ell B_\ell^T A\right\|_2^2 \leq \left\|A - A_k\right\|_2^2 + 2\left\|E_{AA^T}\right\|_2 + 3\left\|E_{C^TC}\right\|_2 + \gamma \left\|W\right\|_F^2 . \tag{55}$$

From Lemmas 1 and 3,

$$\left\|A - \tilde{H}_\ell \tilde{H}_\ell^T A\right\|_2^2 \leq \left(1 + \frac{\epsilon}{100}\right) \left\|A - B_\ell B_\ell^T A\right\|_2^2 + \left(1 + \frac{100}{\epsilon}\right) \left\|\Delta\right\|_2^2 \left\|A\right\|_2^2 . \tag{56}$$

Recall that $\gamma = \epsilon/100$, that $\left\|\cdot\right\|_2 \leq \left\|\cdot\right\|_F$, and that $\left\|\Delta\right\|_2 \leq \left\|E_{C^TC}\right\|_2 / \gamma \left\|W\right\|_F^2$ by Lemma 2; (36) follows by combining (55) and (56), using the sampling probabilities indicated in the statement of the theorem, and by choosing $c, w = \Omega(\eta^2/\epsilon^4)$.

6 Discussion and Conclusion

We have presented two algorithms to compute approximations to the SVD of a matrix $A \in \mathbb{R}^{m \times n}$ which do not require that A be stored in RAM, but for which the additional space and time required (in addition to a constant number of passes over the matrix) is either linear in $m+n$ or is a constant independent of m and n; we have also proven error bounds for both algorithms with respect to both the Frobenius and spectral norms. Figure 1 in Section 1 presents a summary of the dependence of the sampling complexity on k and ϵ. With the LinearTimeSVD algorithm, the additional error (beyond the optimal rank k approximation) in the spectral norm bound can be made less than $\epsilon \left\|A\right\|_F^2$ by sampling $\Theta(1/\epsilon^2)$ columns and the additional error in the Frobenius norm can be made less than $\epsilon \left\|A\right\|_F^2$ by sampling $\Theta(k/\epsilon^2)$ columns. Likewise, with the ConstantTimeSVD algorithm, the additional error in the spectral norm can be made less than $\epsilon \left\|A\right\|_F^2$ by sampling $\Theta(1/\epsilon^4)$ columns and rows and the additional error in the Frobenius norm can be made less than $\epsilon \left\|A\right\|_F^2$ by sampling $\Theta(k^2/\epsilon^4)$ columns and rows. The results of [18] require $\Theta(k^4/\epsilon^3)$ columns and rows for the Frobenius (and thus the spectral) norm bound.

Recent work has focused on developing new techniques for proving lower bounds on the number of queries a sampling algorithm is required to perform in order to approximate a given function accurately with a low probability of error [4, 5]. In [5] these methods have been applied to the low-rank matrix approximation problem (defined as approximating the SVD with respect to the Frobenius norm) and to the matrix reconstruction problem. It is shown that any sampling algorithm that with high probability finds a good low-rank approximation requires $\Omega(m+n)$ queries. In addition, it is shown that even if the algorithm is given the exact weight distribution over the columns of a matrix it will still require $\Omega(k/\epsilon^2)$ column queries to approximate A. Thus, the LinearTimeSVD algorithm (see also the original [11]) is optimal with respect to Frobenius norm bounds for the rank parameter k and the ConstantTimeSVD algorithm (see also the original [18]) is optimal with respect to Frobenius norm bounds up to polynomial factors.

Acknowledgments We would like to thank the following individuals for comments and fruitful discussions: Dimitris Achlioptas, Alan Frieze, Mauro Maggioni, Frank McSherry, and Santosh Vempala. We would also like to thank Dimitris Achlioptas and Frank McSherry for providing us with a preprint of [1], i.e., the journal version of [2], which provides a useful comparison of their results with ours. Finally, we would like to thank the National Science Foundation for partial support of this work.

References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. *submitted.*

[2] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 611–618, 2001.

[3] O. Alter, P.O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[4] Z. Bar-Yossef. *The Complexity of Massive Data Set Computations.* PhD thesis, University of California, Berkeley, 2002.

[5] Z. Bar-Yossef. Sampling lower bounds via information theory. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 335–344, 2003.

[6] M.W. Berry, Z. Drmac, and E.R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, 1999.

[7] M.W. Berry, S.T. Dumais, and G.W. O'Brian. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[8] R. Bhatia. *Matrix Analysis.* Springer-Verlag, New York, 1997.

[9] S.T. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[10] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. *submitted.*

[11] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 291–299, 1999.

[12] P. Drineas and R. Kannan. Fast Monte-Carlo algorithms for approximate matrix multiplication. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 452–459, 2001.

[13] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 223–232, 2003.

[14] P. Drineas, R. Kannan, and M.W. Mahoney. (unpublished results).

[15] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. Technical Report YALEU/DCS/TR-1269, Yale University Department of Computer Science, New Haven, CT, February 2004.

[16] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. Technical Report YALEU/DCS/TR-1271, Yale University Department of Computer Science, New Haven, CT, February 2004.

[17] A. Frieze, R. Kannan, and S. Vempala. Fast Monte Carlo algorithms for finding low-rank approximations. *submitted.*

[18] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 370–378, 1998.

[19] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989.

[20] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[21] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 189–197, 2000.

[22] J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 599–608, 1997.

[23] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[24] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, London, 1979.

[25] H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[26] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 159–168, 1998.

[27] S. Raychaudhuri, J.M. Stuart, and R.B. Altman. Principal component analysis to summarize microarray experiments: application to sporulation time series. In *Pacific Symposium on Biocomputing 2000*, pages 452–463, 2000.

[28] G.W. Stewart and J.G. Sun. *Matrix Perturbation Theory.* Academic Press, New York, 1990.

[29] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[30] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–96, 1991.

[31] S. Vempala. Random projection: A new approach to VLSI layout. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 389–395, 1998.