An Improved Approximation Algorithm for the Column Subset Selection Problem

Christos Boutsidis * Michael W. Mahoney † Petros Drineas ‡

Abstract

We consider the problem of selecting the "best" subset of *exactly* k *columns* from an $m \times n$ matrix A. In particular, we present and analyze a novel two-stage algorithm that runs in $O(\min\{mn^2, m^2n\})$ time and returns as output an $m \times k$ matrix C consisting of exactly k columns of A. In the first stage (the *randomized* stage), the algorithm randomly selects $O(k \log k)$ columns according to a judiciously-chosen probability distribution that depends on information in the top-k right singular subspace of A. In the second stage (the *deterministic* stage), the algorithm applies a deterministic column-selection procedure to select and return exactly k columns from the set of columns selected in the first stage. Let C be the $m \times k$ matrix containing those k columns, let P_C denote the projection matrix onto the span of those columns, and let A_k denote the "best" rank-k approximation to the matrix A as computed with the singular value decomposition. Then, we prove that

$$\|A - P_C A\|_2 \leq O\left(k^{\frac{3}{4}} \log^{\frac{1}{2}}(k)\,(n-k)^{\frac{1}{4}}\right) \|A - A_k\|_2\,,$$

with probability at least 0.7. This spectral norm bound improves upon the best previously-existing result (of Gu and Eisenstat [21]) for the spectral norm version of this Column Subset Selection Problem. We also prove that

$$\|A - P_C A\|_F \leq O\left(k\sqrt{\log k}\right) \|A - A_k\|_F\,,$$

with the same probability. This Frobenius norm bound is only a factor of $\sqrt{k \log k}$ worse than the best previously existing existential result and is roughly $O(\sqrt{k!})$ better than the best previous algorithmic result (both of Deshpande et al. [11]) for the Frobenius norm version of this Column Subset Selection Problem.

*Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, boutsc@cs.rpi.edu.

†Department of Mathematics, Stanford University, Stanford, CA, mmahoney@cs.stanford.edu.

‡Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, drinep@cs.rpi.edu.

1 Introduction

We consider the problem of selecting the "best" set of *exactly* k *columns* from an $m \times n$ matrix A. More precisely, we consider the following Column Subset Selection Problem (CSSP):

Definition 1. ***(The CSSP)*** *Given a matrix $A \in \mathbb{R}^{m \times n}$ and a positive integer k, pick k columns of A forming a matrix $C \in \mathbb{R}^{m \times k}$ such that the residual*

$$\|A - P_C A\|_\xi$$

is minimized over all possible $\binom{n}{k}$ choices for the matrix C. Here, $P_C = CC^+$ denotes the projection onto the k-dimensional space spanned by the columns of C and $\xi = 2$ or F denotes the spectral norm or Frobenius norm.

That is, the goal of the CSSP is to find a subset of exactly k columns of A that "captures" as much of A as possible, with respect to the spectral norm and/or Frobenius norm, in a projection sense. The CSSP has been studied extensively in numerical linear algebra, where it has found applications in, e.g., scientific computing [6]. More recently, a relaxation has been studied in theoretical computer science, where it has been motivated by applications to large scientific and internet data sets [15].

1.1 Complexity of the CSSP We briefly comment on the complexity of the problem. Clearly, in $O(n^k)$ time we can generate all possible matrices C and thus solve the problem exactly, i.e., find the optimal solution. However, from a practical perspective, in data analysis applications of the CSSP (see Section 1.2), n is often in the order of hundreds or thousands. Thus, in practice, algorithms that run in $O(n^k)$ time are prohibitively slow even if k is, from a theoretical perspective, a constant. Finally, the NP-hardness of the CSSP is an interesting open problem. Note, though, that a similar problem asking for the k columns of the $m \times n$ matrix A that maximize the volume of the parallelepiped spanned by the columns of C, is provably NP-hard [10].

Existential Result				Algorithmic Result			Our Result	
	Ref	p(k,n)	Time	Ref	p(k,n)	Time	p(k,n)	Time
$\xi = 2$	[22]	$\sqrt{k(n-k)+1}$	$O(n^k)$	[21]	$O(k^{\frac{1}{2}}(n-k)^{\frac{1}{2}})$	$O(mn^2)$	$O(k^{\frac{3}{4}}(n-k)^{\frac{1}{4}})$	$O(mn^2)$
$\xi = F$	[12]	$\sqrt{k+1}$	$O(n^k)$	[12]	$\sqrt{(k+1)!}$	$O(mnk)$	$O(k\sqrt{\log k})$	$O(mn^2)$

Table 1: Comparison of the results of this paper with the state-of-the-art existential and algorithmic results for the CSSP. Here, $p(k,n)$ is involved in the approximation $\|A - P_C A\|_\xi \leq p(k,n) \|A - A_k\|_\xi$. (In addition, $m \geq n$ for this table.)

1.2 The CSSP in statistical data analysis In data applications, where the input matrix A models m objects represented with respect to n features, the CSSP corresponds to unsupervised feature selection. Standard motivations for feature selection include facilitating data visualization, reducing training times, avoiding overfitting, and facilitating data understanding.

Consider, in particular, Principal Components Analysis (PCA), which is the predominant linear dimensionality reduction technique, and which has been widely applied on datasets in all scientific domains, from the social sciences and economics, to biology and chemistry. In words, PCA seeks to map or embed data points from a high dimensional Euclidean space to a low dimensional Euclidean space while keeping all the relevant linear structure intact. PCA is an unsupervised dimensionality reduction technique, with the sole input parameters being the coordinates of the data points and the number of dimensions that will be retained in the embedding (say k), which is typically a constant independent of m and n; often it is $k \ll \{m, n\}$ too. Data analysts often seek a subset of k actual features (that is, k actual columns, as opposed to the k eigenvectors or eigenfeatures returned by PCA) that can accurately reproduce the structure derived by PCA. The CSSP is the obvious optimization problem associated with such unsupervised feature selection tasks.

We should note that similar formulations appeared in [23, 33, 36, 37, 26, 1]. In addition, applications of such ideas include: (*i*) [34], where a "compact CUR matrix decomposition" was applied to static and dynamic data analysis in large sparse graphs; (*ii*) [25, 24, 13], where these ideas were used for compression and classification of hyperspectral medical data and the reconstruction of missing entries from recommendation systems data in order to make high-quality recommendations; and (*iii*) [30], where the concept of "PCA-correlated SNPs" (Single Nucleotide Polymorphisms) was introduced and applied to classify individuals from throughout the world without the need for any prior ancestry information. Finally, see the full version of this paper [4] for an empirical evaluation of our main algorithm; and see [3] for a detailed evaluation of our main algorithm as an unsupervised feature selection strategy in three application domains of modern statistical data analysis (finance, document-term data, and genetics).

1.3 Our main results We present a novel two-stage algorithm for the CSSP. This algorithm is presented in detail in Section 3 as Algorithm 1. In the first stage of this algorithm (the *randomized stage*), we randomly select $O(k \log k)$ columns of V_k^T, i.e., of the transpose of the $n \times k$ matrix consisting of the top k right singular vectors of A, according to a judiciously-chosen probability distribution that depends on information in the top-k right singular subspace of A. Then, in the second stage (the *deterministic stage*), we apply a deterministic column-selection procedure to select exactly k columns from the set of columns of V_k^T selected by the first stage. The algorithm then returns the corresponding k columns of A. In Section 4 we prove the following theorem.

THEOREM 1. *There exists an algorithm (the two-stage Algorithm 1) that approximates the solution to the CSSP. This algorithm takes as input an $m \times n$ matrix A of rank $\rho \leq \min\{m, n\}$ and a positive integer k; it runs in $O(\min\{mn^2, m^2 n\})$ time; and it returns as output an $m \times k$ matrix C consisting of exactly k columns of A such that with probability at least 0.7:*

$$\|A - P_C A\|_2 \leq O\left(k^{3/4} \log^{1/2}(k) (\rho - k)^{1/4}\right) \|A - A_k\|_2,$$

$$\|A - P_C A\|_F \leq O\left(k \log^{1/2} k\right) \|A - A_k\|_F.$$

Here, $P_C = CC^+$ denotes a projection onto the column span of the matrix C, and A_k denotes the best rank-k approximation to the matrix A as computed with the singular value decomposition (SVD).

Note that we can trivially boost the success probability in the above theorem to $1-\delta$ by repeating the algorithm $O(\log(1/\delta))$ times. Note also that the running time of our algorithm is linear in the larger of the dimensions m and n, quadratic in the smaller one, and independent of k. Thus, it is practically useful and efficient.

To put our results into perspective, we compare them to the best existing results for CSSP. Prior work provided bounds of the form

$$\|A - P_C A\|_\xi \leq p(k,n) \|A - A_k\|_\xi , \quad (1.1)$$

where $p(k,n)$ is a polynomial on n and k. For $\xi = 2$, i.e., for the spectral norm, the best previously-known bound for approximating the CSSP is $p(k,n) = O\left(\sqrt{1+k(n-k)}\right)$ [21], while for $\xi = F$, i.e., for the Frobenius norm, the best bound is $p(k,n) = \sqrt{(k+1)!}$ [11]. Both results are algorithmically efficient, running in time polynomial in all three parameters m, n, and k; the former runs in $O(\min\{mn^2, m^2n\})$ time and the latter runs in $O(mnk+kn)$ time. Thus, our approach asymptotically improves the best previously-known result for the spectral norm version of the CSSP by a factor of $n^{1/4}$. (Here we assume that k is independent of n, as is typically the case in data applications of these techniques.) Our approach also provides an algorithmic bound for the Frobenius norm version of the CSSP that is roughly $O(\sqrt{k!})$ better than the best previously-known algorithmic result. It should be noted that [11] also proves that by exhaustively testing all $\binom{n}{k}$ possibilities for the matrix C, the best one will satisfy eqn. (1.1) with $p(k,n) = \sqrt{k+1}$. Our algorithmic result is only $O(\sqrt{k \log k})$ worse that this existential result. A similar existential result for the spectral norm version of the CSSP is proved in [22] with $p(k,n) = \sqrt{1+k(n-k)}$. Obviously, the result in [11] is a lower bound for the Frobenius norm version of the CSSP. On the other hand, the result in [22] is the best existing existential result for the spectral norm version of the CSSP. A lower bound for the spectral norm version of the CSSP, which should be between $||A - A_k||_2$ and $\sqrt{1+k(n-k)}||A - A_k||_2$, is an interesting open problem. These results are summarized in Table 1.

Finally, we should emphasize that a novel feature of the algorithm that we present in this paper is that it combines in a nontrivial manner recent algorithmic developments in the theoretical computer science community with more traditional techniques from the numerical linear algebra community in order to obtain improved bounds for the CSSP.

2 Background and prior work

2.1 Notation and 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. In addition, let $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$ denote the square of its Frobenius norm, and let $\|A\|_2 = \sup_{x \in \mathbb{R}^n,\ x \neq 0} |Ax|_2 / |x|_2$ denote its spectral norm. If $A \in \mathbb{R}^{m \times n}$, then the Singular Value Decomposition (SVD) of A can be written as

$$\begin{aligned} A &= U_A \Sigma_A V_A^T \\ &= \begin{pmatrix} U_k & U_{\rho-k} \end{pmatrix} \begin{pmatrix} \Sigma_k & \mathbf{0} \\ \mathbf{0} & \Sigma_{\rho-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ V_{\rho-k}^T \end{pmatrix}. \end{aligned}$$

In this expression, $\rho \leq \min\{m,n\}$ denotes the rank of A, $U_A \in \mathbb{R}^{m \times \rho}$ is an orthonormal matrix, Σ_A is a $\rho \times \rho$ diagonal matrix, and $V_A \in \mathbb{R}^{n \times \rho}$ is an orthonormal matrix. Also, Σ_k denotes the $k \times k$ diagonal matrix containing the top k singular values of A, $\Sigma_{\rho-k}$ denotes the $(\rho-k) \times (\rho-k)$ matrix containing the bottom $\rho-k$ singular values of A, V_k denotes the $n \times k$ matrix whose columns are the top k right singular vectors of A, and $V_{\rho-k}$ denotes the $n \times (\rho-k)$ matrix whose columns are the bottom $\rho-k$ right singular vectors of A, etc.

The $m \times k$ orthogonal matrix U_k consisting of the top k left singular vectors of A is the "best" set of k linear combinations of the columns of A, in the sense that $A_k = P_{U_k} A = U_k \Sigma_k V_k^T$ is the "best" rank k approximation to A. Here, $P_{U_k} = U_k U_k^T$ is a projection onto the k-dimensional space spanned by the columns of U_k. In particular, A_k minimizes $\|A - A'\|_\xi$, for both $\xi = 2$ and F, over all $m \times n$ matrices A' whose rank is at most k. We will use the notation $\|\cdot\|_\xi$ when writing an expression that holds for both the spectral and the Frobenius norm. We will subscript the norm by 2 and F when writing expressions that hold for one norm or the other. Finally, the Moore-Penrose generalized inverse, or pseudoinverse, of A, denoted by A^+, may be expressed in terms of the SVD as $A^+ = V_A \Sigma_A^{-1} U_A^T$.

2.2 Related prior work Since solving the CSSP exactly is a hard combinatorial optimization problem, research has historically focused on computing approximate solutions to it. Since $\|A - A_k\|_\xi$ provides an immediate lower bound for $\|A - P_C A\|_\xi$, for $\xi = 2, F$ and for any choice of C, a large number of approximation algorithms have been proposed to select a subset of k columns of A such that the resulting matrix C satisfies

$$\|A - A_k\|_\xi \leq \|A - P_C A\|_\xi \leq p(k,n) \|A - A_k\|_\xi$$

for some function $p(k,n)$. Within the numerical linear algebra community, most of the work on the CSSP has focused on spectral norm bounds and is related to the so-called Rank Revealing QR (RRQR) factorization:

Definition 2. ***(The RRQR factorization)*** *Given a matrix $A \in R^{m \times n}$ ($m \geq n$) and an integer k ($k \leq n$), assume partial QR factorizations of the form:*

$$A\Pi = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

Method	Reference		p(k,n)	Time
Pivoted QR	[Golub, 1965]	[19]	$\sqrt{(n-k)}2^k$	$O(mnk)$
High RRQR	[Foster, 1986]	[16]	$\sqrt{n(n-k)}2^{n-k}$	$O(mn^2)$
High RRQR	[Chan, 1987]	[5]	$\sqrt{n(n-k)}2^{n-k}$	$O(mn^2)$
RRQR	[Hong and Pan, 1992]	[22]	$\sqrt{k(n-k)+k}$	$O(n^k)$
Low RRQR	[Chan and Hansen, 1994]	[7]	$\sqrt{(k+1)n}2^{k+1}$	$O(mn^2)$
Hybrid-I RRQR	[Chandrasekaran and Ipsen, 1994]	[8]	$\sqrt{(k+1)(n-k)}$	$O(n^k)$
Hybrid-II RRQR		[8]	$\sqrt{(k+1)(n-k)}$	$O(n^k)$
Hybrid-III RRQR		[8]	$\sqrt{(k+1)(n-k)}$	$O(n^k)$
Strong RRQR	[Gu and Eisenstat, 1996]	[21]	$\sqrt{k(n-k)+1}$	$O(n^k)$
Strong RRQR		[21]	$O(\sqrt{k(n-k)+1})$	$O(mn^2)$
DGEQPY	[Bischof and Orti, 1998]	[2]	$O(\sqrt{(k+1)^2(n-k)})$	-
DGEQPX		[2]	$O(\sqrt{(k+1)(n-k)})$	$O(n^k)$
SPQR	[Stewart, 1999]	[32]	-	-
PT Algorithm 1	[Pan and Tang, 1999]	[29]	$O(\sqrt{(k+1)(n-k)})$	-
PT Algorithm 2		[29]	$O(\sqrt{(k+1)^2(n-k)})$	-
PT Algorithm 3		[29]	$O(\sqrt{(k+1)^2(n-k)})$	-
Pan Algorithm 2	[Pan, 2000]	[28]	$O(\sqrt{k(n-k)+1})$	$O(mn^2)$

Table 2: Accuracy of deterministic algorithms for the CSSP. A dash means that the algorithm either runs in $O(n^k)$ time, or the authors do not provide a running time bound. (In addition, $m \geq n$ for this table.)

where $Q \in R^{m\times n}$ is an orthonormal matrix, $R \in R^{n\times n}$ is upper triangular, $R_{11} \in R^{k\times k}$, $R_{12} \in R^{k\times(n-k)}$, $R_{22} \in R^{(n-k)\times(n-k)}$, and $\Pi \in R^{n\times n}$ is a permutation matrix. The above factorization is called a RRQR factorization if it satisfies

$$\begin{aligned} \frac{\sigma_k(A)}{p_1(k,n)} \leq &\quad \sigma_{min}(R_{11}) \quad \leq \sigma_k(A) \\ \sigma_{k+1}(A) \leq &\quad \sigma_{max}(R_{22}) \quad \leq p_2(k,n)\sigma_{k+1}(A), \end{aligned}$$

where $p_1(k,n)$ and $p_2(k,n)$ are functions bounded by low degree polynomials in k and n.

The work of Golub on pivoted QR factorizations [19] was followed by much research addressing the problem of constructing an efficient RRQR factorization. Most researchers improved RRQR factorizations by focusing on improving the functions $p_1(k,n)$ and $p_2(k,n)$ in Definition 2. Let Π_k denote the first k columns of a permutation matrix Π. Then, if $C = A\Pi_k$ is an $m \times k$ matrix consisting of k columns of A, it is straightforward to prove that

$$\|A - P_C A\|_\xi = \|R_{22}\|_\xi ,$$

for both $\xi = 2, F$. Thus, in particular, when applied to the spectral norm, it follows that

$$\|A - P_C A\|_2 \leq p_2(k,n)\sigma_{k+1}(A) = p_2(k,n)\|A - A_k\|_2 ,$$

i.e., any algorithm that constructs an RRQR factorization of the matrix A with provable guarantees also provides provable guarantees for the CSSP. See Table 2 for a summary of existing results, and see [17] for a survey and an empirical evaluation of some of these algorithms. More recently, [27, 35] proposed random-projection type algorithms that achieve the same spectral norm bounds as prior work while improving the running time.

Within the theoretical computer science community, much work has followed that of Frieze, Kannan, and Vempala [18] on selecting a small subset of representative columns of A, forming a matrix C, such that the projection of A on the subspace spanned by the columns of C is as close to A as possible. The algorithms from this community are randomized, which means that they come with a failure probability, and focus mainly on the Frobenius norm. It is worth noting that they provide a strong tradeoff between the number of selected columns and the desired approximation accuracy. A typical scenario for these algorithms is that the desired approximation error (see ϵ below) is given as input, and then the algorithm selects the minimum number of appropriate columns in order to achieve this error. One of the most relevant results for this paper is a bound of [11], which states that there exist exactly k columns in any $m \times n$ matrix A such that

$$\left\|A - CC^+A\right\|_F \leq \sqrt{k+1}\,\|A - A_k\|_F .$$

Here, C contains exactly k columns of A. The only known algorithm to find these k columns is to try all $\binom{n}{k}$ choices and keep the best. This existential result relies on the so-called volume sampling method [11, 12]. In [12], an adaptive sampling method is used to approximate the volume sampling method and leads to an $O(mnk + kn)$ algorithm which finds k columns of A such that

$$\left\|A - CC^+A\right\|_F \leq \sqrt{(k+1)!}\left\|A - A_k\right\|_F.$$

As mentioned above, much work has also considered algorithms choosing slightly more than k columns. This relaxation provides significant flexibility and improved error bounds. For example, in [12], an adaptive sampling method leads to an $O\left(mn\left(k/\epsilon^2 + k^2\log k\right)\right)$ algorithm, such that

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

holds with high probability for some matrix C consisting of $O\left(k/\epsilon^2 + k^2\log k\right)$ columns of A. Similarly, in [14, 15], Drineas, Mahoney, and Muthukrishnan leverage the subspace sampling method to give an $O(\min\{mn^2, m^2n\})$ algorithm such that

$$\left\|A - CC^+A\right\|_F \leq (1+\epsilon)\left\|A - A_k\right\|_F \tag{2.2}$$

holds with high probability if C contains at most $O(k\log k/\epsilon^2)$ columns of A.

3 A two-stage algorithm for the CSSP

In this section, we present and describe Algorithm 1, our main algorithm for approximating the solution to the CSSP. This algorithm takes as input an $m \times n$ matrix A and a rank parameter k. After an initial setup, the algorithm has two stages: a randomized stage and a deterministic stage. In the *randomized stage*, a randomized procedure is run to select $O(k\log k)$ columns from the $k \times n$ matrix V_k^T, i.e., the transpose of the matrix containing the top-k right singular vectors of A. The columns are chosen by randomly sampling according to a judiciously-chosen nonuniform probability distribution that depends on information in the top-k right singular subspace of A. Then, in the *deterministic stage*, a deterministic procedure is employed to select exactly k columns from the $O(k\log k)$ columns chosen in the randomized stage. The algorithm then outputs exactly k columns of A that correspond to those columns chosen from V_k^T. Theorem 1 states that the projection of A on the subspace spanned by these k columns of A is (up to bounded error) close to the best rank k approximation to A.

3.1 Detailed description of our main algorithm

In more detail, Algorithm 1 first computes a probability distribution $p_1, p_2, \ldots, p_n$ over the set $\{1, \ldots, n\}$, i.e., over the columns of V_k^T, or equivalently over the columns of A. The probability distribution depends on information in the top-k right singular subspace of A. In particular, for all $i \in [n]$, define

(3.3)

$$p_i = \frac{\frac{1}{2}\left\|(V_k)_{(i)}\right\|_2^2}{\sum_{j=1}^n \left\|(V_k)_{(j)}\right\|_2^2} + \frac{\frac{1}{2}\left\|\left(\Sigma_{\rho-k}V_{\rho-k}^T\right)^{(i)}\right\|_2^2}{\sum_{j=1}^n \left\|\left(\Sigma_{\rho-k}V_{\rho-k}^T\right)^{(j)}\right\|_2^2},$$

and note that $p_i \geq 0$, for all $i \in [n]$, and that $\sum_{i=1}^n p_i = 1$. We will describe the computation of probabilities of this form below.

In the *randomized stage*, Algorithm 1 employs the following randomized column selection algorithm to choose $O(k\log k)$ columns from V_k^T to pass to the second stage. Let $c = \Theta(k\log k)$ be a positive integer. For each $i \in [n]$, independently, the algorithm keeps the i-th column of V_k^T with probability $\min\{1, cp_i\}$. Additionally, if the i-th column is kept, then a scaling factor equal to $1/\sqrt{\min\{1, cp_i\}}$ is kept as well. Thus, at the end of this process, we will be left with $\tilde{c}$ columns of V_k^T and their corresponding scaling factors. Notice that due to random sampling, $\tilde{c}$ will generally be different than c. However, it can be proved that if $c = \Theta(k\log k)$ then $\tilde{c} = O(k\log k)$ with constant probability.

In order to conveniently represent the $\tilde{c}$ selected columns and the associated scaling factors, we will use the following sampling matrix formalism. First, define an $n \times \tilde{c}$ sampling matrix S_1 as follows: S_1 is initially empty; for all i, in turn, if the i-th column of V_k^T is selected by the random sampling process, then e_i (an n-vector of all-zeros, except for its i-th entry which is set to one) is appended to S_1. Next, define the $\tilde{c} \times \tilde{c}$ diagonal rescaling matrix D_1 as follows: if the i-th column of V_k^T is selected, then a diagonal entry of D_1 is set to $1/\sqrt{\min\{1, cp_i\}}$. Thus, we may view the randomized stage as outputting the matrix $V_k^T S_1 D_1$ consisting of a small number of rescaled columns of V_k^T, or simply as outputting S_1 and D_1.

In the *deterministic stage*, Algorithm 1 employs a deterministic column selection algorithm to the output of the first stage in order to choose *exactly k columns* from the input matrix A. To do so, we run the Algorithm 1 of [28] on the $k \times \tilde{c}$ matrix $V_k^T S_1 D_1$, i.e., the column-scaled version of the columns of V_k^T chosen in the first stage.[1] Thus, a matrix $V_k S_1 D_1 S_2$ is formed, or equivalently, in the sampling matrix

[1] Most deterministic algorithms for the CSSP operate on

formalism described previously, a new matrix S_2 is constructed. Its dimensions are $\tilde{c} \times k$, since it selects exactly k columns out of the $\tilde{c}$ columns returned after the end of the randomized stage. The algorithm then returns the corresponding k columns of the original matrix A, i.e., after the second stage of the algorithm is complete, the $m \times k$ matrix $C = AS_1S_2$ is returned as the final output.

Input: $m \times n$ matrix A, integer k.
Output: $m \times k$ matrix C with k columns of A.

1. **Initial setup:**
   - Compute the top k right singular vectors of A, denoted by V_k.
   - Compute the sampling probabilities p_i, for $i \in [n]$, using eqn. (3.3) or (3.4).
   - Let $c = \Theta(k \log k)$.
2. **Randomized Stage:**
   - For $i = 1, \ldots, n$, keep the i-th index with probability $\min\{1, cp_i\}$. If the i-th index is kept, keep the scaling factor $\sqrt{\min\{1, cp_i\}}$.
   - Form the sampling matrix S_1 and the rescaling matrix D_1 (see text).
3. **Deterministic Stage:**
   - Run Algorithm 1 of Pan [28] (see also Lemma 3.5 in [28]) on the matrix $V_k^T S_1 D_1$ in order to select exactly k columns of $V_k^T S_1 D_1$, thereby forming the sampling matrix S_2 (see text).
   - Return the corresponding k columns of A, i.e., return $C = AS_1S_2$.

Algorithm 1: A two-stage algorithm for the CSSP.

3.2 Running time analysis We now discuss the running time of our algorithm. Note that manipulating the probability distribution (3.3) yields:

$$p_i = \frac{\left\|(V_k)_{(i)}\right\|_2^2}{2k} + \frac{\left\|(A)^{(i)}\right\|_2^2 - \left\|\left(AV_kV_k^T\right)^{(i)}\right\|_2^2}{2\left(\|A\|_F^2 - \left\|AV_kV_k^T\right\|_F^2\right)}. \tag{3.4}$$

Thus, knowledge of V_k, i.e., the $n \times k$ matrix consisting of the top-k right singular vectors of A, suffices to compute the p_i's.[2] By (3.4), $O(\min\{mn^2, m^2n\})$ time suffices for our theoretical analysis; in practice, of course, Lanczos/Arnoldi algorithms could be used to speed up the algorithm. Note also that in order to obtain a Frobenius norm bound of the form in Theorem 1, our theoretical analysis holds if the sampling probabilities are of the form:

$$p_i = \left\|(V_k)_{(i)}\right\|_2^2 / k. \tag{3.5}$$

That is, the Frobenius norm bound of Theorem 1 holds even if the second term in the sampling probabilities of (3.3) or (3.4) is omitted. Finally, the deterministic stage of our algorithm (using Algorithm 1 of [28]) takes $O(k^3 \log k)$ time, since $V_k^T S_1 D_1$ has w.h.p. $O(k \log k)$ columns.

An interesting open problem would be to identify other suitable importance sampling probability distributions that avoid the computation of a basis for the top-k right singular subspace.

3.3 Intuition underlying our main algorithm Intuitively, we achieve improved bounds for the CSSP because we apply the deterministic algorithm to a lower dimensional matrix (the matrix $V_k^T S_1 D_1$ with $O(k \log k)$ columns, as opposed to the matrix A with n columns) in which the columns are "spread out" in a "nice" manner. To see this, note that the probability distribution of equation (3.5), and thus one of the two terms in the probability distribution of (3.3) or (3.4), equals (up to scaling) the diagonal elements of the projection matrix onto the span of the top-k right singular subspace. In diagnostic regression analysis, these quantities have a natural interpretation in terms of *statistical leverage*, and thus they have been used extensively to identify "outlying" data points [9]. Thus, the importance sampling probabilities that we employ in the randomized stage of our main algorithm provide a bias toward more "outlying" columns, which then provide a "nice" starting point for the deterministic stage of our main algorithm. (This also provides intuition as to why using importance sampling probabilities of the form (3.5) leads to relative-error low-rank matrix approximation bounds of the form (2.2); see [14, 15].)

matrices that are $m \times n$ with $m \geq n$. In our case, in the second stage, we need to apply a deterministic column selection algorithm to a matrix with more columns than rows. Even though, to the best of our understanding, theoretical bounds for most of the algorithms reviewed in Section 2 hold even if $m < n$, for our theoretical analysis we opt to employ Algorithm 1 (and the related Lemma 3.5) of [28] which is explicitly designed to work for $m < n$.

[2] Actually, from (3.4) it is clear that *any* orthogonal matrix spanning the top-k right singular subspace suffices.

4 Proof of Theorem 1

We start with an outline of our proof, pointing out conceptual improvements that were necessary in order to obtain improved bounds. An important condition in the first phase of the algorithm is that when we sample columns from the $k \times n$ matrix V_k^T, we obtain a $k \times \tilde{c}$ matrix $V_k^T S_1 D_1$ that does not lose any rank. To do so, we will apply a result from matrix perturbation theory to prove that if $c = \Theta(k \log k)$ then $\left|\sigma_k^2\left(V_k^T S_1 D_1\right) - 1\right| \leq 1/2$. (See Lemma 4.1 below.) Then, under the assumption that $V_k^T S_1 D_1$ is full rank, we will prove that the $m \times k$ matrix C returned by the algorithm will satisfy:

$$\begin{aligned}\left\|A - P_C A\right\|_\xi \leq\ & \left\|A - A_k\right\|_\xi \\ & + \sigma_k^{-1}\left(V_k^T S_1 D_1 S_2\right) \left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi\end{aligned}$$

for both $\xi = 2, F$. (See Lemma 4.2 below.) Next, we will provide a bound on $\sigma_k^{-1}\left(V_k^T S_1 D_1 S_2\right)$. In order to get a strong accuracy guarantee for the overall algorithm, the deterministic column selection algorithm must satisfy

$$\sigma_k\left(V_k^T S_1 D_1 S_2\right) \geq \frac{\sigma_k\left(V_k^T S_1 D_1\right)}{p(k, \tilde{c})} > 0,$$

where $p(k, \tilde{c})$ is a polynomial in both k and $\tilde{c}$. Thus, for our main theorem, we will employ Algorithm 1 of Pan [28], which guarantees the above bound with $p(k, \tilde{c}) = \sqrt{k\left(\tilde{c} - k\right) + 1}$.[3] (See Lemma 4.3 below.) Finally, we will show, using relatively straightforward matrix perturbation techniques, that $\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi$ is not too much more, in a multiplicative sense, than $\left\|A - A_k\right\|_\xi$, where we note that the factors differ for $\xi = 2, F$. (See Lemmas 4.4 and 4.5 below.) By combining these results, the main theorem will follow.

We should note here that existing proofs for the relative error bound of eqn. (2.2) of Section 2 break down if $o(k \log k)$ columns of A are selected. ($\Omega(k \log k)$ columns seem necessary to guarantee that the matrix of the sampled columns preserves a certain rank constraint that does not seem easy to circumvent.) Thus, extending the theoretical computer science results to pick exactly k columns does not seem easy using existing techniques. On the other hand, it should be noted that if we allow existing numerical linear algebra algorithms to pick more than k columns, it is not clear whether relative error approximation guarantees of the form described in eqn. (2.2) can be obtained. However, a hybrid approach that first selects roughly $k \log k$ columns, and then nails down exactly k columns using a deterministic algorithm, seems to combine the best of both worlds and thus achieve the stated improvement.

[3] To be exact, in the parlance of this paper, Lemma 3.5 of [28] guarantees that $p(k, \tilde{c}) = \sqrt{\mu^2 k\left(\tilde{c} - k\right) + 1}$, for a user-controlled parameter $\mu \geq 1$. [28] suggests using $\mu = 1 + u$, where u is the machine precision. We note that by choosing a larger μ the deterministic step of our algorithm becomes (up to constant factors) faster. We defer a more detailed description of the work of [28] to the full version of this paper [4].

4.1 The rank of $V_k^T S_1 D_1$

The following lemma provides a bound on the singular values of the matrix $V_k^T S_1 D_1$ computed by the *randomized phase* of Algorithm 1, from which it will follow that the matrix $V_k^T S_1 D_1$ is full rank. To prove the lemma, we apply a recent result of Rudelson and Vershynin on approximating the spectral norm of an operator [31, 15]. Note that probabilities of the form (3.5) actually suffice to establish Lemma 4.1. Note also that, by the Coupon Collecting Problem, we cannot set the column sampling parameter c to be less than $\Theta(k \log k)$ (in worst case) at this step.

LEMMA 4.1. *Let S_1 and D_1 be constructed using Algorithm 1. Then, there exists a choice for $c = \Theta(k \log k)$ such that with probability at least 0.9,*

$$\sigma_k\left(V_k^T S_1 D_1\right) \geq 1/2.$$

In particular, $V_k^T S_1 D_1$ has full rank.

Proof: In order to bound $\sigma_k\left(V_k^T S_1 D_1\right)$, we will bound $\left\|V_k^T S_1 D_1 D_1 S_1^T V_k - I_k\right\|_2$. Towards that end, we will use Theorem 7 of [15] with $\beta = 1/2$. This theorem (followed by Markov's inequality) guarantees that given our construction of S_1 and D_1, with probability at least 0.9,

$$\begin{aligned}\left\|V_k^T S_1 D_1 D_1 S_1^T V_k - I_k\right\|_2 &\leq O(1)\sqrt{\frac{\log c}{\beta c}} \left\|V_k\right\|_F \left\|V_k\right\|_2 \\ &\leq O(1)\sqrt{\frac{k \log c}{c}}.\end{aligned}$$

Standard matrix perturbation theory results [20] now imply that for all $i = 1, \ldots, k$,

$$\left|\sigma_i^2\left(V_k^T S_1 D_1\right) - 1\right| \leq O(1)\sqrt{\frac{k \log c}{c}} \leq 1/2$$

for some $c = \Theta(k \log k)$.

4.2 Bounding the spectral and Frobenius norms of $A - P_C A$

LEMMA 4.2. *Let S_1, D_1, and S_2 be constructed as described in Algorithm 1 and recall that $C = A S_1 S_2$. If $V_k^T S_1 D_1$ has full rank, then for $\xi = 2, F$,*

$$\begin{aligned}\left\|A - P_C A\right\|_\xi \leq\ & \left\|A - A_k\right\|_\xi \\ & + \sigma_k^{-1}\left(V_k^T S_1 D_1 S_2\right) \left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi\end{aligned}$$

Proof: We seek to bound the spectral and Frobenius norms of $A - P_C A$, where $C = AS_1S_2$ is constructed by Algorithm 1. To do so, first notice that scaling the columns of a matrix (equivalently, post-multiplying the matrix by a diagonal matrix) by any non-zero scale factors does not change the subspace spanned by the columns of the matrix. Thus,

$$\begin{aligned} A - P_C A &= A - (AS_1S_2)\,(AS_1S_2)^+ A \\ &= A - (AS_1D_1S_2)\,(AS_1D_1S_2)^+ A \\ &= A - (A\mathcal{S})\,(A\mathcal{S})^+ A, \end{aligned} \tag{4.6}$$

where, in the last line, we have introduced the convenient notation $\mathcal{S} = S_1D_1S_2 \in R^{n\times k}$ that we will use throughout the remainder of this proof. By using the SVD of A (i.e., $A = U_A\Sigma_A V_A^T$), it easily follows from (4.6) that

$$\begin{aligned} &\|A - P_C A\|_\xi \\ &= \left\| U_A \left(\Sigma_A - \left(\Sigma_A V_A^T \mathcal{S}\right) \left(U_A\Sigma_A V_A^T \mathcal{S}\right)^+ U_A\Sigma_A \right) V_A^T \right\|_\xi \\ &= \left\| \Sigma_A - \left(\Sigma_A V_A^T \mathcal{S}\right) \left(\Sigma_A V_A^T \mathcal{S}\right)^+ \Sigma_A \right\|_\xi, \end{aligned} \tag{4.7}$$

where (4.7) follows since $(QX)^+ = X^+Q^T$, for any matrix X and any orthogonal matrix Q, and since U_A and V_A^T may be dropped by using the unitary invariance of the spectral and the Frobenius norms. For this equation we had $Q = U_A$ and $X = \Sigma_A V_A^T \mathcal{S}$. In the sequel, for notational simplicity, let

$$\Pi = (\Sigma_A V_A^T \mathcal{S})(\Sigma_A V_A^T \mathcal{S})^+ \in R^{\rho\times\rho}$$

denote the projection matrix onto the column space of $\Sigma_A V_A^T \mathcal{S} = \Sigma_A V_A^T S_1 D_1 S_2$, and let

$$\Sigma_A = \begin{pmatrix} \Sigma_k & \mathbf{0} \\ \mathbf{0} & \Sigma_{\rho-k} \end{pmatrix}.$$

With this notation and some manipulations, (4.7) becomes

$$\begin{aligned} \|A - P_C A\|_\xi &= \left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \Pi \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} \right\|_\xi \\ &+ \left\| \begin{pmatrix} \mathbf{0} \\ \Sigma_{\rho-k} \end{pmatrix} - \Pi \begin{pmatrix} \mathbf{0} \\ \Sigma_{\rho-k} \end{pmatrix} \right\|_\xi. \end{aligned} \tag{4.8}$$

We next bound the two terms in (4.8), starting with the second term. Since $I - (\Sigma_A V_A^T \mathcal{S})(\Sigma_A V_A^T \mathcal{S})^+$ is a projection matrix, it may be dropped without increasing a unitarily invariant norm. Thus,

$$\begin{aligned} \left\| \begin{pmatrix} \mathbf{0} \\ \Sigma_{\rho-k} \end{pmatrix} - \Pi \begin{pmatrix} \mathbf{0} \\ \Sigma_{\rho-k} \end{pmatrix} \right\|_\xi &\le \left\| \begin{pmatrix} \mathbf{0} \\ \Sigma_{\rho-k} \end{pmatrix} \right\|_\xi \\ &= \|A - A_k\|_\xi, \end{aligned} \tag{4.9}$$

thus providing a bound for the second term in (4.8). We next bound the first term in (4.8). To do so, the critical observation is that we can relate the first term to the value of a least squares approximation problem:

$$\begin{aligned} &\left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \Pi \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} \right\|_\xi \\ &= \min_{X\in\mathbb{R}^{k\times k}} \left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \left(\Sigma_A V_A^T \mathcal{S}\right) X \right\|_\xi. \end{aligned} \tag{4.10}$$

This follows for both $\xi = 2, F$ from the fact that $\Pi \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix}$ is the exact projection of the matrix $\begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix}$ on the subspace spanned by the columns of $\Sigma_A V_A^T \mathcal{S}$. We will bound (4.10) by providing a bound for a suboptimal – but for our purposes very convenient – choice for X, namely

$$X = \left(\Sigma_k V_k^T \mathcal{S}\right)^+ \Sigma_k \in \mathbb{R}^{k\times k}.$$

Since X is suboptimal for the least squares approximation problem (4.10), it follows that

$$\begin{aligned} &\left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \Pi \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} \right\|_\xi \\ &\le \left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \left(\Sigma_A V_A^T \mathcal{S}\right) \left(\Sigma_k V_k^T \mathcal{S}\right)^+ \Sigma_k \right\|_\xi. \end{aligned} \tag{4.11}$$

Our suboptimal choice for X leads to an expression on the right hand side in (4.11) that is easier to manipulate and bound. We claim that

$$\left(\Sigma_k V_k^T \mathcal{S}\right)^+ = \left(V_k^T \mathcal{S}\right)^{-1} \Sigma_k^{-1}. \tag{4.12}$$

This follows since the statement of our lemma assumes that the matrix $V_k^T S_1 D_1$ has full rank; also, the construction of S_2 guarantees that the columns of $V_k^T S_1 D_1$ that are selected in the second stage of Algorithm 1 are linearly independent. Thus, the $k \times k$ matrix $V_k^T \mathcal{S} = V_k^T S_1 D_1 S_2$ has full rank and thus is invertible, from which (4.12) follows. Substituting (4.12) into (4.11), it follows that

$$\begin{aligned} &\left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \Pi \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} \right\|_\xi \\ &\le \left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \left(\Sigma_A V_A^T \mathcal{S}\right) \left(V_k^T \mathcal{S}\right)^{-1} \right\|_\xi \\ &= \left\| \begin{pmatrix} \Sigma_k \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \Sigma_k V_k^T \\ \Sigma_{\rho-k} V_{\rho-k}^T \end{pmatrix} \mathcal{S} \left(V_k^T \mathcal{S}\right)^{-1} \right\|_\xi \\ &\le \left\| \Sigma_k - \Sigma_k V_k^T \mathcal{S} \left(V_k^T \mathcal{S}\right)^{-1} \right\|_\xi \\ &\qquad + \left\| \Sigma_{\rho-k} V_{\rho-k}^T \mathcal{S} \left(V_k^T \mathcal{S}\right)^{-1} \right\|_\xi \\ &\le \left\| \Sigma_{\rho-k} V_{\rho-k}^T \mathcal{S} \left(V_k^T \mathcal{S}\right)^{-1} \right\|_\xi. \end{aligned} \tag{4.13}$$

In the above we used the triangle inequality for the spectral and the Frobenius norms, and the fact that $V_k^T \mathcal{S} \left(V_k^T \mathcal{S}\right)^{-1}$ is the $k \times k$ identity matrix.

By combining (4.8), (4.9), and (4.13) we claim that

$$\begin{aligned} \|A - P_C A\|_\xi \leq\ & \|A - A_k\|_\xi \\ & + \sigma_k^{-1}\left(V_k^T \mathcal{S}\right) \left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi . \end{aligned} \tag{4.14}$$

To establish equation (4.14), note that

$$\begin{aligned} &\|A - P_C A\|_\xi \\ &\quad \leq \|A - A_k\|_\xi + \left\|\Sigma_{\rho-k} V_{\rho-k}^T \mathcal{S} \left(V_k^T \mathcal{S}\right)^{-1}\right\|_\xi \\ &\quad \leq \|A - A_k\|_\xi + \left\|\Sigma_{\rho-k} V_{\rho-k}^T \mathcal{S}\right\|_\xi \left\|\left(V_k^T \mathcal{S}\right)^{-1}\right\|_2 \\ &\quad \leq \|A - A_k\|_\xi + \left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi \left\|\left(V_k^T \mathcal{S}\right)^{-1}\right\|_2 \\ &\quad \leq \|A - A_k\|_\xi + \sigma_k^{-1}\left(V_k^T \mathcal{S}\right) \left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi . \end{aligned}$$

The inequality in the third line follows from the fact that for any two matrices X and Y and $\xi = 2, F$, $\|XY\|_\xi \leq \|X\|_\xi \|Y\|_2$. The inequality in the fourth line follows since S_2 is an orthogonal matrix and thus dropping it does not increase any unitarily invariant norm. Finally, if $\sigma_k\left(V_k^T \mathcal{S}\right)$ is the k-th singular value of the $k \times k$ matrix $V_k^T \mathcal{S}$, the inequality in the last line follows from standard linear algebra.

4.3 Upper bounds for $\sigma_k^{-1}\left(V_k^T S_1 D_1 S_2\right)$ and $\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_\xi$, $\xi = 2, F$

LEMMA 4.3. *Let S_1, D_1, and S_2 be constructed using Algorithm 1. If $c = \Theta(k \log k)$, then with probability at least* 0.9*,*

$$\sigma_k^{-1}\left(V_k^T S_1 D_1 S_2\right) \leq 2\sqrt{k\left(\tilde{c} - k\right) + 1}.$$

Proof: From Lemma 4.1 we know that $\sigma_i\left(V_k^T S_1 D_1\right) \geq 1/2$ holds for all $i = 1, \ldots, k$ with probability at least 0.9. The deterministic construction of S_2 (see Algorithm 1 and Lemma 3.5 in [28]) guarantees that

$$\sigma_k(V_k^T S_1 D_1 S_2) \geq \frac{\sigma_k(V_k^T S_1 D_1)}{\sqrt{k\left(\tilde{c} - k\right) + 1}} \geq \frac{1}{2\sqrt{k\left(\tilde{c} - k\right) + 1}}.$$

LEMMA 4.4. *($\xi = 2$) If S_1 and D_1 are constructed as described in Algorithm 1, then with probability at least* 0.9*,*

$$\begin{aligned} &\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_2 \\ &\quad \leq \left(1 + O(1)\sqrt{\frac{(\rho - k + 1)\log c}{c}}\right)^{1/2} \|A - A_k\|_2 . \end{aligned}$$

Proof: We manipulate $\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_2^2$ as follows:

$$\begin{aligned} &\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_2^2 \\ &\quad = \left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1 D_1 S_1^T V_{\rho-k} \Sigma_{\rho-k}\right\|_2 \\ &\quad = ||\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1 D_1 S_1^T V_{\rho-k} \Sigma_{\rho-k} \\ &\qquad - \Sigma_{\rho-k} V_{\rho-k}^T V_{\rho-k} \Sigma_{\rho-k} + \Sigma_{\rho-k} V_{\rho-k}^T V_{\rho-k} \Sigma_{\rho-k}||_2 \\ &\quad \leq ||\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1 D_1 S_1^T V_{\rho-k} \Sigma_{\rho-k} \\ &\qquad - \Sigma_{\rho-k} V_{\rho-k}^T V_{\rho-k} \Sigma_{\rho-k}||_2 + \left\|\Sigma_{\rho-k}^2\right\|_2 . \end{aligned}$$

Given our construction of S_1 and D_1, and applying Markov's inequality and Theorem 7 of [15] with $\beta = 1/2$, we get that with probability at least 0.9,

$$\begin{aligned} &\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1 D_1 S_1^T V_{\rho-k} \Sigma_{\rho-k} - \Sigma_{\rho-k} V_{\rho-k}^T V_{\rho-k} \Sigma_{\rho-k}\right\|_2 \\ &\qquad \leq O(1)\sqrt{\frac{\log c}{\beta c}} \|\Sigma_{\rho-k}\|_F \|\Sigma_{\rho-k}\|_2 . \end{aligned}$$

Thus, by combining these expressions, we have that

$$\begin{aligned} &\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_2^2 \\ &\qquad \leq O(1)\sqrt{\frac{\log c}{c}} \|\Sigma_{\rho-k}\|_F \|\Sigma_{\rho-k}\|_2 + \|\Sigma_{\rho-k}\|_2^2 . \end{aligned}$$

Using $\|\Sigma_{\rho-k}\|_2 = \|A - A_k\|_2$ and $\|\Sigma_{\rho-k}\|_F \leq \sqrt{\rho - k + 1}\,\|A - A_k\|_2$ concludes the proof of the lemma.

LEMMA 4.5. *($\xi = F$) If S_1 and D_1 are constructed as described in Algorithm 1, then with probability at least* 0.9*, $\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_F \leq 4 \|A - A_k\|_F$.*

Proof: It is straightforward to prove that with our construction of S_1 and D_1, the expectation of $\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_F^2$ is equal to $\left\|\Sigma_{\rho-k} V_{\rho-k}^T\right\|_F^2$. In addition, note that the latter quantity is exactly equal to $\|A - A_k\|_F^2$. Applying Markov's inequality, we get that with probability at least 0.9,

$$\left\|\Sigma_{\rho-k} V_{\rho-k}^T S_1 D_1\right\|_F^2 \leq 10 \|A - A_k\|_F^2 ,$$

which implies the lemma.

4.4 Completing the proof of Theorem 1

To prove the Frobenius norm bound in Theorem 1 we use Lemmas 4.1, 4.2, 4.3, and 4.5. Notice that Lemmas 4.1 and 4.3 fail with probability at most 0.1, and that Lemma 4.5 fails with probability at most 0.1. Overall, by combining all these and applying the standard union bound, it follows that the Frobenius norm bound in Theorem 1 holds with probability at least 0.7. The proof of the spectral norm bound in Theorem 1 is similar, except for employing Lemma 4.4 instead of Lemma 4.5.

Acknowledgements

We are grateful to Daniel Spielman for useful discussions as well as his encouragement and enthusiasm for the results of this paper.

References

[1] A. Ben-Hur and I. Guyon. Detecting stable clusters using principal component analysis. *Methods Mol Biol*, 224:159–182, 2003.

[2] C.H. Bischof and G. Quintana-Ortí. Computing rank-revealing QR factorizations of dense matrices. *ACM Trans on Math Soft*, 24(2):226–253, 1998.

[3] C. Boutsidis, M.W. Mahoney, and P. Drineas. Unsupervised Feauture Selection for Principal Components Analysis. In *KDD,* 2008.

[4] C. Boutsidis, M.W. Mahoney, and P. Drineas. An Improved Approximation Algorithm for the Column Subset Selection Problem. *Manuscript*, 2008.

[5] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra Appl*, 88/89:67–82, 1987.

[6] T.F. Chan and P.C. Hansen. Some applications of the rank revealing QR factorization. *SIAM J Sci and Stat Comp*, 13:727–741, 1992.

[7] T. F. Chan and P.C. Hansen. Low-rank revealing QR factorizations. *Linear Algebra Appl*, 1:33–44, 1994.

[8] S. Chandrasekaran and I. C. F. Ipsen. On rank-revealing factorizations. *SIAM J Matrix Anal Appl*, 15:592–622, 1994.

[9] S. Chatterjee and A.S. Hadi. Sensitivity Analysis in Linear Regression. John Wiley and Sons, 1988.

[10] A. Civril and M. Magdon-Ismail. Finding Maximum Volume Sub-matrices of a Matrix. *RPI Comp Sci Dept TR 07-08*, 2007.

[11] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *SODA*, 2006.

[12] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In *APPROX-RANDOM*, 2006.

[13] P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. In *STOC*, 2002.

[14] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *APPROX-RANDOM*, 2006.

[15] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J Matrix Anal Appl*, 30:844–881, 2008.

[16] L. V. Foster. Rank and null space calculations using matrix decomposition without column interchanges. *Linear Algebra Appl*, 74:47–71, 1986.

[17] L.V. Foster and Xinrong Liu. Comparison of rank revealing algorithms applied to matrices with well defined numerical ranks. *Manuscript*, 2006.

[18] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *FOCS*, 1998.

[19] G. H. Golub. Numerical methods for solving linear least squares problems. *Numer Math*, 7:206–216, 1965.

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

[21] M. Gu and S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J Sci Comp*, 17:848–869, 1996.

[22] Y. P. Hong and C. T. Pan. Rank-revealing QR factorizations and the singular value decomposition. *Math Comp*, 58:213–232, 1992.

[23] W. J. Krzanowski. Selection of variables to preserve multivariate data structure, using principal components. *Applied Statistics*, 36(1):22–33, 1987.

[24] M.W. Mahoney, M. Maggioni, and P. Drineas. Tensor-CUR decompositions for tensor-based data. *SIAM J Matrix Anal Appl*, 30:957–987, 2008.

[25] M.W. Mahoney, M. Maggioni, and P. Drineas. Tensor-CUR decompositions for tensor-based data. In *KDD*, 2006.

[26] K. Z. Mao. Identifying critical variables of principal components for unsupervised feature selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(2):339–344, 2005.

[27] P.G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the approximation of matrices. *Yale Comp Sci Dept TR 1361*, 2006.

[28] C. T. Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra Appl*, 316:199–222, 2000.

[29] C. T. Pan and P. T. P. Tang. Bounds on singular values revealed by QR factorizations. *BIT Numerical Mathematics*, 39:740–756, 1999.

[30] P. Paschou, E. Ziv, E.G. Burchard, S. Choudhry, W.R. Cintron, M.W. Mahoney, and P. Drineas. PCA-Correlated SNPs for Structure Identification in Worldwide Human Populations. *PLoS Genetics*, 9(3), 2007.

[31] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J ACM*, 54:4, Article No. 21, 2007.

[32] G.W. Stewart. Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix. *Num Math*, 83:313–323, 1999.

[33] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar. Ranking a random feature for variable and feature selection. *J. Mach. Learn. Res.*, 3:1399–1414, 2003.

[34] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos. Less is more: Compact matrix decomposition for large sparse graphs. In *SDM*, 2007.

[35] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Yale Comp Sci Dept TR 1380*, 2007.

[36] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *J. Mach. Learn. Res.*, 6:1855–1887, 2005.

[37] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, 2007.