# Subspace Sampling and Relative-Error Matrix Approximation: Column-Based Methods

Petros Drineas[1], Michael W. Mahoney[2,⋆], and S. Muthukrishnan[3]

[1] Department of Computer Science, RPI
[2] Yahoo Research Labs
[3] Department of Computer Science, Rutgers University

**Abstract.** Given an $m \times n$ matrix $A$ and an integer $k$ less than the rank of $A$, the "best" rank $k$ approximation to $A$ that minimizes the error with respect to the Frobenius norm is $A_k$, which is obtained by projecting $A$ on the top $k$ left singular vectors of $A$. While $A_k$ is routinely used in data analysis, it is difficult to interpret and understand it in terms of the *original data*, namely the columns and rows of $A$. For example, these columns and rows often come from some application domain, whereas the singular vectors are linear combinations of (up to all) the columns or rows of $A$. We address the problem of obtaining low-rank approximations that are directly interpretable in terms of the *original* columns or rows of $A$. Our main results are two polynomial time randomized algorithms that take as input a matrix $A$ and return as output a matrix $C$, consisting of a "small" (i.e., a low-degree polynomial in $k$, $1/\epsilon$, and $\log(1/\delta)$) number of actual columns of $A$ such that

$$\left\|A - CC^{+}A\right\|_F \leq (1+\epsilon)\left\|A - A_k\right\|_F$$

with probability at least $1-\delta$. Our algorithms are simple, and they take time of the order of the time needed to compute the top $k$ right singular vectors of $A$. In addition, they sample the columns of $A$ via the method of "subspace sampling," so-named since the sampling probabilities depend on the lengths of the rows of the top singular vectors and since they ensure that we capture entirely a certain subspace of interest.

## 1 Introduction

### 1.1 Motivation and Overview

In many applications, the data are represented by a real $m \times n$ matrix $A$. Such a matrix may arise if the data consist of $m$ objects, each of which is described by $n$ features. Examples of objects include documents, genomes, stocks, hyperspectral images, and web groups, while examples of the corresponding features are terms, environmental conditions, temporal resolution, frequency resolution, and individual users. In each of these application areas, practitioners spend vast

[⋆] Part of this work was done while at the Department of Mathematics, Yale University.

J. Diaz et al. (Eds.): APPROX and RANDOM 2006, LNCS 4110, pp. 316–326, 2006.

amounts of time analyzing the data in order to understand, interpret, and ultimately use this data. Often the central task in this analysis is to develop a compressed representation of $A$ that may be easier to analyze and interpret.

The most common compressed representation of $A$ used by data analysts is that obtained by truncating the SVD at some number $k \ll \min\{m, n\}$ terms, in large part because this provides the "best" rank-$k$ approximation to $A$ when measured with respect to any unitarily invariant matrix norm. However, there is a fundamental difficulty with this representation: the new "dimensions" (the so-called eigencolumns and eigenrows) of $A_k$ are linear combinations of (up to all) the original dimensions. As such, they are notoriously difficult to interpret in terms of the underlying data and processes generating that data. For example, the vector [(1/2) age - $(1/\sqrt{2})$ height + (1/2) income], being one of the significant uncorrelated "factors" from a dataset of people's features is not particularly informative. From an analyst's point of view, it would be highly preferable to have a low-rank approximation that is nearly as good as that provided by the SVD but that is expressed in terms of a small number of *actual columns* and/or *actual rows* of a matrix, rather than linear combinations of those columns and rows. For example, consider recent data analysis work in DNA microarray and DNA Single Nucleotide Polymorphism (SNP) analysis [15, 16, 18], where linear combinations of genes or loci in the human genome have no clear biological interpretation.

In this paper, we focus on choosing columns of a matrix $A$ in order to approximate very precisely a data matrix $A$ as the product $CX$, where $C$ consists of a few columns of $A$ and where $X$ is a matrix that expresses every column of $A$ in terms of the basis provided by the columns of $C$.

### 1.2 Review of Linear Algebra

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For any matrix $A \in \mathbb{R}^{m\times n}$, let $A_{(i)}, i \in [m]$ denote the $i$-th row of $A$ as a row vector, and let $A^{(j)}, j \in [n]$ denote the $j$-th column of $A$ as a column vector. The Singular Value Decomposition (SVD) of $A$ will be denoted by $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times\rho}$, $\Sigma \in \mathbb{R}^{\rho\times\rho}$, $V \in \mathbb{R}^{n\times\rho}$, and where $\rho$ is the rank of $A$. The "best" rank-$k$ approximation to $A$ (with respect to, e.g., the Frobenius norm, $||A||_F = \sqrt{\sum_{i,j} A_{ij}^2}$) will be denoted by $A_k = U_k\Sigma_k V_k^T$, where $U_k \in \mathbb{R}^{m\times k}$ is the first $k$ columns of $U$, etc. The SVD and hence the best rank-$k$ approximation of a general matrix $A$ can be computed in $O(\min\{n^2m, nm^2\})$ time, and optimal rank-$k$ approximations to it can be computed more rapidly with, e.g., Lanczos methods. We will use $SVD(A_k)$ to denote the time to compute $A_k$. For more details on linear algebra, see [1, 12, 14, 17], and for more details on notation and our sampling matrix formalism, see [5, 9].

### 1.3 Problem Definition

We start with the following definition.

**Definition 1.** *Let $A$ be an $m \times n$ matrix, and let $C$ be an $m \times c$ matrix whose columns consist of a small number $c$ of columns of the matrix $A$. Then the $m \times n$*

*matrix $A'$ is a* column-based low-rank matrix approximation *to $A$, or a* CX matrix approximation*, if it may be explicitly written as $A' = CX$ for some $c \times n$ matrix $X$.*

We prefer not to provide too precise a characterization of what we mean by a "small" number of columns, but one should think of $c \ll n$. Also, the low-rank matrix approximation provided by truncating the SVD at some value of $k < \rho = \text{rank}(A)$ will not in general satisfy the conditions of the definition. Finally, given a set of columns $C$, the approximation $A' = P_C A = CC^+A$ clearly satisfies the requirements of Definition 1. Indeed, this is the "best" such approximation to $A$, in the sense that $\|A - C(C^+A)\|_F = \min_{X \in \mathbb{R}^{c \times n}} \|A - CX\|_F$.

The quality of a CX matrix approximation depends on the choice of $C$ as well as on the matrix $X$. We consider the following problem.

**Problem 1** ***(Column-based low-rank matrix approximation problem.)*** *Given a matrix $A \in \mathbb{R}^{m \times n}$ and an integer $k \ll min\{m, n\}$, choose a sufficient number of columns of $A$ such that*

$$\left\|A - CC^+A\right\|_F \leq (1 + \epsilon) \|A - A_k\|_F . \qquad (1)$$

*Here, $C$ is a matrix consisting of the chosen columns of $A$, $CC^+A$ is the projection of $A$ on the subspace spanned by the chosen columns, and $A_k$ is the best rank $k$ approximation to $A$. The number of columns of $C$ should be a function of $k$, $1/\epsilon$, and – in the case of randomized algorithms – a failure probability $\delta$, and the running time of the algorithm should be a low-degree polynomial in $m$ and $n$.*

Note that is not obvious whether there exist, and if so whether one can efficiently find, a small (depending on $k$, $1/\epsilon$, and $1/\delta$, but independent of $m$ and $n$) number of columns that provide such relative-error guarantees.

### 1.4 "Subspace Sampling" and Our Main Result

Our main result is the following theorem, which asserts the existence of two related algorithms to solve Problem 1.

**Theorem 1.** *There exists randomized algorithms that solve Problem 1.*

- *In one algorithm, exactly $c = O(k^2 \log(1/\delta)/\epsilon^2)$ columns of $A$ are chosen to construct $C$.*
- *In the other algorithm, $c = O(k \log k \log(1/\delta)/\epsilon^2)$ columns in expectation are chosen to construct $C$.*

*Both algorithms satisfy (1) with probability at least $1 - \delta$, both run in time $O(SVD(A_k))$, and both use the method of "subspace sampling" to sample columns to form $C$.*

The algorithms of Theorem 1 for constructing a matrix $C$ consisting of a few columns of $A$ are simple:

1. Construct sampling probabilities $\{p_i\}_{i=1}^n$ satisfying the "subspace sampling" Condition (2) below.
2. Use these probabilities to randomly sample columns from $A$ and construct a matrix $C$ using one of two sampling procedures.
3. Repeat these two steps $O(\log(1/\delta))$ times, and return the set of columns $C$ such that $\|A - CC^+A\|_F$ is smallest over all $O(\log(1/\delta))$ trials.

The first sampling procedure, which we call the Exactly$(c)$ sampling algorithm, picks *exactly* $c$ columns of $A$ to be included in $C$ in $c$ i.i.d. trials, where in each trial the $i$-th column of $A$ is picked with probability $p_i$. Notice that some columns of $A$ may be included in the sample more than once. The second sampling procedure, which we call the Expected$(c)$ sampling algorithm, picks *in expectation* at most $c$ columns of $A$ to create $C$, by including the $i$-th column of $A$ in $C$ with probability $\min\{1, cp_i\}$. No column of $A$ is included in the sample more than once.

The key technical insight that leads to the relative-error guarantees is that the columns are selected by a novel sampling procedure that we call "subspace sampling." Rather than sample columns from $A$ with a probability distribution that depends on the Euclidean norms of the columns of $A$ (which gives provable additive-error bounds [5, 6, 7]), in "subspace sampling" we randomly sample columns of $A$ with a probability distribution that depends on the Euclidean norms of the rows of the top $k$ right singular vectors of $A$. This allows us to capture entirely a certain subspace of interest. The "subspace sampling" probabilities $p_i, i \in [n]$ will satisfy

$$p_i \geq \frac{\beta \left|(V_k)_{(i)}\right|^2}{k} \qquad \forall i \in [n], \tag{2}$$

for some $\beta \in (0, 1]$. Note that $\sum_{j=1}^n \left|(V_k)_{(j)}\right|^2 = k$ and that $\sum_{i\in[n]} p_i = 1$. To construct sampling probabilities satisfying Condition (2), it is sufficient to spend $O(SVD(A_k))$ time to compute (exactly or approximately, in which case $\beta = 1$ or $\beta < 1$, respectively) the top $k$ right singular vectors of $A$.

### 1.5 Related Work

The seminal work of Frieze, Kannan and Vempala [10, 11] can be viewed, in our parlance, as sampling columns from a matrix $A$ to form a matrix $C$ such that $\|A - CX\|_F \leq \|A - A_k\|_F + \epsilon \|A\|_F$. The matrix $C$ has $poly(k, 1/\epsilon, 1/\delta)$ columns and is constructed after making only two passes over $A$ using $O(m+n)$ work space. Under similar resource constraints, a series of papers have followed [10, 11] in the past seven years [4, 6, 20], improving the dependency of $c$ on $k, 1/\epsilon$, and $1/\delta$, and analyzing the spectral as well as the Frobenius norm, yielding bounds of the form

$$\|A - CX\|_\xi \leq \|A - A_k\|_\xi + \epsilon \|A\|_F \tag{3}$$

for $\xi = 2, F$, and thus providing additive-error guarantees for column-based low-rank matrix approximations.

Most relevant for our relative-error column-based low-rank matrix approximation of Problem 1 is the recent work of Rademacher, Vempala and Wang [19] and Deshpande, Rademacher, Vempala and Wang [2]. Using two different methods (in one case iterative sampling in a backwards manner and an induction on $k$ argument [19] and in the other case an argument which relies on estimating the volume of the simplex formed by each of the $k$-sized subsets of the columns [2]), they reported the *existence* of a set of $O(k^2/\epsilon^2)$ columns that provide relative-error CX matrix approximation. No algorithmic result was presented, except for an exhaustive algorithm that ran in $\Omega(n^k)$ time.

To the best of our knowledge, the first nontrivial *algorithmic result* for relative-error low-rank matrix approximation was provided by a preliminary version of this paper [8]. In particular, an earlier version of Theorem 1 provided the first known relative-error column-based low-rank approximation in polynomial time [8]. The major difference between our Theorem 1 and our result in [8] is that the sampling probabilities in [8] are more complicated. The algorithm of [8] runs in $O(SVD(A_k))$ time (although it was originally reported to run in $O(SVD(A))$ time), and it has a sampling complexity of $O(k^2 \log(1/\delta)/\epsilon^2)$ columns.

Subsequent to the completion of the preliminary version of this paper [8], several developments have been made on relative-error low-rank matrix approximation algorithms. First, Har-Peled reported an algorithm that in roughly $O(mnk^2 \log k)$ time returns as output a rank-$k$ matrix $A'$ with a relative-error approximation guarantee [13]. His algorithm uses geometric ideas and involves sampling and merging approximately-optimal $k$-flats; it is not clear if this approximation can be expressed in terms of a small number of columns of $A$. Then, Deshpande and Vempala [3] reported an algorithm that also returns a relative-error approximation guarantee. Their algorithm extends ideas from [19, 2] and it leads to a CX matrix approximation consisting of $O(k \log k)$ columns of $A$. The complexity of their algorithm is $O(Mk^2 \log k)$, where $M$ is the number of nonzero elements of $A$, and their algorithm can be implemented with $O(k \log k)$ passes over the data. In light of these developments, we simplified and generalized our preliminary results [8], and we performed a more refined analysis to improve our sampling complexity to $O(k \log k)$.

## 2 Proof of Theorem 1

Regardless of whether the columns are chosen with the Exactly($c$) algorithm or Expected($c$) algorithm, we can construct a *column sampling matrix* $S$, such that $C = AS$. Similarly, we may introduce a *diagonal rescaling matrix* $D$ in this expression, which rescales each sampled column by $1/\sqrt{cp_j}$ for the Exactly($c$) algorithm and $1/\min\{1, \sqrt{cp_j}\}$ for the Expected($c$) algorithm. For details on this formalism, see [9]. Since scaling the columns of a matrix does not change the subspace spanned by its columns, $A - CC^+A = A - ASD\,(ASD)^+\,A$. Our careful choice for $S$ and $D$ will allow us to apply matrix perturbation results from [5, 21] to bound this latter expression. For simplicity, we assume that $\epsilon \in (0, 1]$.

## 2.1 Constructing $C$ with the Exactly($c$) Algorithm

The first claim of Theorem 1 considers the situation when the columns of $A$ are sampled with the Exactly($c$) algorithm. In this subsection, we provide its proof. The proof of the second claim is similar, and we outline the differences in the next subsection.

To prove our main result, we must "disentangle" the "top" singular subspace of $A$ from the "bottom" singular subspace. To do so, first note that using the unitary invariance of the Frobenius norm, and since $\left(U_A \Sigma_A V_A^T SD\right)^+ = \left(\Sigma_A V_A^T SD\right)^+ U_A^T$, it follows that

$$\left\|A - CC^+A\right\|_F^2 = \left\|\Sigma_A - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+ \Sigma_A\right\|_F^2 \tag{4}$$

$$= \left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+ \begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}\right\|_F^2$$

$$+ \left\|\begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+ \begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix}\right\|_F^2 . \tag{5}$$

Next, to upper bound the second term on the right hand side of (5), recall that since $I - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+$ is a projector matrix, it may be dropped without increasing a unitarily invariant norm, and thus

$$\left\|\left(I - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+\right)\begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix}\right\|_F^2 \leq \left\|A - A_k\right\|_F^2 . \tag{6}$$

Finally, to establish the first claim of Theorem 1, we seek to upper bound the first term on the right hand side of (5) by $\epsilon \left\|A - A_k\right\|_F^2$. That is, we seek an upper bound that does not depend at all on *any* of the top $k$ singular values of $A$. To this end, note that

$$\left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+ \begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}\right\|_F^2$$

$$= \min_{X \in \mathbb{R}^{c \times k}} \left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right) X\right\|_F^2 \tag{7}$$

$$\leq \left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_k V_k^T SD\right)^+ \Sigma_k\right\|_F^2 . \tag{8}$$

Equations (7) and (8) follow from least-squares approximation theory: (7) follows since $\left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^+ \begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}$ is the exact projection of the matrix $\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}$ on the subspace spanned by the columns of $\Sigma_A V_A^T SD$; and (8) follows since $X = \left(\Sigma_k V_k^T SD\right)^+ \Sigma_k \in \mathbb{R}^{c \times k}$ is a suboptimal – but as we will see below very convenient – choice for $X$ in (7).

To see that (8) provides the bound we seek, let the rank of the $k \times c$ matrix $V_k^T SD$ be $\tilde{k}$, and let its SVD be $V_k^T SD = U_{V_k^T SD} \Sigma_{V_k^T SD} V_{V_k^T SD}^T$. Clearly $\tilde{k} \leq k$. Among other things, the following lemma states that, given our construction of $S$ and $D$, all the singular values of $V_k^T SD$ are close to 1 and thus that the rank of $V_k^T SD$ is equal to $k$.

**Lemma 1.** *If $c \geq 40k^2/\beta\epsilon^2$, then with probability at least $0.9$:*

- $\tilde{k} = k$, *i.e.*, $rank(V_k^T SD) = rank(V_k)$,
- $\left\| \left(V_k^T SD\right)^+ - \left(V_k^T SD\right)^T \right\|_2 = \left\| \Sigma_{V_k^T SD}^{-1} - \Sigma_{V_k^T SD} \right\|_2$,
- $\left(\Sigma_k V_k^T SD\right)^+ = \left(V_k^T SD\right)^+ \Sigma_k^{-1}$, *and*
- $\left\| \Sigma_{V_k^T SD} - \Sigma_{V_k^T SD}^{-1} \right\|_2 \leq \epsilon/\sqrt{2}$.

*Proof:* Note that for all $i \in [\tilde{k}]$,

$$\begin{aligned} \left|1 - \sigma_i^2\left(V_k^T SD\right)\right| &= \left|\sigma_i\left(V_k^T V_k\right) - \sigma_i\left(V_k^T SDDS^T V_k\right)\right| \\ &\leq \left\|V_k^T V_k - V_k^T SDDS^T V_k\right\|_2 . \end{aligned} \tag{9}$$

Since the probabilities of (2) satisfy the condition of Theorem 1 of [5]

$$\mathbf{E}\left[\left\|V_k^T V_k - V_k^T SDDS^T V_k\right\|_F^2\right] \leq \frac{1}{\beta c}\left\|V_k\right\|_F^4 = \frac{k^2}{\beta c}, \tag{10}$$

where the equality follows since $\|V_k\|_F^2 = k$. By applying Markov's inequality to (10), taking square roots of both sides, combining it with (9), and using $\|\cdot\|_2 \leq \|\cdot\|_F$ and the assumed choice of $c$, it follows that $\left|1 - \sigma_i^2\left(V_k^T SD\right)\right| \leq \epsilon/2 \leq 1/2$, since $\epsilon \leq 1$. This implies that all singular values of $V_k^T SD$ are strictly positive, and thus that $\tilde{k} = k$. The remainder of the proof is similar to that of Lemma 4.1 of [9].

◇

Using Lemma 1, we manipulate the right hand side of (8) as follows:

$$\begin{aligned} &\left\| \begin{bmatrix} \Sigma_k \\ \mathbf{0} \end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_k V_k^T SD\right)^+ \Sigma_k \right\|_F^2 \\ &\quad = \left\| \begin{bmatrix} \Sigma_k \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \Sigma_k & \mathbf{0} \\ \mathbf{0} & \Sigma_{\rho-k} \end{bmatrix} \begin{bmatrix} V_k^T \\ V_{\rho-k}^T \end{bmatrix} SD\left(V_k^T SD\right)^+ \right\|_F^2 \\ &\quad = \left\| \begin{bmatrix} \Sigma_k \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \Sigma_k V_k^T \\ \Sigma_{\rho-k} V_{\rho-k}^T \end{bmatrix} SD\left(V_k^T SD\right)^+ \right\|_F^2 \\ &\quad = \left\| \Sigma_k - \Sigma_k \underbrace{V_k^T SD\left(V_k^T SD\right)^+}_{=I_k} \right\|_F^2 + \left\| \Sigma_{\rho-k} V_{\rho-k}^T SD\left(V_k^T SD\right)^+ \right\|_F^2 \qquad (11) \\ &\quad = \left\| \Sigma_{\rho-k} V_{\rho-k}^T SD\left(V_k^T SD\right)^+ \right\|_F^2 . \qquad (12) \end{aligned}$$

The first term of (11) is the most important point of the proof. The sampling probabilities $\{p_i\}$ are carefully constructed to guarantee that the $k \times c$ matrix $V_k^T SD$ has full rank; thus its columns – which are $k$-dimensional vectors – span $\mathbb{R}^k$. As a result, the projection of $\Sigma_k$ on the subspace spanned by the columns of $V_k^T SD$ is equal to $\Sigma_k$. Thus, since $\Sigma_k$ does not appear in (12), at this point in the proof, we have removed any dependency of the error on the top $k$ singular values of $A$.

We can combine (5), (6), (8), and (12), and take the square root of both sides to get

$$\left\|A - CC^+A\right\|_F \leq \|A - A_k\|_F + \left\|\Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+\right\|_F . \tag{13}$$

From this, the triangle inequality, and the fact that for any two matrices $A$ and $B$, $\|AB\|_F \leq \|B\|_2 \|A\|_F$, we have that

$$\begin{aligned} &\left\|\Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+\right\|_F \\ &\quad\leq \left\|XSD\left(V_k^T SD\right)^T\right\|_F + \left\|XSD\left(\left(V_k^T SD\right)^+ - \left(V_k^T SD\right)^T\right)\right\|_F \\ &\quad\leq \left\|XSDDS^T V_k\right\|_F + \left\|\Sigma_{V_k^T SD}^{-1} - \Sigma_{V_k^T SD}\right\|_2 \|XSD\|_F , \end{aligned} \tag{14}$$

where we have let $X = \Sigma_{\rho-k} V_{\rho-k}^T$. The following lemma will be used to bound (14); the proof is omitted.

**Lemma 2.** *For any probabilities* $\{p_i\}$, $\left\|\Sigma_{\rho-k} V_{\rho-k}^T SD\right\|_F \leq 10 \|A - A_k\|_F$, *with probability at least* 0.9.

The following lemma will also be used to bound (14).

**Lemma 3.** *If* $c \geq 10k/\beta\epsilon^2$, *then* $\left\|\Sigma_{\rho-k} V_{\rho-k}^T SDDS^T V_k\right\|_F \leq \epsilon \|A - A_k\|_F$, *with probability at least* 0.9.

*Proof:* Note that $\Sigma_{\rho-k} V_{\rho-k}^T V_k = \mathbf{0}$, and we will view $\Sigma_{\rho-k} V_{\rho-k}^T SDDS^T V_k$ as approximating this matrix product. We apply Lemma 4 of [5] (see also Figure 5 of [5]) to get

$$\begin{aligned} \mathbf{E}\left[\left\|\Sigma_{\rho-k} V_{\rho-k}^T SDDS^T V_k - \Sigma_{\rho-k} V_{\rho-k}^T V_k\right\|_F^2\right] &\leq \frac{1}{\beta c} \|A - A_k\|_F^2 \|V_k\|_F^2 \\ &= \frac{k}{\beta c} \|A - A_k\|_F^2 . \end{aligned}$$

The lemma follows by applying Markov's inequality and taking the square roots of both sides of the resulting inequality.

◇

If $c \geq 40k^2/\beta\epsilon^2$, then Lemmas 1, 2, and 3 hold simultaneously with probability at least $1 - 3(0.1) = 0.7$. We condition on this event. Then, from (14), using Lemmas 1, 2, and 3, we get

$$\left\|\Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+\right\|_F \leq 9\epsilon \|A - A_k\|_F .$$

By combining this with (13), it follows that

$$\left\|A - CC^{+}A\right\|_F \leq (1 + 9\epsilon)\, \|A - A_k\|_F .$$

The first claim of Theorem 1 follows with probability at least 0.7 by letting $\epsilon' = \epsilon/9$ and adjusting $c$ to $O(k^2/\beta\epsilon'^2)$; it follows with probability at least $1-\delta$ by running $O(\log(1/\delta)$ trials and using standard boosint procedures.

Note that setting $c = O(k^2/\epsilon^2)$ was required by Lemma 1, but that Lemmas 2 and 3 hold with $c = O(k/\epsilon^2)$. In particular, (10) of Lemma 1 required setting $c = O(k^2/\epsilon^2)$ in order to bound the error by $\epsilon/2$. We conjecture that the same bound holds if $c = O(k \log k/\epsilon^2)$. This result would follow from a stronger spectral norm bound than that provided by the Frobenius norm bound of Theorem 1 of [5]. Instead, in the next section, we will reduce $c$ to $O(k \log k/\epsilon^2)$ by slightly modifying our sampling technique and using Theorem 3.1 of [21].

### 2.2 Constructing $C$ with the Expected($c$) Algorithm

The second claim of Theorem 1 considers the situation when the columns of $A$ are sampled with the Expected($c$) algorithm. In this subsection, we outline its proof.

If the columns of $A$ are sampled with the Expected($c$) algorithm, then the number of columns of $S$, and thus the number of rows and columns of $D$, is a random variable with expectation at most $c$. On the other hand, with this sampling procedure we can directly bound the spectral norm of (9), as opposed to bounding it indirectly via the Frobenius norm. To do so, consider the following theorem, which is a small extension of Theorem 3.1 in [21] to include the $\beta$ factor; see also [20].

**Theorem 2.** *Let $X \in \mathbb{R}^{m \times n}$ and let $c \leq n$ be a positive integer. If $S$ and $D$ are constructed with the* Expected($c$) *algorithm using sampling probabilities $p_i, i \in [n]$ such that $\sum_i p_i = 1$ and $p_i \geq \beta \left|X^{(i)}\right|^2 / \|X\|_F^2$, then*

$$\mathbf{E}\left[\left\|XX^T - XSDDS^TX^T\right\|_2\right] \leq O\left(\sqrt{\frac{\log c}{\beta c}}\right) \|X\|_F\, \|X\|_2 .$$

All of the derivations of Section 2.1 up to Lemma 1 hold for this modified sampling procedure. The following lemma is the analog of Lemma 1 with this new sampling prodecure, and it leads to an improved dependency of $c$ on $k$.

**Lemma 4.** ***(Analog of Lemma 1)*** *If $c = O\left(k \log k/\beta\epsilon^2\right)$, then each of the claims of Lemma 1 holds with probability at least* 0.9.

*Proof:* From Theorem 2 and since $\|V_k\|_F = \sqrt{k}$ and $\|V_k\|_2 = 1$, it follows that

$$\begin{aligned}\mathbf{E}\left[\left\|V_k^TV_k - V_k^TSDDS^TV_k\right\|_2\right] &\leq O\left(\sqrt{\log c/\beta c}\, \|V_k\|_F\, \|V_k\|_2\right) \\ &= O\left(\sqrt{k \log c/\beta c}\right).\end{aligned}$$

Using the assumed value of $c$, by Markov's inequality, and since $\epsilon \leq 1$, $\left|1 - \sigma_i^2\left(V_k^T SD\right)\right| \leq \epsilon/2 \leq 1/2$ with probability at least 0.9, which implies that $\tilde{k} = k$. The rest of the proof is the same as in Lemma 1.

$\diamond$

The remainder of the proof parallels the proof of Section 2.1.

## 3 Concluding Remarks

We conclude with three open problems.

- To what extent do the results of the present paper generalize to other matrix norms?
- What hardness results can be established for the optimal choice of columns?
- Does there exist a deterministic (any factor) approximation algorithm to the problem we consider?

**Acknowledgements.** We would like to thank Ravi Kannan for numerous useful discussions, Sariel Har-Peled for writing up his results amidst travel in India [13], and Amit Deshpande and Santosh Vempala for graciously providing a copy of [3].

## References

1. R. Bhatia. *Matrix Analysis.* Springer-Verlag, New York, 1997.
2. A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1117–1126, 2006.
3. A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. Technical Report TR06-042, Electronic Colloquium on Computational Complexity, March 2006.
4. 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.
5. P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *To appear in: SIAM Journal on Computing.*
6. P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *To appear in: SIAM Journal on Computing.*
7. P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *To appear in: SIAM Journal on Computing.*
8. P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Polynomial time algorithm for column-row based relative-error low-rank matrix approximation. Technical Report 2006-04, DIMACS, March 2006.
9. P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Sampling algorithms for $\ell_2$ regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1127–1136, 2006.

10. 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.
11. A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.
12. G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989.
13. S. Har-Peled. Low rank matrix approximation in linear time. *Manuscript. January 2006.*
14. R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.
15. F.G. Kuruvilla, P.J. Park, and S.L. Schreiber. Vector algebra in the analysis of genome-wide expression data. *Genome Biology*, 3:research0011.1–0011.11, 2002.
16. Z. Lin and R.B. Altman. Finding haplotype tagging SNPs by use of principal components analysis. *American Journal of Human Genetics*, 75:850–861, 2004.
17. M.Z. Nashed, editor. *Generalized Inverses and Applications.* Academic Press, New York, 1976.
18. P. Paschou, M.W. Mahoney, J.R. Kidd, A.J. Pakstis, S. Gu, K.K. Kidd, and P. Drineas. Intra- and inter-population genotype reconstruction from tagging SNPs. *Manuscript submitted for publication.*
19. L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via iterative sampling. Technical Report MIT-LCS-TR-983, Massachusetts Institute of Technology, Cambridge, MA, March 2005.
20. M. Rudelson and R. Vershynin. Approximation of matrices. *Manuscript.*
21. R. Vershynin. Coordinate restrictions of linear operators in $l_2^n$. *Manuscript.*