9-2020

Matrix Low Rank Approximation at Sublinear Cost

Qi Luan
The Graduate Center, City University of New York

Matrix Low Rank Approximation at Sublinear Cost

by

Qi Luan

A dissertation submitted to the Graduate Faculty in Mathematics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2020

This manuscript has been read and accepted for the Graduate Faculty in Mathematics in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

(required signature)

Date — Chair of Examining Committee

(required signature)

Date — Executive Officer

Victor Pan(Chair)

Christina Zamfirescu

Christina Sormani

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Matrix Low Rank Approximation at Sublinear Cost

by

Qi Luan

Advisor: Victor Y. Pan

A matrix algorithm runs at sublinear cost if the number of arithmetic operations involved is far fewer than the number of entries of the input matrix. Such algorithms are especially crucial for applications in the field of Big Data, where input matrices are so immense that one can only store a fraction of the entire matrix in memory of modern machines. Typically, such matrices admit Low Rank Approximation (LRA) that can be stored and processed at sublinear cost. Can we compute LRA at sublinear cost? Our counter example presented in Appendix C shows that no sublinear cost algorithm can compute accurate LRA for arbitrary input. However, for a decade, researchers observed that many sublinear cost algorithms, such as Cross Approximations (C–A) iterations, routinely compute accurate LRA.

We partly resolve this long-known contradiction by proving that:

(i) sublinear cost variations of a popular subspace sampling algorithm can compute accurate LRA for a large class of inputs with high probability;

(ii) a single two-stage C–A loop computes accurate LRA given that the input is reasonably close to a low rank matrix and the C–A loop starts with a submatrix that shares the same numerical rank with the input;

(iii) for arbitrary Symmetric Positive Semi-Definite (SPSD) input, there exists a deterministic sublinear cost algorithm that outputs close to optimal LRA in the Chebyshev norm;

(iv) for any input, an LRA based on given sets of columns and rows can be computed at sublinear cost, and this approximation is near optimal.

Acknowledgements

My journey as a Ph.D. student has been invaluable to me, and I am indebted to those who have advised and supported me along the way. First and foremost, I would like to express my deepest gratitude to my advisor Professor Victor Y. Pan. His passion for mathematical research has greatly influenced me since the first day he introduced me to the exciting field of algebraic computation. I have always got inspiration from the constructive discussion with him when I encountered research challenge. Without his guidance, I would not be able to complete this thesis. I thank Professor Christina Zamfirescu and Professor Christina Sormani for serving on my dissertation committee, and I am grateful for their helpful suggestions and their efforts to accommodate my remote defense during the difficult time of pandemic. I thank Professor Liang Zhao for his collaboration, and for inviting me to the study group he organized, in which I had the opportunity to get in touch with many other graduate students and learn about their research. Finally,

I thank my parents and my wife, Dian Yu, for supporting my pursuit of the doctoral degree. It is their love and encouragement motivating me to overcome the difficulties both in study and in life.

This work and my research are supported by NSF Grants CCF–1563942 and CCF–1733834, and PSC CUNY Award 69813 00 48.

List of Figures

Contents

Chapter 1

Introduction

1.1 Matrix Low Rank Approximation at Sublinear Cost

Low rank approximation (LRA) of a matrix has wide applications to fundamental numerical computation, machine learning, and data mining, and it remains a hot research area of Numerical Linear Algebra (NLA) and Computer Science (CS) [33, 46, 15, 40].

Fix a matrix norm $||\cdot||$, and a positive tolerance ϵ, an $m \times n$ matrix W has close approximation of rank at most r if and only if W has *numerical rank* less or equal to r (write as $\text{nrank}(W) \leq r$), or equivalently

$$W = AB + E,\ ||E||/||W|| \leq \epsilon, \tag{1.1}$$

Portions of this chapter previously appeared in our work [43], [44], and [57].

for $A \in \mathbb{C}^{m\times r}$, $B \in \mathbb{C}^{r\times n}$.

In such an LRA $A \cdot B$ approximates W using only $(m+n)r$ entries rather than mn entries. This compression in size is especially beneficial with applications in the area of Big Data, where the data matrices are usually so immense that only a tiny fraction of them can be stored in memory or compute with, but at the same time it is quite typical that such matrices have LRA of (1.1) where $(m+n)r \ll mn$. (Here and hereafter we let inequalities $a \ll b$ and $b \gg a$ indicate that the ratio $|a/b|$ is small in context.)

One can operate with low rank matrices *at sublinear computational cost*, that is, by using much fewer arithmetic operations and memory cells than an input matrix has entries, but can we compute LRA at sublinear cost? Yes and no. The answer can be no since every sublinear cost LRA algorithm fails even on the small input family of Appendix C. The answer can be yes, because (i) sublinear cost variations of a popular *subspace sampling* algorithm in Chapter 3 output accurate LRA for a large class of input; (ii) a single two-stage C–A loop computes accurate LRA given that the input is reasonably close to a low rank matrix and the C–A loop starts with a submatrix that shares the same numerical rank with the input; (iii) sublinear cost maximal volume based CUR algorithms in Chapter 4 have quasi-optimal error in the Chebyshev Norm if the input matrix satisfies certain conditions, e.g., being

Symmetrical Positive Semi-Definite (SPSD).

We will provide more details in Sections 1.2, 1.3, and 1.4.

1.2 Matrix LRA via Subspace Sampling

Subspace sampling algorithms compute LRA of a matrix M with the help of auxiliary matrices FM, MH or FMH, where F and H are generated randomly, have smaller sizes, and are called *test matrices*. These algorithms output nearly optimal LRA with high probability (whp) given that F and H are randomly generated under certain probability distribution;[1] these algorithms also output accurate LRA in practice with other randomly generated multipliers consistently, however all of the aforementioned multipliers multiply with M at superlinear cost.

We modify these algorithms in Chapter 3 such that the dense randomized multipliers are replaced with sparse orthogonal (e.g., *subpermutation*) multipliers[2] F and H, and as we proved, the modified algorithms run at sublinear computation cost, and output LRA that is reasonably close to the input matrix given that the input matrix is under two distinct random low rank

[1] These randomly generated multipliers include (1) *"Gaussian"* multiplier where all entries are iid standard normal variables; (2) *SRHT* and *SRFT* multipliers which refer to "Subsampled Randomized Hadamard and Fourier Transform"; (3) Rademacher's multiplier where all entries are iid random variables being ± 1 with equal probability.

[2] We define subpermutation matrices as submatrices of permutation matrices that have full rank.

models; and we prove the probabilistic error bound of the LRA computation with these two models in Section 3.4.

We acknowledge that any sublinear cost LRA algorithm fails on some constructed hard inputs, however our proposed classes of random low rank matrices are quite natural for many real world applications, and our approach leads to new insights of this subject. Our extensive numerical tests with both synthetic and real world inputs are in good accordance with our formal study.

1.3 Matrix LRA via Maximal Volume Generator

For more than a decade Cross–Approximation (C–A) iterations, running at sublinear cost, have been routinely computing close LRA worldwide. Moreover they output LRA in its special form of CUR LRA (see Section 4.1.1), particularly memory efficient and defined by a proper choice of a submatrix G of W, which we call a generator of CUR LRA or a CUR generator.

Let $\sigma_j(M)$ denote the jth largest singular value of a matrix M and recall that this is the minimal distance from M to a matrix of rank $j-1$ in spectral norm. The first result of Chapter 4 provides partial formal support for this empirical phenomenon. Namely suppose that C-A iterations are applied to an $m \times n$ matrix W that admits a sufficiently close LRA (1.1), that is,

$\sigma_{r+1}(W)$ is small. Let W_i and V_i denote the input and output submatrices of W at the ith C-A iteration for $i = 1, 2, \dots$ and let $||\cdot||$ denote the spectral or Frobenius matrix norm. Then we prove (see Corollary 4.4.1 and Remark 4.4) that the error norm $||W - V_{i+1}||$ is within a specified reasonable factor f from optimal unless W_i lies close to a matrix of rank $r-1$, that is, unless $\sigma_r(W_i)$ is small. Our estimates of Appendix B.1 imply rather mild upper and lower bounds on the values $\sigma_{r+1}(W)$ and $\sigma_r(W_i)$, respectively.

Our proof relies on deep known results (cf. [50], the references therein, and Theorems 4.1 and 4.2) on bounding the output errors of CUR LRA in term of maximization of the volume $v_2(G)$ or r-projective volume $v_{2,r}(G)$ of its $k \times l$ CUR generator G,

$$v_2(G) = \prod_{j=1}^{\min\{k,l\}} \sigma_j(G) \text{ and } v_{2,r}(G) = \prod_{j=1}^{\min\{k,l\}} \sigma_j(G). \tag{1.2}$$

Clearly the required bounds on $\sigma_{r+1}(W)$ and $\sigma_r(W_i)$ cannot hold for the families of the matrices W of Appendix C, for these families do not admit LRA at sublinear cost. Indeed for such matrices $r = 1$ while typically $\text{rank}(W_i) = 0$. Such consistent degeneracy of all submatrices W_i is exceptional in the class of all matrices that admit rank-1 approximation, however: if a random input matrices W lies near rank-r matrix than so does its any fixed $r \times r$ submatrix whp. Moreover whp the submatrices W_i do not degen-

erate for inputs FW, WH, and FWH for random Gaussian matrices F and H (cf. [56]) and empirically also for various sparse orthogonal multipliers F and H, for which we can move from W to FW, WH, and FWH and back at sublinear cost. In practice, even with no preprocessing, one typically obtains close CUR LRA in a small number of C-A iterations, although rarely in two iterations. The error bound of the above computation deviates from the optimal error bound by some factor f, which can be considered a price for obtaining CUR LRA at sublinear cost, but if the optimal error bound is reasonably small, we can optimistically apply any of our two heuristic algorithms of [53] for iterative refinement of LRA running at sublinear cost.

The second result of Chapter 4 computes reasonably close CUR LRA of an SPSD Matrix, by applying our novel algorithm, rather than C-A iterations, and do not restrict the input class by imposing any further assumptions. Then again our algorithm design and analysis rely on the cited link of the error bounds of an output CUR LRA and maximization of the volume or r-projective volume of a CUR generator. Then again our errors deviate from optimal by a reasonable factor f, and one can try to decrease this factor by applying iterative refinement algorithms of [53].

1.4 Improving CUR Matrix LRA via Double-Sided LSR

Matrix CUR decomposition aims at finding low-rank matrix approximation with original matrix rows and columns. Volume-based CUR LRA algorithms [2, 31, 30, 28, 27, 70, 50, 56] select columns C and rows R such that their intersection matrix W has near maximal volume(or projective volume), and then compute the U matrix as W^{-1}(or W_r^+). However, U constructed this way only takes the local information in to consideration, and is often not optimal in terms of the Frobenius norm error $||A-CUR||_F$. Sampling-based randomized CUR LRA algorithms [19, 68, 7] first construct C as a subset of columns sampled with a probability distribution reflecting the "importance" of each column, then sample rows to obtain R, and lastly construct an appropriate middle factor U connecting C and R. Especially for the algorithm proposed in [7], it is proved to achieve relative-error and number of columns and rows selected are asymptotically optimal. However, the constant coefficient on the sample numbers are likely to be huge, and therefore in practice the C and R constructed with these algorithms are usually heavily undersampled, making the corresponding U less reliable.

In Chapter 5 we develop a method that can be used to further improve

existing sampling-based CUR algorithms by providing a near-optimal choice of the middle block U. Given factor C and R, we treat the task of finding U as *double-sided least squares* problem $\min_Z \|A - CZR\|_F$. The optimal solution is $Z_{opt} = C^+AR^+$, where M^+ represents the Moore-Penrose matrix pseudo inverse. Ideally one would use $U = Z_{opt}$ to form a CUR approximation, but its cost is unbearable when the size of A is much greater than the size of Z. We instead develop a randomized algorithm that solves a down-sampled problem

$$\min_{Z \in \mathbb{R}^{d_1 \times d_2}} \sum_{(i,j) \in S} \frac{(A_{ij} - C_i Z R^j)^2}{p_{ij}|S|}. \tag{1.3}$$

Here the index set S is a small subset of matrix indices sampled with replacement according to probability distribution $\{p_{ij}\}_{1 \le i \le m, 1 \le j \le n}$. Problem (1.3) is much easier to solve than the double-sided least squares regression(LSR) problem. We show that if the sampling probabilities are carefully chosen, its solution can well approximate Z_{opt}, and thus providing a better overall CUR approximation to the matrix A. Besides facilitating deterministic and randomized CUR algorithms, the algorithm developed in Chapter 5 applies to the double-sided least squares problem in general, which has its own applications such as computing the Karhunen-Loeve expansion in image processing [22].

To demonstrate the effectiveness of the proposed algorithm, we run nu-

merical tests on several large-scale matrices representing real-world datasets, some of which contain over one billion values. Combined with a sampling strategy that uses leverage-scores on general matrices [19] or uniform-sampling on partially observed low-coherence matrices [69], our algorithm can produce CUR approximations with approximation error constantly closer to the optimal error comparing to existing CUR algorithms.

Chapter 2

Related Work

We refer the readers to articles [7], [68], [46], [33], [40], [55], [63], [50], [62], [52], [19] and the references therein for part of the huge bibliography on Matrix LRA and CUR LRA.

The study of CUR (aka CGR and pseudo-skeleton) LRA can be traced back to the skeleton decomposition in [23] and QRP factorization in [25] and [8], redefined and refined as rank-revealing factorization in [10]. The CUR LRA algorithms in [11], [12], [36], [37], [13], [32], and [51] largely rely on the maximization of the volume $(\det(G^*G))^{1/2}$ of a CUR generator G (which is a submatrix of an input matrix). This fundamental idea goes back to [41] and has been developed in [66], [64], [30], [31], [28], [27], [29], and most recently in [50]. The study in these papers reveals the crucial property that

Portions of this chapter previously appeared in our work [43], [44], and [57].

the computation of LRA requires no factorization of the input matrix but just proper selection of its row and column sets and was the springboard for our progress.

C–A iterations were a natural extension of the latter observation preceded by the Alternating Least Squares method of [9] and [35] and leading to dramatic empirical decrease of quadratic memory space and cubic arithmetic time used by LRA algorithms. The concept of C–A was implicit in [64] and coined in [65]; we credit [2], [4], [27], [48], [3], and [39] for devising efficient C–A and adaptive C-A algorithms. [49] proposed efficient randomized low cost algorithms for the approximation of maximal volume 1×1 submatrix.

Part of the results in Chapter 4 have appeared in arxiv reports [54, Section 5] and [55, Part II] together with various results on LRA of random input matrices.[1] Our progress on Sublinear SPSD CUR LRA in Chapter 4 has been ignited by the observation in [17] that in the case of an SPSD input it is sufficient to maximize the volume of just principal submatrices of W. The paper [47], which followed [54] and [55], proposed sublinear cost randomized

[1]The papers [54] and [55] provide first formal support for LRA at sublinear cost, which they call "superfast" LRA. That work, unsuccessfully submitted to ACM STOC 2017 and published only in the above preprints in arxiv, has extended to LRA the earlier techniques of [58], [59], and [60], proposed for the analysis of randomized Gaussian elimination with no pivoting and other fundamental matrix computations. In turn it was followed by some progress by other authors in devising sublinear cost LRA algorithms for some important special input classes, in particular by Musco and Woodruff in [47]

algorithms for LRA of an SPSD matrix. The algorithms are much more involved than our and exploit a distinct approach and distinct techniques of random subspace sampling. We also cite a link of our Algorithm 4.3 to [16].

[69] studys CUR matrix approximation of low-coherence matrices that can only be observed partially, and they propose an additive-error algorithm using uniform sampling for rows/columns, as well as uniform sampling for solving the resulting double-sided least squares problem. The algorithm proposed in Chapter 5 differentiates from [69] in that our proposed algorithm uses sampling probabilities derived from the input matrices, and thus it is applicable to arbitrary inputs. Moreover, we show that our algorithm can achieve small relative error, which is more desirable in practice.

Chapter 3

Low Rank Approximation by Means of Subspace Sampling

3.1 Basic Definitions and Notations

We use basic definitions for matrix computations recalled in Appendix A.1. We let "$\ll$" and "$\gg$" denote "much less than" and "much greater than", respectively. We let *"Flop"* denote "floating point arithmetic operation", and *"iid"* denote "independent identically distributed". We let $||\cdot||$ and $||\cdot||_F$ denote the spectral and the Frobenius matrix norms, respectively; $|\cdot|$ can denote either of them and is specified in context. M^+ denotes the Moore – Penrose pseudo inverse of matrix M, and $\sigma_i(M)$ denotes the i-th singular value of M.

Portions of this chapter previously appeared in our work [53] and [57].

We let $\mathbb{R}^{p\times q}$ denote the set of $p \times q$ matrices with real entries. We present our results on real matrices, however most of our results extend to complex matrices readily. We refer the interested readers to [20], [14], [21], and [63] for relevant results on complex Gaussian matrices.

3.2 Known Algorithms of LRA by Means of Subspace Sampling

Algorithm 3.1: Column Subspace Sampling (see Remark 3.1).

Input: Matrix $W \in \mathbb{R}^{m\times n}$ and positive integer r as target rank.
Output: Two matrices $A \in \mathbb{R}^{m\times l}$ and $B \in \mathbb{R}^{l\times m}$, and $\tilde{W} = AB$ is an LRA of W.

Initialization: Fix an integer l, $r \le l \le n$, and an $n \times l$ matrix H of full rank l.

Compute the $m \times l$ matrix WH.

Fix a nonsingular $l \times l$ matrix T^{-1}.

Compute the $m \times l$ matrix $A := WHT^{-1}$.

Compute the $l \times n$ matrix $B := \mathrm{argmin}_V \ |AV - W| = A^+W$.

Output A and B.

Remark 3.1. *Let WH have full rank l. Then $AB = WH(WH)^+W$ is independent from the choice of T^{-1}, however, a proper choice of matrix T can stabilizes the computation numerically. Suppose* $\mathrm{nrank}(WH) \le r < l$, *then matrix WH is ill-conditioned and is numerically unstable under matrix*

(pseudo) inversion. Let $WH = QR\Pi$ be the (rank-reveling) QR factorization of WH, and $T = R\Pi$, then $A = WHT^{-1} = Q$ is an orthogonal matrix (with minimum condition number 1).

Algorithm 3.2: Row Subspace Sampling (See Remark 3.2.)

Input: Matrix $W \in \mathbb{R}^{m\times n}$ and positive integer r as target rank.
Output: Two matrices $A \in \mathbb{R}^{k\times n}$ and $B \in \mathbb{R}^{m\times k}$, and $\tilde{W} = AB$ is an LRA of W.

Initialization: Fix an integer k, $r \le k \le m$, and a $k \times m$ matrix F of full numerical rank k.

Compute the $k \times m$ matrix FW.

Fix a nonsingular $k \times k$ matrix S^{-1}.

Compute $k \times n$ matrix $A := S^{-1}FW$.

Compute $m \times k$ matrix $B := \mathrm{argmin}_V \, |VA - W| = WA^+$.

Output A and B.

Remark 3.2. *Similar to Remark 3.1, proper choice of S stabilizes the computation.*

We combine row and column subspace Sampling in the following Algorithm 3.3. The algorithm of [63, Section 1.4] is a special case where matrix S is set as the identity.

Algorithm 3.3: Row and Column Subspace Sampling. (See Remark 3.3.)

Input: Matrix $W \in \mathbb{R}^{m \times n}$ and positive integer r as target rank.
Output: Two matrices $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times m}$, and $\tilde{W} = AB$ is an LRA of W.

Initialization: Fix two integers k and l, $r \le k \le m$ and $r \le l \le n$; fix two matrices $F \in \mathbb{R}^{k \times m}$ and $H \in \mathbb{R}^{n \times l}$ of full numerical ranks and two nonsingular matrices $S \in \mathbb{R}^{k \times k}$ and $T \in \mathbb{R}^{l \times l}$.

Compute the matrix $A = WHT^{-1} \in \mathbb{R}^{m \times l}$.

Compute the matrices $U := S^{-1}FW \in \mathbb{R}^{k \times n}$ and $V := S^{-1}FA \in \mathbb{R}^{k \times l}$.

Compute the $l \times n$ matrix $B := \mathrm{argmin}_Z |VZ - U|$.

Output matrices A and B.

Algorithm 3.4: Column and Row Subspace Sampling

Input: Matrix $W \in \mathbb{R}^{m \times n}$ and positive integer r as target rank.
Output: Two matrices $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times m}$, and $\tilde{W} = AB$ is an LRA of W.

Apply Algorithm 3.3 to W^T and r, and let A, B be the output.

Output A^T and B^T.

Remark 3.3. *Similar to Remark 3.1, S and T can be chosen appropriately to stabilize the computation.*

The bottle-neck of Algorithm 3.3 is the matrix by matrix product involving the input matrix W. More specifically, the computation of WH, FW, and FWH cannot be performed at sublinear cost if F and H are arbitrary matrices. However by letting multipliers F and H be subpermutation matrices, these matrix by matrix multiplications, and hence the overall cost of Algorithm 3.3 is sublinear. The above claims apply to Algorithm 3.4 as well.

In the next section, we bound the output error of Algorithm 3.1 for any input provided that $W_r HT^{-1}$ has rank at least r, then we extend these error bounds to random inputs.

3.3 Deterministic Output Error Bounds for Subspace Sampling Algorithms

Assuming WHT^{-1} and $S^{-1}FW$ are given, and $kl \ll m$, then the rest of Algorithm 3.3 can be completed at sublinear arithmetic cost in $O(kln)$.

Further assume that $k^2 \ll m$ and $l^2 \ll n$. Choosing appropriate sparse and full rank multipliers H and F, the cost of computing WHT^{-1} and $S^{-1}FW$ can also be reduced to sublinear. While, as we have pointed out, we cannot guarantee the error bound of the computed LRA, we can esti-

mate the error given that $W_r HT^{-1}$ has rank at least r, and F is constructed accordingly at sublinear cost. In the next section, we extend this result to random inputs under our random models, and prove that whp the error of the output LRA is within a specified error bound.

In this section, we deduce the deterministic output error bounds for any input matrix, and according to our study the error incurred during the sampling stage is the dominating part of the overall output error bound.

3.3.1 Deterministic Error Bounds of Column Subspace Sampling

Theorem 3.1. [33, Theorem 9.1] *Let W be a matrix such that*

$$W = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{3.1}$$

is a SVD, and let $W_r = U_1 \Sigma_1 V_1^Y$, and $W - W_r = U_2 \Sigma_2 V_2^T$. Fix multiplier H and let

$$C_1 = V_1^T H,\ C_2 = V_2^T H,\ \ \mathrm{rank}(C_1) = r. \tag{3.2}$$

Let A and B be the output of Algorithm 3.1 (Column Subspace Sampling), then

$$|W - AB|^2 \le |\Sigma_2|^2 + |\Sigma_2 C_2 C_1^+|^2. \tag{3.3}$$

Furthermore, $\Sigma_2 = O$ and $AB = W$ if $\mathrm{rank}(W) = r$. The columns of V_1 span the top right singular space of W.

We simplify inequality (3.3) and obtain that

$$|W - AB| \leq (1 + |C_1^+|^2)^{1/2} \bar{\sigma}_{r+1}(W) \tag{3.4}$$

where $C_1 = V_1^T H$ and $\bar{\sigma}_{r+1}(W)$ is $\sigma_{r+1}(W)$ when Spectral norm is in use, and $\sqrt{\sum_{i=r+1}^{\min(m,n)} \sigma_i^2(W)}$[1] when Frobenious norm is in use, and thus the error norm is $(1 + |C_1^+|^2)^{1/2}$ times the optimal error bound (in both norms). In the following, we will compute an upper bound for $|C_1^+|$ and hence an upper bound for the output error of the Algorithm 3.1.

Corollary 3.1.1. *Under the assumptions of Theorem 3.1 let the matrix $W_r H$ have full rank r. Then*

$$|C_1^+| \leq |(W_r H)^+| \, |W_r| \leq |(W_r H)^+| \, |W|.$$

Proof. Recall that $W_r H = U_1 \Sigma_1 C_1$, and hence $C_1 = \Sigma_1^{-1} U_1^T W_r H$. Apply Lemma A.1, and obtain that

$$|C_1| \leq |\Sigma_1^{-1} U_1^T| \, |W_r H|.$$

Since U_1 is orthogonal, so it does not change the norm of Σ_1^{-1}, and the Corollary is proved. □

[1] Assume that $W \in \mathbb{R}^{m \times n}$.

Corollary 3.1.2. *Under the assumptions of Corollary 3.1.1, and the assumption that*

$$\eta := 2\sigma_{r+1}(W)\ ||((WH)_r)^+|| < 1.$$

the following inequality holds,

$$||C_1^+\ || \leq \frac{||W||}{1-\eta}\ ||((WH)_r)^+||.$$

Proof. Recall Lemma A.3 and obtain that

$$\max\{||W_rH - WH||, ||WH - (WH)_r||\} \leq \sigma_{r+1}(W).$$

Therefore, $||W_rH - (WH)_r|| \leq 2\sigma_{r+1}(W)$, and we have

$$\sigma_r(W_rH) \geq \sigma_r(WH) - 2\sigma_{r+1}(W).$$

Hence we obtain

$$\begin{aligned} ||(W_rH)^+|| = 1/\sigma_r(W_rH) &\leq \frac{1}{\sigma_r(WH) - 2\sigma_{r+1}(W)} \\ &= \frac{\sigma_r^{-1}(WH)}{1 - 2\sigma_{r+1}(W)\sigma_r^{-1}(WH)} \end{aligned}$$

Recall that $\sigma_r^{-1}(WH) = ||(WH)_r^+||$, substitute and obtain the Corollary. □

Once given matrix $WH \in \mathbb{R}^{m\times l}$, and that $l^2 \ll n$, $||(WH)_r^+||$ can be computed at sublinear cost. Therefore if the input matrix is ξ close to a rank r matrix, which is a reasonable assumption since otherwise the input matrix

has higher numerical rank, we can determine if the error bound of Corollary 3.1.2 holds at sublinear cost by verifying $||(WH)_r^+|| > 2\xi$. If the error bound holds, and we have estimates for $||W||$, we are able to deduce *a posteriori estimates* of the output error of Algorithm 3.1.

3.3.2 Error Bound of Column and Row Subspace Sampling

In the previous section, we establish the output error bound of Algorithm 3.1 under mild assumption. However, in order to achieve sublinear cost, we avoid computing the full regression of

$$B = \arg\min_V ||AV - W||$$

by carefully constructing multiplier F, and apply Algorithm 3.3. We present two ways of constructing such F, one randomized and one deterministic, and our study show that the error bounds for both ways of construction are dominated by the bound of Corollary 3.1.2, and both constructions achieve sublinear cost under mild assumption specified in this section.

We first present the **randomized construction**, and this construction is due to Algorithm *Exactly(c)* proposed in [19]. Given matrix $A = WH$ and its compact SVD $A = U_A \Sigma_A V_A^T$, compute probabilities

$$p_i = \frac{\sum_{j=1}^{l} U_A(i,j)}{l} \quad \text{for } i = 1, ..., m, \tag{3.5}$$

where $U_A(i,j)$ is the (i,j)-th entry of U_A. If we sample enough rows according to probability distribution $\{p_i\}_{i=1}^m$, we can form a down-sampled problem whose solution ensures close to optimal error bound in Frobenious norm. We formally state the result with the following Lemma.

Lemma 3.1. *Under the assumption of Corollary 3.1.2, let $\{p_i\}_{i=1}^m$ be defined as in (3.5). Fix tolerance $\epsilon < 1/2$, and let S and D be the sampling and scaling matrix of Algorithm* Exactly(c) *[19] applied with $\{p_i\}_{i=1}^m$ and ϵ. Write $F = DS$, and compute*

$$\bar{B} = \arg\min_V ||FAV - FW||_F,$$

then

$$||A\bar{B} - W||_F \leq (1+\epsilon) \cdot \min_B ||AB - W||_F, \tag{3.6}$$

and thus

$$||W - A\bar{B}||_F \leq (1+\epsilon)\sqrt{1 + l(\frac{||W||\ ||((WH)_r)^+||}{1-\eta})^2} \cdot \bar{\sigma}_{r+1}(W). \tag{3.7}$$

Proof. Inequality (3.6) is due to [19, Theorem 5], which we modified and adapted in Chapter 5 as Theorem 5.1.

Recall that $||C_1^+|| \leq \frac{||W||\ ||((WH)_r)^+||}{1-\eta}$, and $||C_1^+||_F \leq \sqrt{l} \cdot ||C_1^+||$. Then inequality (3.7) follows from (3.6). □

Remark 3.4. *The multiplier F constructed from Lemma 3.1 has size $k \times m$, where $k = 3200 \cdot l^2 \cdot \epsilon^{-2}$. The cost of constructing F can be divided into two parts: the first part is computing the probability distribution, whose dominating cost is computing the SVD of A with $O(ml^2)$ arithmetic operations, and the second part is forming matrix F implicitly[2], whose cost is $O(l^2)$. Moreover, F is a matrix such that each row has exactly one entry that is non-zero, and thus FW can be computed in $O(l^2 n)$. Finally, computing $\bar{B}$ is solving n linear regression problems of size $k \times l$, and the total cost is $O(l^4 n)$. Combine the above arguments, and deduce that assuming $l^4 \ll m$, the cost of computing $\bar{B}$ is sublinear, given that $A = WH$ satisfying the requirement of Corollary 3.1.2.*

Now we present the **deterministic construction**, and such construction choose multipliers F by taking the leading submatrix of the permutation matrix found by matrix rank-revealing factorization, such as the rank-revealing LU factorization proposed in [51].

We start by showing how multiplier F affect the LRA error with the following Lemma.

Lemma 3.2. *Let A, B be the output of Algorithm 3.3, and assume that*

[2]Since matrix F is sparse, we can store it efficiently by leaving out zero entries. Also, we consider the cost of drawing a random value as a constant.

$B = (FA)^+FW$ *and* $m \geq k \geq l = \text{rank}(FA)$. *Then*

$$W - AB = X(W - AA^+B) \text{ for } X = I_m - A(FA)^+F, \tag{3.8}$$

moreover,

$$|W - AB| \leq |X|\ |W - AA^+M|,\ |X| \leq |I_m| + |A|\ |F|\ |(FA)^+|. \tag{3.9}$$

Proof. Recall that if $k \geq l$ and $\text{rank}(FA) = l$, according to the property of pseudo inverse, $(FA)^+FA = I_l$ is an identity matrix. Then (3.8) and (3.9) follows. □

Recall that the norm of $|W - AA^+W|$ has proven upper bound, given that WH satisfies the assumption stated in Corollary 3.1.2. Next we will show how to construct the appropriate sparse multiplier F, and show the upper bound of $||(FA)^+||$. In the rest of this section, we assume that A has been orthogonalized by finding an appropriate $l \times l$ matrix T, and let $A = WHT^{-1}$ as mentioned in Remark 3.1. This does not change the error bound of Corollary 3.1.2, because $W - AA^+W$ is invariant for either $A = MH$ or $A = MHT^{-1}$ under the aforementioned assumption.

Theorem 3.2. *Fix positive number* $h > 1$*, and apply [51, Algorithm 1] to an* $m \times l$ *orthogonal matrix* A*. The algorithm terminates in* $O(ml^2)$

flops and outputs an $l \times m$ subpermutation matrix F such that $||(FA)^+|| \leq \sqrt{(m-l)lh^2+1}$, and consequently

$$||X|| \leq ||I_m|| + ||A||\ ||F||\ ||(FA)^+|| \leq 1 + \sqrt{(m-l)lh^2+1}.$$

Remark 3.5. *With mild assumption that $l^2 \ll m$ and $l^3 \ll mn$, constructing such multiplier F has sublinear computational cost $O(ml^2)$, and the dominating cost of computing $B = (FA)^+FA$ is the cost for computing the pseudo inversion of FA, which is also sublinear at $O(l^3)$. Overall, pre-multiplying F accelerates the LRA computation to sublinear, and at the mean time the error bound from Corollary 3.1.2 is increased by a factor of $1 + \sqrt{(m-l)lh^2+1} = O(\sqrt{ml})$.*

3.4 Accuracy of Sublinear Cost Dual LRA Algorithms

In this section, we first introduce two models for random low numerical rank inputs, and then deduce the probabilistic output error bound on Algorithm 3.1 (Column Subspace Sampling) with these random inputs and any fixed full rank/orthogonal multiplier H. Recall the result of last section, we can deduce the probabilistic error bound of Algorithms 3.3 and 3.4 accordingly, and if the multiplier H is chosen to be sparse, the overall computation cost of Algorithms 3.3 and 3.4 is sublinear.

Our approach also supports error estimates for dense multipliers. We refer the interested readers to [33, Section7.4] and the bibliography therein.

Here and hereafter, we let $\stackrel{d}{=}$ denote *equal in distribution*. We deduce the error estimates with known results for norms of Gaussian matrices and pseudo inverse of Gaussian matrices, and we list these results in Appendix A.4.

Definition 3.1. *A random matrix with all entries being iid Standard Normal variables is called a* **Gaussian** *matrix. For simplicity, we let $G_{p\times q}$ denote a $p \times q$ Gaussian matrix.*

Theorem 3.3. [Non-degeneration of a Gaussian Matrix.] *Fix integers m, n, r such that $r \leq \min(m, n)$. Let F and H be independent $r \times m$ and $n \times r$ Gaussian matrices. Let $M \in \mathbb{R}^{m\times n}$ be any matrix such that $\operatorname{rank}(M) \geq r$. Then*

$$\operatorname{rank}(FM) = \operatorname{rank}(MH) = \operatorname{rank}(FMH) = r$$

almost surely[3]*.*

Assumption 3.1. *For the rest of the section, we omit events where Gaussian matrices degenerate by conditioning on all involved Gaussian matrices (and*

[3]With probability equals to 1.

products of full rank matrices with Gaussian matrices) having full rank. This does not affect our probability estimates.

3.4.1 Output Errors of Column Subspace Sampling for a Perturbed Factor-Gaussian Input

Assumption 3.2. *Let $m \times n$ matrix $\tilde{M} = AB$ be a right factor Gaussian matrix of rank r, where $A \in \mathbb{R}^{m\times r}$ has full rank, and B is a $r \times n$ Gaussian matrix. Let $H = U_H \Sigma_H V_H^T$ be a $n \times l$ test matrix with full rank and $l \geq r$, and let $\theta = \frac{e\sqrt{l}(\sqrt{n}+\sqrt{r})}{l-r}$ be a constant, where $e := 2.71828182\ldots$ is the base of natrual logarithm. Define random variables $\nu = ||B||$ and $\mu = ||(BU_H)^+||$, and recall that $\nu \overset{d}{=} ||G_{r\times n}||$ and $\mu \overset{d}{=} ||G^+_{r\times l}||$.*

Let E be a perturbation matrix with small norm, and consequently $M = \tilde{M} + E$ is a factor Gaussian matrix with small perturbation. Recall that $\tilde{M}$ has rank r, and let

$$\tilde{M} = U_{\tilde{M}}\ \Sigma_{\tilde{M}}\ V_{\tilde{M}}^T \tag{3.10}$$

be the compact SVD. Similarly, let M_r be the rank r truncation of M by setting the trailing singular values $\sigma_i(M)$ as 0 for all $i > r$, and let

$$M_r = U_r\ \Sigma_r\ V_r^T \tag{3.11}$$

be the compact SVD. Define

$$\tilde{C}_1 = V_{\tilde{M}}^T H \ \text{ and } \ C_1 = V_r^T H. \tag{3.12}$$

By the inequality of (3.4), we could obtain the error bound for Algorithm 3.1 Column Subspace Sampling, once we confirm that C_1 has rank r, and obtain an upper bound of the norm $||C_1^+||$. We first compute an upper bound for $||\tilde{C_1}^+||$, then we deduce an upper bound for $||C_1^+||$ with the assumption that $||E||_F$ is sufficiently small, and finally we estimate the output error of Algorithm 3.1 in Theorem 3.4.

Lemma 3.3. *Under assumption 3.2, let $V_{\tilde{M}}$ and $\tilde{C}_1$ be defined as in (3.10) and (3.12). Fix any positive number $\xi < 1/4$, the with probability no less than $1 - 2\sqrt{\xi}$,*

$$||\tilde{C_1}^+|| \leq \xi^{-1}\ \theta\ ||H^+|| \tag{3.13}$$

Proof. Let $B = U_B\ \Sigma_B\ V_B^T$ be the compact SVD of B. Then $V_{\tilde{M}}^T = QV_B^T$ for some $r \times r$ orthogonal matrix Q, because $V_{\tilde{M}}^T$ and V_B^T span the same linear space. Let $H = U_H\ \Sigma_H\ V_H^T$ be the SVD of H, then

$$\tilde{C}_1 = V_{\tilde{M}}H = Q\Sigma_B^{-1}U_B^T B U_H \Sigma_H V_H^T, \tag{3.14}$$

and thus we can obtain the following inequality

$$\sigma_r(\tilde{C}_1) \geq \sigma_r(Q\Sigma_B^{-1}U_B^T)\ \sigma_r(BU_H)\ \sigma_l(\Sigma_H V_H^T) \tag{3.15}$$

$$\geq \sigma_r(\Sigma_B^{-1})\ \sigma_r(BU_H)\ \sigma_l(H) \tag{3.16}$$

$$= ||B||^{-1}||BU_H^+||^{-1}||H^+||^{-1} \tag{3.17}$$

$$= \nu^{-1}\ \mu^{-1}\ ||H^+||^{-1} \tag{3.18}$$

Let $\xi < 1/4$ be a positive number, then according to Lemma A.5, with probability no less than $1 - 2\sqrt{\xi}$

$$\sigma_r(\tilde{C}_1) \geq \nu^{-1}\ \mu^{-1}\ ||H^+||^{-1} \geq \xi/\theta||H^+||^{-1} \tag{3.19}$$

and thus

$$||\tilde{C_1}^+|| \leq \xi^{-1}\ \theta\ ||H^+||. \tag{3.20}$$

□

Lemma 3.4. *Under assumption 3.2, let $V_{\tilde{M}}$, V_r, $\tilde{C}_1$, and C_1 be defined as in (3.10), (3.11) and (3.12). Fix positive number $\xi < 1/4$, and further assume that $||E||_F \leq 0.02\ \xi\ \frac{\sigma_r(A)}{\theta\ \kappa(H)}\ \frac{n-r}{e\sqrt{n}}$, then with probability no less than $0.9 - 2\sqrt{\xi}$*

$$||C_1^+|| \leq 5\ \xi^{-1}\ \theta\ ||H^+||. \tag{3.21}$$

Proof. Recall that

$$||C_1^+|| = ||(V_r^T H)^+|| = \left(\sigma_r(V_r^T H)\right)^{-1}$$

and that

$$\sigma_r(V_r^T H) \geq \sigma_r(V_{\tilde{M}}^T H) - ||(V_r^T - V_{\tilde{M}}^T)H||. \tag{3.22}$$

In Lemma 3.3, we show that with high probability $\sigma_r(V_{\tilde{M}}^T H) \geq \xi/\theta||H^+||^{-1}$, therefore it is enough to show that $||(V_r^T - V_{\tilde{M}}^T)H|| \leq ||V_r^T - V_{\tilde{M}}^T||\ ||H||$ is small (whp), given that the perturbation E is small.

Recall that $\sigma_{r+1}(\tilde{M}) = 0$ and $\sigma_r(\tilde{M}) \geq \sigma_r(A)\sigma_r(B)$. According to Theorem A.4 claim (iii), we have

$$\text{Prob}\left\{\sigma_r(B) < a \cdot \frac{n-r}{e\sqrt{n}}\right\} \leq a \quad \text{for any } 0 < a < 1. \tag{3.23}$$

Therefore, if $||E||_F \leq 0.02\ \sigma_r(A)\ \frac{n-r}{e\sqrt{n}}\ \Pi$ where $\Pi < 1$, then with probability no less than 0.9,

$$\sigma_r(\tilde{M}) \geq 0.1\sigma_r(A)\ \frac{n-r}{e\sqrt{n}} \quad \text{and} \quad \frac{||E||_F}{\sigma_r(\tilde{M})} \leq 0.2\ \Pi \leq 0.2. \tag{3.24}$$

Then apply Theorem A.1 on the impact of a perturbation of a matrix on its top singular spaces, and deduce that we can select the top r singular vectors of $\tilde{M}$ and M such that

$$||V_{\tilde{M}}^T - V_r^T|| \leq \frac{4||E||_F}{\sigma_r(\tilde{M}) - \sigma_{r+1}(\tilde{M})} = \frac{4||E||_F}{\sigma_r(\tilde{M}} \leq 0.8\ \Pi.$$

Combining the result from Lemma 3.3, we obtain the following inequality with probability no less than $0.9 - 2\sqrt{\xi}$

$$||(V_r^T - V_{\tilde{M}}^T)H|| \leq 0.8\ \Pi\ ||H||,$$

and according to (3.22),

$$\sigma_r(V_r^T H) \geq \xi/\theta||H^+||^{-1} - 0.8\ \Pi\ ||H|| \geq 0.2\xi/\theta||H^+||^{-1}$$

if we let $\Pi = \frac{\xi}{\theta\kappa(H)}$. This substitution is fine since $\theta, \kappa(H) \geq 1$, and $\xi < 1$, and consequently $\Pi < 1$. Finally, deduce that

$$||C_1^+|| \leq \left(\sigma_r(V_r^T H)\right)^{-1} \leq 5\ \xi^{-1}\ \theta\ ||H^+||. \tag{3.25}$$

□

Remark 3.6. *The assumption* $||E||_F \leq 0.02\ \xi\ \frac{\sigma_r(A)}{\theta\ \kappa(H)}\ \frac{n-r}{e\sqrt{n}}$ *is quite reasonable. Matrix* M *is constructed such that it should have numerical rank* r*, and is sufficiently close to a rank* r *matrix. These would imply that it is reasonable to have* $|E|$ *be a fraction of* $\sigma_r(\tilde{M})$*, which on average is no less than* $\sigma_r(A) \cdot O(\frac{n-r}{e\sqrt{n}})$*. Furthermore, if we select* H *to be orthogonal, for example a subpermutation matrix, then* $\kappa(H) = 1$*, and it does not contribute to the bound on perturbation norm. Lastly, given that* l *is sufficiently large comparing to* r*, factor* $1/\theta = O(\frac{\sqrt{l}}{\sqrt{n}})$ *is still quite substantial.*

Theorem 3.4. [Errors of Algorithm 3.1 Column Subspace Sampling for a perturbed factor-Gaussian matrix.] *Under assumption 3.2, let* $M = \tilde{M}+E$ *be a right factor Gaussian with perturbation such that* $||E||_F \leq 0.02\ \xi\ \frac{\sigma_r(A)}{\theta\ \kappa(H)}\ \frac{n-r}{e\sqrt{n}}$*, where* $0 < \xi < 1/4$ *is a constant parameter. Apply Algorithm 3.1 to* M *with*

test matrix H, and let the output factors be $A = MH$ and $B = AA^+M$ respectively. Then with probability no less than $0.9 - 2\sqrt{\xi}$,

$$||M - AB||^2 \leq \Big(1 + \phi^2\Big)\sigma_{r+1}^2(M), \tag{3.26}$$

where $\phi := 5\ \xi^{-1}\ \theta\ ||H^+||$.

Proof. Let $V_{\tilde{M}}$, V_r, $\tilde{C}_1$, and C_1 be defined as in (3.10), (3.11) and (3.12). Apply Lemma 3.3 and 3.4, and deduce that $C_1 = V_r^T H$ has full rank r, and that $||C_1^+|| \leq 5\ \xi^{-1}\ \theta\ ||H^+|| = \phi$ with probability no less than $0.9 - 2\sqrt{\xi}$. Recall from Theorem 3.1 that

$$||M - AB|| \leq \big(1 + ||C_1^+||^2\big)^{1/2}\ \sigma_{r+1}(M),$$

and (3.26) follows. □

Remark 3.7. *Given that l is sufficient large comparing to r, and that H is orthogonal or close to orthogonal, the output error bound is in*

$$\big(1 + O(\frac{\sqrt{n}}{\sqrt{l}})\big)\ \sigma_r(M),$$

which is quite reasonable, and is not too much larger than the error bound of the optimal LRA considering the computation can be extended to sublinear cost Algorithm if H is chosen to be sparse.

3.4.2 Output Errors of Column Subspace Sampling for a Matrix with a Random Singular Space

In this subsection we present the error bound of Algorithm 3.1 Column Subspace Sampling where the input low numerical rank matrix is under a different model where the singular space is randomized.

Theorem 3.5. [Errors of Range Finder for an input with a random singular space.] *Let $n \times r$ matrix V_1 in Theorem 3.1 be an orthogonalization of a $n \times r$ Gaussian matrix G (for example, the Q factor of the QR factorization of G). Let $n \times l$ multiplier H have full rank $l \geq r$, and let $H = U_H \, \Sigma_H \, V_H$ be SVD. Apply Algorithm 3.1 to matrix M and multiplier H, and let $A = MH$, $B = AA^+M$ be the output. Then for $n \geq l \geq r + 4 \geq 6$, and $0 < \xi < 1/4$,*

(i) with probability no less than $1 - 2\sqrt{\xi}$,

$$||M - AB|| \leq (1 + \phi^2)^{1/2} \sigma_{r+1}(M), \tag{3.27}$$

where $\phi = \xi^{-1} \; e \; ||H^+|| \; \frac{\sqrt{l}(\sqrt{n}+\sqrt{r})}{l-r}$.

(ii) with probability no less than $1 - 2\sqrt{\xi}$,

$$||M - AB||_F \leq (1 + \psi^2)^{1/2} \sqrt{\sum_{i=r+1}^{\min(m,n)} \sigma_i^2(M)}, \tag{3.28}$$

where $\psi = \xi^{-1} \; r \; ||H^+||_F \; \sqrt{\frac{n}{l-r-1}}$.

Proof. Let $C_1 = V_1 H$, and recall that C_1 has full rank r with probability 1.

Apply Theorem 3.1, and we can easily deduce claims (i) and (ii) by showing $||C_1^+|| \le \phi$ and $||C_1^+|| \le \psi$ with high probability.

Without loss of generality, write $G = V_1 R$, where R is a $r \times r$ matrix sharing the singular values with G, that is $\sigma_i(R) = \sigma_i(G)$ for $i = 1, ..., r$. Then

$$C_1 = V_1^T H = (R^T)^{-1} G^T U_H \Sigma_H V_H^T,$$

and therefore we have

$$|C_1^+| \le |R|\ |(G^T U_H)^+|\ |\Sigma_H^{-1}| = |R|\ |(G^T U_H)^+|\ |H^+|.^4 \tag{3.29}$$

Define random variables $\mu = ||R||$, $\mu_F = ||R||_F$, $\nu = ||(G^T U_H)^+||$, and $\nu_F = ||(G^T U_H)^+||_F$. By orthogonal invariant property of Gaussian matrices, $G^T U_H$ has the distribution of a $r \times l$ Gaussian matrix. Let $G_{p\times q}$ denote a $p \times q$ Gaussian matrix, and we have $\mu \overset{d}{=} ||G_{n\times r}||$, $\mu_F \overset{d}{=} ||G_{n\times r}||_F$, $\mu \overset{d}{=} ||G^+_{r\times l}||$, and $\mu_F \overset{d}{=} ||G^+_{r\times l}||_F$.

Recall from Theorem A.3 and A.4 that

$$\mathbb{E}\ \nu \le \sqrt{n} + \sqrt{r} \ \text{ and } \ \mathbb{E}\ \nu_F \le \sqrt{nr}, \tag{3.30}$$

and

$$\mathbb{E}\ \mu \le \frac{e\ \sqrt{l}}{l - r} \ \text{ and } \ \mathbb{E}\ \mu_F \le \sqrt{\frac{r}{l - r - 1}} \tag{3.31}$$

[4]The inequality holds of all $|\cdot|$ being Spectral or Frobenius norm at the same time.

ν, ν_F, μ, and μ_F has low probability being much greater than their expected value. Let $1 > \xi > 0$ be a number, and apply Markov Inequality and union bound similar to Lemma A.5 , and deduce that

$$\text{Prob}\{\nu\mu \leq \xi^{-1}\frac{e\sqrt{l}(\sqrt{n}+\sqrt{l})}{l-r}\} \geq 1 - 2\sqrt{\xi} \tag{3.32}$$

and

$$\text{Prob}\{\nu_F\mu_F \leq \xi^{-1}\ r\ \sqrt{\frac{n}{l-r-1}}\} \geq 1 - 2\sqrt{\xi}. \tag{3.33}$$

Note that although ν and μ (as well as ν_F and μ_F) are dependent, the dependency does not affect our union bound. Finish the proof by combining the above two probability estimates with inequality (3.29). □

The output error of Algorithm 3.3 can be deduced by combining the results of this section and results of Section 3.3.2, and the output error of Algorithm 3.4 can be deduced similarly.

3.5 Numerical tests

In this section we present the test results of Algorithm 3.4 on four types of inputs consist of synthetic and real-world data with varying spetrums. We measure the performance of Algorithm 3.4 by computing the relative error ratio,

$$r = \frac{||M - \tilde{M}||_F}{||M - M_\rho||_F}, \tag{3.34}$$

where M denotes the input matrix, $\tilde{M}$ denotes the approximation output by Algorithm 3.4, and M_ρ denotes the ρ-top SVD of M. Here ρ is a rank parameter pre-determined for each input matrix. The relative error ratio r is greater or equal to 1, ignoring rounding errors, when rank($\tilde{M}$)$\leq \rho$. However in our experiments, we increase the rank of the approximation after each iteration, which may result in r being less than 1. The result shows that we consistently achieved a relative error ratio approximately 1 or less than 1, indicating that upon termination, Algorithm 3.4 output accurate low rank approximations, which is in good accordance to our theoretical analysis.

The algorithm was implemented in Python, and all experiments were run on a 64bit MacOS Sierra 10.12.6 machine with 1.6GHz CPU and 4GB Memory. We called scipy.linalg version 0.4.9 for numerical linear algebra routines such as QR factorization with pivoting, Moore-Penrose matrix inversion and linear least squares regression.

Synthetic Input: Synthetic inputs consist of two types of random inputs, one with rapidly decaying spectrum and one with slow decaying spectrum. Both types of random matrices are of size 1024×1024, and are constructed through product $U\Sigma V^T$, where U and V are the left and right singular vectors of a random Gaussian matrix. In the case with rapidly decaying spectrum, $\Sigma = \text{diag}(v)$, where $v_i = 1$ for $i = 1, 2, 3, \ldots 40$, $v_i = \frac{1}{2}^i$ for $i = 41, \ldots, 100$,

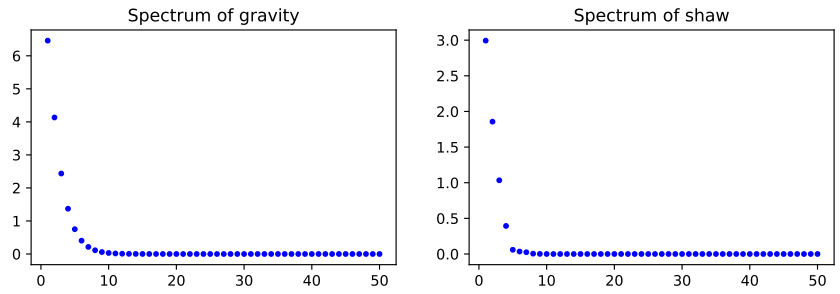

Figure 3.1: Spectrums of Real World Input Matrices

and $v_i = 0$ for $i > 100$. For the one with slowly decaying spectrum, $\Sigma = \text{diag}(u)$, where $u_i = 1$ for $i = 1, 2, 3, \ldots 40$, and $u_i = \frac{1}{1+i}$ for $i > 40$.

Real-wold Input: The input matrices used in this category are 1000×1000 dense matrices with real values. They are with low numerical rank, and they are constructed by discretizing Integral Equations, and provided in the built-in problems of the Regularization Tools [5].

The two test matrices we used are namely **gravity**, which is from a one-dimensional gravity surveying model problem, and **shaw**, which is from a one-dimensional image restoration model problem. Their distribution of singular values are displayed in figure 3.1 and for simplicity, we padded these two matrices with 0 to increase their size to 1024×1024.

We use **iterative refinement** method to control the residual error, and more specifically, in the i-th iteration step we draw two multipliers F and H,

[5]For more details see Chapter 4 of
http://www.imm.dtu.dk/~pcha/Regutools/RTv4manual.pdf

then approximate the residual $R_i = M - \tilde{M}_{i-1}$ by $\tilde{R}_i = R_i H (F R_i H)^+ F R_i$ as instructed in Algorithm 3.4, and finally compute the i-th approximation $\tilde{M}_i = \tilde{M}_{i-1} + \tilde{R}_i$. In our tests, we used the *abridged* SRHT multipliers[6], with size 5×1024 and recursion depth 3. We also included results with Gaussian multipliers as suggested in [63] for comparison.

For each input matrix, we iteratively apply algorithm 3.4 to the approximation residual R_i for $i = 0, 1, \ldots, 100$ and recorded the mean relative-error ratio for every iteration step in figure 3.2. We notice that the abridged SRHT multipliers performed similarly compared to Gaussian multipliers in our tests, and are only slightly worse in few places. However, this is a reasonable price since abridged SRHT multipliers are very sparse and only access a small fraction of the input matrix each iteration step. We also notice that in the test with random input having slowly decaying spectrum, the relative error ratio did not decrease to less than 1, which could be caused by the "heavier"" tail in the spectrum. We acknowledge that for some inputs that are not "well-mixed", it is necessary to increase the recursion depth in order to achieve accurate approximation.

[6]The construction of multipliers of this type follows from [54] and [54].

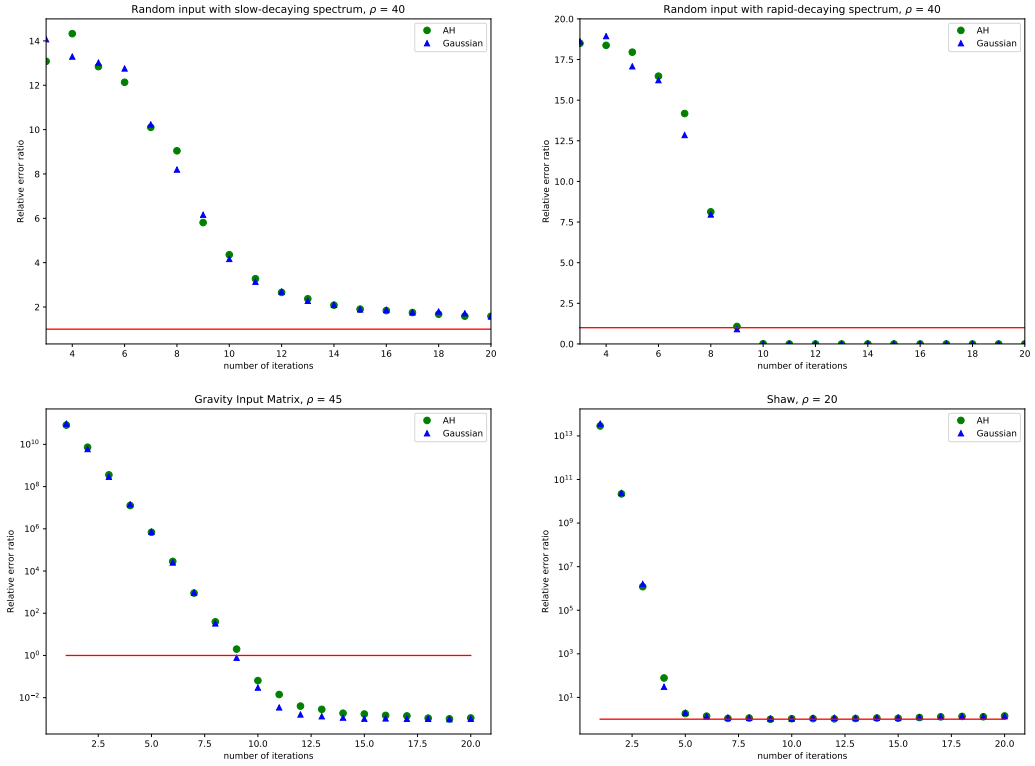

Figure 3.2: Test Result for Algorithm 3.4

Chapter 4

CUR Low Rank Approximation Based on Volume Maximization

4.1 Background

4.1.1 CUR LRA

We use basic definitions for matrix computations recalled in Appendix A.1. We simplify our presentation by confining it to the case of real matrices, but the extension to the case of complex matrices is straightforward.

CUR LRA of a matrix W of numerical rank at most r is defined by three matrices C, U, and R, with C and R made up of l columns and k rows of

Portions of this chapter previously appeared in our work [43].

W, respectively, $U \in \mathbb{R}^{l\times k}$ said to be the *nucleus* of CUR LRA,[1]

$$0 < r \le k \le m,\ r \le l \le n,\ kl \ll mn, \tag{4.1}$$

$$W = CUR + E,\ \text{and}\ ||E||/||W|| \le \epsilon,\ \text{for a small tolerance}\ \epsilon > 0. \tag{4.2}$$

CUR LRA is a special case of LRA of (1.1) where, say, $A = LU$, $B = R$, and $k = l = r$. Conversely, given LRA of (1.1) one can compute CUR LRA of (4.2) at linear cost (see [53] and [56]).

Define a *canonical* CUR LRA as follows.

(i) Fix two sets of columns and rows of W and define its two submatrices C and R made up of these columns and rows, respectively.

(ii) Define the $k \times l$ submatrix $W_{k,l}$ made up of all common entries of C and R, and call it a *CUR generator*.

(iii) Compute its rank-r truncation $W_{k,l,r}$ by setting to 0 all its singular values, except for the r largest ones.

(iv) Compute the Moore–Penrose pseudo inverse $U =: W^{+}_{k,l,r}$ and call it the *nucleus* of CUR LRA of the matrix W (cf. [19], [50]); see an alternative choice of a nucleus in [45]).

[1]The pioneering papers [66], [30], [31], [28], [29], [27], [70], and [50] define CGR approximations having nuclei G; "G" can stand, say, for "germ". We use the acronym CUR, more customary in the West. "U" can stand, say, for "unification factor", and we notice the alternatives of CNR, CCR, or CSR with N, C, and S standing for *"nucleus", "core",* and *"seed"*.

Notice that $W_{r,r} = W_{r,r,r}$, and if a CUR generator $W_{r,r}$ is nonsingular, then $U = W_{r,r}^{-1}$.

4.1.2 Matrix Volumes and the Hadamard's Bound

Definition 4.1. *For a triple of integers k, l, and r such that $1 \le r \le \min\{k,l\}$, the* volume $v_2(M)$ *and the* r-projective volume $v_{2,r}(M)$ *of a $k \times l$ matrix M are defined as follows:*

$$v_2(M) := \prod_{j=1}^{\min\{k,l\}} \sigma_j(M), \quad v_{2,r}(M) := \prod_{j=1}^{r} \sigma_j(M), \tag{4.3}$$

$$v_{2,r}(M) = v_2(M) \text{ if } r = \min\{k,l\}, \tag{4.4}$$

$v_2^2(M) = \det(MM^*)$ if $k \ge l$; $v_2^2(M) = \det(M^*M)$ if $k \le l$, $v_2^2(M) = |\det(M)|^2$ if $k = l$, *and* $\sigma_j(M)$ *denotes the jth largest singular value of M (cf. Appendix A.1).*

By following [13], [66], [30], [31], [32], [51], [28], [27], [29], [70], and [50], we use the concepts of volume and projective volume in our study of CUR LRA; [5] shows some distinct applications of the concept of projective volume.

Definition 4.2. *The volume of a $k \times l$ submatrix $W_{\mathcal{I},\mathcal{J}}$ of a matrix W is* h-maximal *over all $k \times l$ submatrices if it is maximal up to a factor of h.*

The volume $v_2(W_{\mathcal{I},\mathcal{J}})$ *is* column-wise *(resp.* row-wise*) h-maximal if it is h-maximal in the submatrix* $W_{\mathcal{I},:}$ *(resp.* $W_{:,\mathcal{J}}$*). The volume of a submatrix* $W_{\mathcal{I},\mathcal{J}}$ *is* column-wise *(resp.* row-wise*)* locally *h-maximal if it is h-maximal over all submatrices of* W *that differ from the submatrix* $W_{\mathcal{I},\mathcal{J}}$ *by a single column (resp. single row). Call volume* (h_c, h_r)-maximal *if it is both column-wise* h_c*-maximal and row-wise* h_r*-maximal. Likewise define* locally (h_c, h_r)-maximal *volume. Write* maximal *instead of* 1*-maximal and* $(1,1)$*-maximal in all these definitions. Extend all these definitions to* r*-projective volumes.*

For a $k \times l$ matrix $M = (m_{ij})_{i,j=1,1}^{k,l}$ write $\mathbf{m}_j := (m_{ij})_{i=1}^{k}$ and $\bar{\mathbf{m}}_i := ((m_{ij})_{j=1}^{l})^*$ for all i and j. For $k = l = r$ recall the *Hadamard's bound*

$$v_2(M) = |\det(M)| \le \min\ \{\prod_{j=1}^{r} ||\mathbf{m}_j||,\ \prod_{i=1}^{r} ||\bar{\mathbf{m}}_j^*||,\ r^{r/2} \max_{i,j=1}^{r} |m_{ij}|^r\}. \quad (4.5)$$

4.1.3 The Impact of Volume Maximization on CUR LRA

The estimates of the two following theorems in the Chebyshev matrix norm $||\cdot||_C$ increased by a factor of $\sqrt{mn}$ turn into estimates in the Frobenius norm $||\cdot||_F$ (see (A.6)).

Theorem 4.1. *[50].*[2] *Suppose that* $r := \min\{k, l\}$*,* $W_{\mathcal{I},\mathcal{J}}$ *is the* $k \times l$ *CUR*

[2]The theorem first appeared in [28, Corollary 2.3] in the special case where $k = l = r$

generator, $U = W^{+}_{\mathcal{I},\mathcal{J}}$ *is the nucleus defining a canonical CUR LRA of an* $m \times n$ *matrix* W*,* $E = W - CUR$*,* $h \geq 1$*, and the volume of* $W_{\mathcal{I},\mathcal{J}}$ *is locally h-maximal, that is,*

$$h\ v_2(W_{\mathcal{I},\mathcal{J}}) = \max_{B} v_2(B)$$

where the maximum is over all $k \times l$ *submatrices* B *of the matrix* W *that differ from* $W_{\mathcal{I},\mathcal{J}}$ *in at most one row and/or column. Then*

$$||E||_C \leq h\ f(k,l)\ \sigma_{r+1}(W) \quad \text{for} \quad f(k,l) := \sqrt{\frac{(k+1)(l+1)}{|l-k|+1}}.$$

Theorem 4.2. *[50]. Suppose that* $W_{k,l} = W_{\mathcal{I},\mathcal{J}}$ *is a* $k \times l$ *submatrix of an* $m \times n$ *matrix* W*,* $U = W^{+}_{k,l,r}$ *is the nucleus of a canonical CUR LRA of* W*,* $E = W - CUR$*,* $h \geq 1$*, and and the r-projective volume of* $W_{\mathcal{I},\mathcal{J}}$ *is locally h-maximal, that is,*

$$h\ v_{2,r}(W_{\mathcal{I},\mathcal{J}}) = \max_{B} v_{2,r}(B)$$

where the maximum is over all $k \times l$ *submatrices* B *of the matrix* W *that differ from* $W_{\mathcal{I},\mathcal{J}}$ *in at most one row and/or column. Then*

$$||E||_C \leq h\ f(k,l,r)\ \sigma_{r+1}(W) \quad \text{for} \quad f(k,l,r) := \sqrt{\frac{(k+1)(l+1)}{(k-r+1)(l-r+1)}}.$$

Remark 4.1. *Theorems 4.1 and 4.2 have been stated in [50] under assumptions that that the matrix* $W_{\mathcal{I},\mathcal{J}}$ *has (globally) h-maximal volume or*

---

and $m = n$.

r-projective volume, respectively, but their proofs in [50] support the above extensions.

Observe the following corollary of Theorem B.2.

Corollary 4.2.1. *Suppose that $BW = (BU|BV)$ for a nonsingular matrix B and that the submatrix U is h-maximal in the matrix $W = (U|V)$. Then the submatrix BU is h-maximal in the matrix BW.*

4.2 C–A Iterations

Next we describe C–A iterations by involving two auxiliary Subalgorithms $\mathcal{A}$ and $\mathcal{B}$.

Given a 4-tuple of integers k, l, p, and q such that $r \le k \le p$ and $r \le l \le q$ subalgorithm $\mathcal{A}$ is applied to $p \times q$ matrix and computes its $k \times l$ submatrix whose volume or projective volume is maximal up to a fixed factor $h \ge 1$ among all its $k \times l$ submatrices.

Subalgorithm $\mathcal{B}$ verifies whether the error norm of the CUR LRA built on a fixed CUR generator is within a fixed tolerance τ (see [53] on some verification recipes).

For simplicity one can first consider the C-A iterations in the case where $k = l = r$ (see Figure 4.1, borrowed from [56]).

Algorithm 4.1: C–A Iterations

Input: $W \in \mathbb{C}^{m\times n}$, $r, k, l,$ ITER > 0 be integers, and τ a positive number.

Output: A CUR LRA of W with error norm at most τ or FAILURE.

Initialization: Fix a submatrix W_0 made up of l columns of W, and obtain the initial set $\mathcal{I}_0$.

for $i = 1, 2, \dots,$ ITER **do**
  if i is even **then**
    "Horizontal" C–A step:
    1. Let $R_i := W_{\mathcal{I}_{i-1},:}$ be the $k \times n$ row submatrix of W.
    2. Apply Subalgorithm $\mathcal{A}$ to R_i and obtain $k \times l$ submatrix $W_i = W_{\mathcal{I}_{i-1},\mathcal{J}_i}$.
  else
    "Vertical" C–A step:
    1. Let $C_i := W_{:,\mathcal{J}_{i-1}}$ be the $m \times l$ column submatrix of W.
    2. Apply Subalgorithm $\mathcal{A}$ to C_i and obtain $k \times l$ submatrix $W_i = W_{\mathcal{I}_i,\mathcal{J}_{i-1}}$.
  end if
  Apply subalgorithm $\mathcal{B}$ and obtain E, the error bound of CUR LRA built from the generator W_i.
  if $E \le \tau$ **then**
    return CUR LRA built from the generator W_i.
  end if
end for
return Failure

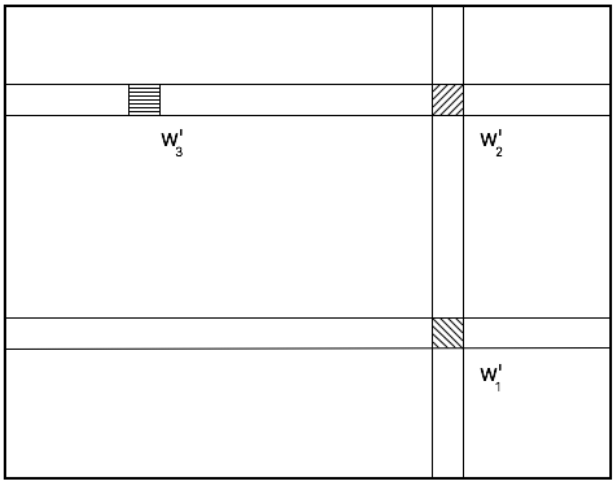

Figure 4.1: The Three Successive C–A Steps Output Three Striped Matrices.

4.3 CUR LRA by Means of C–A Iterations

We can apply C–A steps by choosing deterministic algorithms of [32] for Sub-algorithm $\mathcal{A}$. In this case ml and kn memory cells and $O(ml^2)$ and $O(k^2n)$ flops are involved in "vertical" and "horizontal" C-A iterations, respectively. They run at sublinear cost if $k^2 = o(m)$ and $l^2 = o(n)$ and output submatrices having h-maximal volumes for h being a low degree polynomial in $m+n$. Every iteration outputs a matrix that has locally h-maximal volume in a "vertical" or "horizontal" submatrix, and the hope is to obtain globally $\bar{h}$-maximal submatrix (for reasonably bounded $\bar{h}$) when maximization is

performed recursively in alternate directions.

Remark 4.2. *Algorithms of [51] do the same as those of [32], although they square the h of [32]. Empirically the algorithms of both [32] and [51] are superseded by the algorithm* maxvol *of [27].*

Of course, the contribution of C-A step is nil where it is applied to a $p \times q$ input whose volume is 0 or nearly vanishes compared to the target maximum, but the consistent success of C-A iterations in practice suggests that in a small number of loops such a degeneration is regularly avoided.

Next we show that already two successive C-A iterations output a CUR generator having h-maximal volume and r-projective volume (for any $h > 1$) in the case where the iterations begin at a $p \times q$ submatrix of W that shares its rank $r > 0$ with W. By continuity of the volume the result is extended to small perturbations of such matrices within a norm bound which we estimate in Theorem B.1.

In the next subsection we consider *the worst case input matrix* W of a rank r and two successive C-A steps initiated at its two submatrices of rank r. In this case we prove that the $k \times l$ output matrix $W_{k,l}$ for $\min\{k, l\} = r$ has locally $\bar{h}$-maximal volume,

In Section 4.3.2 we extend this result to the maximization of r-projective

volume rather than the volume of a CUR generator. (Theorem 4.2 shows benefits of such a maximization.)

In Section 4.3.3 we summarize our study in this section and comment on the estimated and empirical performance of C–A iterations.

4.3.1 Volume of the output of a C–A loop

First we compare SVDs of two matrices W and W^+ and obtain the following lemma.

Lemma 4.1. *$\sigma_j(W)\sigma_{\mathrm{rank}(W)+1-j}(W^+) = 1$ for all matrices W and all subscripts j, $j \le \mathrm{rank}(W)$.*

Corollary 4.2.2. *$v_2(W)v_2(W^+) = 1$ and $v_{2,r}(W)v_{2,r}(W_r^+) = 1$ for all matrices W of full rank and all integers r such that $1 \le r \le \mathrm{rank}(W)$.*

We are ready to prove that a $k \times l$ submatrix of rank r that has (h, h')-locally maximal nonzero volume in a rank-r matrix W has hh'-maximal volume globally in W, that is, over all $k \times l$ submatrices of W.

Theorem 4.3. *Suppose that the volume of a $k \times l$ submatrix $W_{\mathcal{I},\mathcal{J}}$ is nonzero and (h, h')-maximal in a matrix W for $h \ge 1$ and $h' \ge 1$ where $\mathrm{rank}(W) = r = \min\{k, l\}$. Then this volume is hh'-maximal over all its $k \times l$ submatrices of the matrix W.*

Proof. The matrix $W_{\mathcal{I},\mathcal{J}}$ has full rank because its volume is nonzero.

Fix any $k \times l$ submatrix $W_{\mathcal{I}',\mathcal{J}'}$ of the matrix W, recall that $W = CUR$, and obtain that

$$W_{\mathcal{I}',\mathcal{J}'} = W_{\mathcal{I}',\mathcal{J}} W^{+}_{\mathcal{I},\mathcal{J}} W_{\mathcal{I},\mathcal{J}'}.$$

If $k \le l$, then first apply claim (iii) of Theorem B.2 for $G := W_{\mathcal{I}',\mathcal{J}}$ and $H := W^{+}_{\mathcal{I},\mathcal{J}}$; then apply claim (i) of that theorem for $G := W_{\mathcal{I}',\mathcal{J}} W^{+}_{\mathcal{I},\mathcal{J}}$ and $H := W_{\mathcal{I},\mathcal{J}'}$ and obtain that

$$v_2(W_{\mathcal{I}',\mathcal{J}}) = v_2(W_{\mathcal{I}',\mathcal{J}} W^{+}_{\mathcal{I},\mathcal{J}} W_{\mathcal{I},\mathcal{J}'}) \le v_2(W_{\mathcal{I}',\mathcal{J}}) v_2(W^{+}_{\mathcal{I},\mathcal{J}}) v_2(W_{\mathcal{I},\mathcal{J}'}).$$

If $k > l$ deduce the same bound by applying the same argument to the matrix equation

$$W^{T}_{\mathcal{I}',\mathcal{J}'} = W^{T}_{\mathcal{I},\mathcal{J}'} W^{+T}_{\mathcal{I},\mathcal{J}} W^{T}_{\mathcal{I}',\mathcal{J}}.$$

Combine this bound with Corollary 4.2.2 for W replaced by $W_{\mathcal{I},\mathcal{J}}$ and deduce that

$$v_2(W_{\mathcal{I}',\mathcal{J}'}) = v_2(W_{\mathcal{I}',\mathcal{J}} W^{+}_{\mathcal{I},\mathcal{J}} W_{\mathcal{I},\mathcal{J}'}) \le v_2(W_{\mathcal{I}',\mathcal{J}}) v_2(W_{\mathcal{I},\mathcal{J}'}) / v_2(W_{\mathcal{I},\mathcal{J}}). \quad (4.6)$$

Recall that the matrix $W_{\mathcal{I},\mathcal{J}}$ is (h, h')-maximal and conclude that

$$h v_2(W_{\mathcal{I},\mathcal{J}}) \ge v_2(W_{\mathcal{I},\mathcal{J}'}) \text{ and } h' v_2(W_{\mathcal{I},\mathcal{J}}) \ge v_2(W_{\mathcal{I}',\mathcal{J}}).$$

Substitute these inequalities into the above bound on $v_2(W_{\mathcal{I}',\mathcal{J}'})$ and obtain that $v_2(W_{\mathcal{I}',\mathcal{J}'}) \le h h' v_2(W_{\mathcal{I},\mathcal{J}})$. □

4.3.2 From maximal volume to maximal r-projective volume

Recall that the CUR LRA error bound of Theorem 4.1 is strengthened when we shift to Theorem 4.2, that is, maximize r-projective volume for $r < k = l$ rather than the volume. Next we reduce maximization of r-projective volume of a CUR generators to volume maximization.

Corollary 4.2.1 implies the following lemma.

Lemma 4.2. *Let M and N be a pair of $k \times l$ submatrices of a $k \times n$ matrix and let Q be a $k \times k$ unitary matrix. Then $v_2(M)/v_2(N) = v_2(QM)/v_2(QN)$, and if $r \le \min\{k, l\}$ then also $v_{2,r}(M)/v_{2,r}(N) = v_{2,r}(QM)/v_{2,r}(QN)$.*

The submatrices[3] R' and $\begin{pmatrix} R' \\ O \end{pmatrix}$ of R have maximal volume and maximal r-projective volume in the matrix R, respectively, by virtue of Theorem B.2 and because $v_2(R) = v_{2,r}(R) = v_{2,r}(R')$. Therefore the submatrix $W_{:,\mathcal{J}}$ has maximal r-projective volume in the matrix W by virtue of Lemma 4.2.

Remark 4.3. *By transposing a horizontal input matrix W and interchanging the integers m and n and the integers k and l we extend the algorithm to computing a $k \times l$ submatrix of maximal or nearly maximal r-projective volume in an $m \times l$ matrix of rank r.*

[3]One can apply other rank-revealing factorizations instead. In all these variants the algorithm performs at sublinear cost if $n \gg (k + l)^2$.

Algorithm 4.2: From the maixmal volume to the maximal r-projective volume

Input: Integers k, l, n, and r, such that $0 < r \le k$ and $r \le l \le n$; $k \times n$ matrix W of rank r; a black box algorithm that finds a $r \times l$ submatrix with maximum volume in a $r \times n$ matrix of full rank r.
Output: A column set $\mathcal{J}$ such that $W_{:,\mathcal{J}}$ has maximal r-projective volume in W.

1. Compute a rank-revealing QRP factorization $W = QRP$, where Q is unitary, P is a permutation matrix, $R = \begin{pmatrix} R' \\ O \end{pmatrix}$, and R' is a $r \times n$ matrix (See [26, Sections 5.4.3 and 5.4.4] and [32].)
2. Compute a $r \times l$ submatrix $R'_{:,\mathcal{J}}$ of R' having maximal volume.

return $\mathcal{J}'$, such that $P : \mathcal{J}' \longrightarrow \mathcal{J}$.

4.3.3 Complexity and Accuracy of a Two-Step C–A Loop

The following theorem summarizes our study in this section.

Theorem 4.4. *Given five integers k, l, m, n, and r such that $r \le k \le m$ and $r \le l \le n$, suppose that two successive C–A steps (say, based on the algorithms of [32] or [51]) combined with Algorithm 4.2 have been applied to an $m \times n$ matrix W of rank r and have output $k \times l$ submatrices W'_1 and $W'_2 = W_{\mathcal{I}_2,\mathcal{J}_2}$ with nonzero r-projective column-wise locally h-maximal and nonzero r-projective row-wise locally h'-maximal volumes, respectively. Then the submatrix W'_2 has locally $h'h$-maximal r-projective volume in the matrix*

W.

In this section we arrived at a C–A algorithm that computes a CUR approximation of a rank-r matrix W. Let us summarize our study by combining Theorems 4.1, 4.2, and 4.4.

Corollary 4.4.1. *Under the assumptions of Theorem 4.4 apply a two-step C–A loop to an $m \times n$ matrix W and suppose that both its C–A steps output $k \times l$ submatrices having nonzero r-projective column-wise and row-wise locally h-maximal volumes (see Remarks 4.2 and 4.4). Build a canonical CUR LRA on a CUR generator $W'_2 = W_{k,l}$ of rank r output by the second C–A step. Then*

(i) the computation of this CUR LRA by using the auxiliary algorithms of [32] or [51] involves $(m+n)r$ memory cells and $O((m+n)r^2)$ flops[4] and

(ii) the error matrix E of the output CUR LRA satisfies the bound

$$||E||_C \le g(k,l,r)\ \bar{h}\ \sigma_{r+1}(W)$$

for $\bar{h}$ of Theorem 4.4 and $g(k,l,r)$ denoting the functions $f(k,l)$ of Theorem 4.1 or $f(k,l,r)$ of Theorem 4.2. In particular $||E||_C \le 2hh'\sigma_2(W)$ for $k = l = r = 1$.

[4]For $r = 1$ an input matrix turns into a vector of dimension m or n, and then we compute its absolutely maximal coordinate just by applying $m-1$ or $n-1$ comparisons, respectively.

Remark 4.4. *Theorem B.1 enables us to extend Algorithm 4.2, Theorem 4.4, and Corollary 4.4.1 to the case of an input matrix W of numerical rank r provided that the volume of the $k \times n$ input submatrix of C–A iterations stays nonzero in the transition from this matrix W to its LRA W'.*

4.4 Sublinear Cost CUR LRA for SPSD with Guaranteed Error Bound

For SPSD matrices we can improve our estimates of Theorem B.1 a little by applying Wielandt–Hoffman theorem (see [26, Theorem 8.6.4]), but we are going to compute reasonably close CUR LRA of an SPSD matrix at sublinear cost with no restriction on its distance from a low rank matrix.

4.4.1 Two Main Theorems

Theorem 4.5. (SPSD CUR LRA via Locally Max-Vol Block) *Suppose that $A \in \mathbb{R}^{n \times n}$ is an SPSD matrix, r and n are two positive integers, $r < n$, ξ is a a positive number, and $\mathcal{I}$ is the output of Algorithm 4.6. Write $C := A_{:,\mathcal{I}}$, $U := A^{-1}_{\mathcal{I},\mathcal{I}}$, and $R := A_{\mathcal{I},:}$. Then*

$$||A - CUR||_C \le (1 + \xi)(r + 1)\sigma_{r+1}(A). \tag{4.7}$$

Furthermore the computation of Algorithm 4.6 involves $O(nr^4 \log r)$ flops.

Theorem 4.6. (SPSD CUR LRA via Locally Maximal r-Projective Volume Block) *Suppose that $A \in \mathbb{R}^{n\times n}$ is an SPSD matrix; r, K, and n are three positive integers where $r < K < n$; ξ is a a positive number, and $\mathcal{I}$ is the output of Algorithm 4.6. Write $C := A_{:,\mathcal{I}}$, $U := (A_{\mathcal{I},\mathcal{I}})_r^+$, and $R := A_{\mathcal{I},:}$. Then*

$$||A - CUR||_C \leq (1+\xi)\frac{K+1}{K-r+1}\sigma_{r+1}(A). \tag{4.8}$$

In particular, let $K = cr - 1$ where $c > 1$, then

$$||A - CUR||_C \leq (1+\frac{1}{c-1})(1+\xi)\ \sigma_{r+1}(A). \tag{4.9}$$

The computation of Algorithm 4.6 involves $O(r^2K^4n + rK^4n\log n)$ flops.

4.4.2 Proof of Theorem 4.5

Algorithm 4.3: Greedy Column Subset Selection([16])

Input: $A \in \mathbb{R}^{m\times n}$, $K < n$.
Output: $\mathcal{I}$.

```
Initialize I = {}.
M^1 ← A.
for t = 1, 2, ..., K do
    Pick i s.t. ||M^t_{:,i}|| is maximal among all columns.
    I ← I ∪ {i}.
    M^{t+1} ← M^t − (M^t_{:,i}) · (M^t_{:,i})^T · (M^t)
end for
return I.
```

Theorem 4.7. *(Adapted from [50, Thm. 6] and [28, Thm. 2.1].) suppose that* $W \in \mathbb{R}^{(r+1)\times(r+1)}$,

$$W = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix},$$

and $A \in \mathbb{R}^{r\times r}$ *has maximal volume among all* $r \times r$ *submatrices of* W*. Then*

$$\frac{v_2(W)}{v_2(A)} \leq (1+r)\sigma_{r+1}(W). \tag{4.10}$$

The error bound of Theorem 4.5 can be readily deduced if the generator is chosen as a locally maximal volume submatrix. The following result shows that submatrix with maximal volume can be found among all principle submatrices.

Theorem 4.8. *([17]) Suppose that* W *is an* $n \times n$ *SPSD matrix and* $\mathcal{I}$ *and* $\mathcal{J}$ *are two sets of integers in* $\{1, \ldots, n\}$ *and have the same cardinality. Then* $v_2(W_{\mathcal{I},\mathcal{J}})^2 \leq v_2(W_{\mathcal{I},\mathcal{I}})\, v_2(W_{\mathcal{J},\mathcal{J}})$.

It is shown that the maximal volume submatrix M of an SPSD matrix A can be chosen to be principal.

This can be exploited to greatly reduce the cost of searching for the maximal volume submatrix. However, as pointed out in [17] and implied in [16], searching for a maximal volume submatrix in a general matrix, as well as a SPSD matrix, is NP hard, and therefore it is impractical for inputs with

moderately large size. Instead, [17] proposed to search for a submatrix with large volume through algorithm that is equivalent to **Gaussian Elimination with Complete Pivoting** (Algorithm 4.4). However the CUR LRA generated by such submatrix only guarantees a Chebyshev error bound of $4^r\sigma_{r+1}(A)$.

Algorithm 4.4: SPSD Matrix: Gaussian Elimination with Complete Pivoting ([2] and [17])

Input: SPSD matrix $A \in \mathbb{R}^{n\times n}$, $K < n$.
Output: $\mathcal{I}$.

Initialize $R \leftarrow A$, and $\mathcal{I} = \{\}$.
for t = 1, 2, ..., K **do**
  $i_t \leftarrow \arg\max_{x\in[n]} |r_{x,x}|$.
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i_t\}$.
  $R \leftarrow R - R_{:,i_t} \cdot r_{i_t,i_t}^{-1} \cdot R_{i_t,:}$.
end for
return $\mathcal{I}$.

In the following theorem, we show that if we iteratively improve the volume of $A_{\mathcal{I},\mathcal{I}}$ by replacing one index in $\mathcal{I}$, we will eventually arrive at some index set $\mathcal{I}$ s.t. $A_{\mathcal{I},\mathcal{I}}$ is a maximal volume submatrix of $A_{\mathcal{S},\mathcal{S}}$ for any $\mathcal{S} \supset \mathcal{I}$ and $|\mathcal{S}| = |\mathcal{I}|+1$. It can be shown that such a submatrix will generate a CUR LRA with Chebyshev error bound $(r+1)\sigma_{r+1}(A)$, considerably improving the aforementioned exponential bound.

Theorem 4.9. *Let $A \in \mathbb{R}^{n\times n}$ be a SPSD matrix, and let $\mathcal{I}$ be an non-empty*

index set and $|\mathcal{I}| = r < n$*. Suppose for any index set* $\mathcal{J}$ *where* $|\mathcal{J}| = r$*, and* $\mathcal{J}$ *only differs from* $\mathcal{I}$ *at one element, we have* $v_2(A_{\mathcal{I},\mathcal{I}}) \geq v_2(A_{\mathcal{J},\mathcal{J}})$*, then for any index set* $\mathcal{S} \supset \mathcal{I}$ *and* $|\mathcal{S}| = r + 1$*,* $A_{\mathcal{I},\mathcal{I}}$ *is a maximal volume submatrix of* $A_{\mathcal{S},\mathcal{S}}$*.*

Proof. Let $\mathcal{S} \supset \mathcal{I}$ and $|\mathcal{S}| = r + 1$ be any index set of A. Since $A_{\mathcal{S},\mathcal{S}}$ is again SPSD, by Theorem 4.8, there exists $\mathcal{I}' \subset \mathcal{S}$ and $|\mathcal{I}'| = r$, such that $A_{\mathcal{I}',\mathcal{I}'}$ is a maximal volume submatrix of $A_{\mathcal{S},\mathcal{S}}$. Since $\mathcal{I}'$ and $\mathcal{I}$ differs at most at one element, $v_2(A_{\mathcal{I},\mathcal{I}}) \geq v_2(A_{\mathcal{I}',\mathcal{I}'})$, and the theorem is proved.

□

As shown in [50] and [28], the condition that the generator $A_{\mathcal{I},\mathcal{I}}$ being a maximal volume submatrix can be relaxed considerably: if for some $\xi > 0$, the maximum submatrix volume is less than $(1+\xi)$ times $v_2(A_{\mathcal{I},\mathcal{I}})$, the error bound will only deteriorate by a factor no more than $(1 + \xi)$. In the case of SPSD inputs, we extend this relaxation further to $A_{\mathcal{I},\mathcal{I}}$ having **close-to-maximal** volume among "nearby" principle submatrices, that is, if the volume of $A_{\mathcal{I},\mathcal{I}}$ can not be improved more than $(1 + \xi)$ times by replacing one index in $\mathcal{I}$.

Theorem 4.10. *Let* $A \in \mathbb{R}^{n\times n}$ *be a SPSD matrix,* $r < n$ *a positive integer, and* ξ *a positive number. Let* $\mathcal{I} \subset [n]$ *be an index set, and* $|\mathcal{I}| = r$*. Suppose*

for all $\mathcal{J} \subset [n]$*, where* $|\mathcal{J}| = r$ *and* $\mathcal{J}$ *differs from* $\mathcal{I}$ *at one element, inequality* $(1+\xi)\, v_2(A_{\mathcal{I},\mathcal{I}}) \geq v_2(A_{\mathcal{J},\mathcal{J}})$ *holds. Then,*

$$||A - A_{:,\mathcal{I}} A_{\mathcal{I},\mathcal{I}}^{-1} A_{\mathcal{I},:}||_C \leq (1+\xi)(r+1)\sigma_{r+1}(A). \tag{4.11}$$

Proof. The theorem essentially follows from [28] Theorem 2.2. However, for the sake of completeness, we include a simplified proof here.

Let $\mathcal{I}^C = [n] - \mathcal{I}$. Since

$$||A - A_{:,\mathcal{I}} A_{\mathcal{I},\mathcal{I}}^{-1} A_{\mathcal{I},:}||_C \tag{4.12}$$

$$= ||A_{\mathcal{I}^C,\mathcal{I}^C} - A_{\mathcal{I}^C,\mathcal{I}} A_{\mathcal{I},\mathcal{I}}^{-1} A_{\mathcal{I},\mathcal{I}^C}||_C, \tag{4.13}$$

and the Schur Complement $A_{\mathcal{I}^C,\mathcal{I}^C} - A_{\mathcal{I}^C,\mathcal{I}} A_{\mathcal{I},\mathcal{I}}^{-1} A_{\mathcal{I},\mathcal{I}^C}$ is again SPSD ([34, Lemma 2.1]), it is only necessary to check all diagonal elements, that is $|A_{j,j} - A_{j,\mathcal{I}} A_{\mathcal{I},\mathcal{I}}^{-1} A_{\mathcal{I},j}|$ for all $j \in \mathcal{I}^C$.

For any $j \in \mathcal{I}^C$, let $\mathcal{S} = \mathcal{I} \cup \{j\}$, and $\gamma = |A_{j,j} - A_{j,\mathcal{I}} A_{\mathcal{I},\mathcal{I}}^{-1} A_{\mathcal{I},j}|$. Excluding the cases where $A_{\mathcal{I},\mathcal{I}}$ or $A_{\mathcal{S},\mathcal{S}}$ is singular, we have

$$\gamma = v_2(A_{\mathcal{S},\mathcal{S}}) / \, v_2(A_{\mathcal{I},\mathcal{I}}). \tag{4.14}$$

Let $A_{\mathcal{J},\mathcal{J}}$ be a maximal volume submatrix of $A_{\mathcal{S},\mathcal{S}}$, for some $\mathcal{J} \subset \mathcal{S}$ and

$|\mathcal{J}| = r$, then we have

$$\gamma = v_2(A_{\mathcal{S},\mathcal{S}}) / \, v_2(A_{\mathcal{I},\mathcal{I}}) \tag{4.15}$$

$$= \frac{v_2(A_{\mathcal{S},\mathcal{S}})}{v_2(A_{\mathcal{J},\mathcal{J}})} \frac{v_2(A_{\mathcal{J},\mathcal{J}})}{v_2(A_{\mathcal{I},\mathcal{I}})} \tag{4.16}$$

$$\leq (r+1)\sigma_{r+1}(A) \frac{v_2(A_{\mathcal{J},\mathcal{J}})}{v_2(A_{\mathcal{I},\mathcal{I}})} \tag{4.17}$$

$$\leq (1+\xi)(r+1)\sigma_{r+1}(A). \tag{4.18}$$

Inequality (4.17) follows from bound (4.10), and inequality (4.18) follows directly from the assumption. □

If $v_2(A_{\mathcal{I},\mathcal{I}})$ is increased by a factor no less than $(1+\xi)$ each time we replace an index in $\mathcal{I}$, Algorithm 4.6 can avoid running into infinite loop due to machine precision. Furthermore, Theorem 4.10 guarantees that the accuracy is mostly preserved, that is, upon termination, the returned index set $\mathcal{I}$ satisfies inequality (4.11).

Let t denote the number of times one index in $\mathcal{I}$ is replaced. In the following, we show that t is bounded by $O(r \log r)$, if the initial set $\mathcal{I}_0$ is greedily chosen(Algorithm 4.3).

Theorem 4.11. *(Adapted from [16, Theorem 10]) Let $C \in \mathbb{R}^{m \times n}$ be a matrix and $r < n$ be a positive integer. Let $\mathcal{I}$ be the output of Algorithm 4.3 with*

input C and r, then

$$v_2(C_{:,\mathcal{I}}) \geq \frac{1}{r!} \max_{\mathcal{S}\subset[n]:|\mathcal{S}|=r} v_2(C_{:,\mathcal{S}}). \tag{4.19}$$

Theorem 4.12. *Let $A \in \mathbb{R}^{n\times n}$ be a SPSD matrix, and $r < n$ a positive integer. Let $\mathcal{I}$ be the output of Algorithm 4.4 with inputs A and k, then,*

$$v_2(A_{\mathcal{I},\mathcal{I}}) \geq \frac{1}{(r!)^2} \max_{\mathcal{S}\subset[n]:|\mathcal{S}|=r} v_2(A_{\mathcal{S},\mathcal{S}}). \tag{4.20}$$

Proof. Since A is SPSD, there exists $C \in \mathbb{R}^{n\times n}$ s.t. $A = C^TC$. Therefore, for any non-empty index set $\mathcal{J} \subset [n]$,

$$A_{\mathcal{J},\mathcal{J}} = C^T_{:,\mathcal{J}} C_{:,\mathcal{J}}, \tag{4.21}$$

and thus

$$v_2(A_{\mathcal{J},\mathcal{J}}) = \big(\, v_2(C_{:,\mathcal{J}})\big)^2. \tag{4.22}$$

The submatrices correspond to the outputs of Algorithm 4.3 and Algorithm 4.4 with inputs (C, r) and (A, r) respectively would have the same volume. Therefore,

$$\begin{aligned} v_2(A_{\mathcal{I},\mathcal{I}}) &= \big(\, v_2(C_{:,\mathcal{I}})\big)^2 && (4.23)\\ &\geq \frac{1}{(r!)^2} \max_{\mathcal{S}\subset[n]:|\mathcal{S}|=r} v_2(C_{:,\mathcal{S}})^2 && (4.24)\\ &= \frac{1}{(r!)^2} \max_{\mathcal{S}\subset[n]:|\mathcal{S}|=r} v_2(A_{\mathcal{S},\mathcal{S}}) && (4.25) \end{aligned}$$

□

Corollary 4.12.1. *Let $A \in \mathbb{R}^{n\times n}$ be a SPSD matrix, $r = K < n$ positive integers, and ξ a positive number. Algorithm 4.6 will call Algorithm 4.5 at most $O(r \log r)$ times.*

Algorithm 4.5: Index Swap

Input: SPSD matrix $A \in \mathbb{R}^{n\times n}$, $\mathcal{I}$, $r \leq |\mathcal{I}|$ and $\xi > 0$.
Output: $\mathcal{J}$

```
Compute v_{2,r}(A_{I,I})
for all i ∈ I do
  I' ← I − {i}
  for all j ∈ [n] − I do
    J ← I' ∪ {j}
    Compute v_{2,r}(A_{J,J})
    if v_{2,r}(A_{J,J}) / v_{2,r}(A_{I,I}) > 1 + ξ then
      return J
    end if
  end for
end for
return I
```

4.4.3 Proof of Theorem 4.6

Lemma 4.3. *([50] Lemma 2 Item 2) Let $A \in \mathbb{R}^{n\times K}$ be a matrix, and assume that $n \geq K > r > 0$. Let $W = \begin{bmatrix} A & b \end{bmatrix} \in \mathbb{R}^{n\times(K+1)}$, and assume that A has maximal r-projective volume among all submatrices of size $n \times K$ in W. Then,*

$$v_{2,\mathrm{r}}(W) \leq v_{2,\mathrm{r}}(A)\sqrt{\tfrac{K+1}{K-r+1}}. \tag{4.26}$$

Corollary 4.12.2. *Let* $W = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)}$ *be a matrix, and suppose that* $A \in \mathbb{R}^{K\times K}$ *is the maximal* r*-projective volume submatrix in* W*. Then,*

$$v_{2,\mathrm{r}}(W) \leq v_{2,\mathrm{r}}(A)\tfrac{K+1}{K-r+1}. \tag{4.27}$$

Theorem 4.13. *(Adapted from [50] Thm. 7) Let* $A \in \mathbb{R}^{K\times K}$ *and* $W = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)}$ *be two matrices. Let* $\gamma = d - c^T(A)_r^+ b$ *for some* $0 < r \leq K$*, then*

$$|\gamma| \leq \frac{v_{2,\mathrm{r}}\left(W\right)}{v_{2,\mathrm{r}}\left(A\right)}\sigma_{r+1}(W). \tag{4.28}$$

Lemma 4.4. *Let* $A \in \mathbb{R}^{n\times n}$ *be a SPSD matrix. Let* $\mathcal{I}$ *and* $\mathcal{J}$ *be two non-empty index sets s.t.* $|\mathcal{I}| = |\mathcal{J}| = K < n$*, and let* $r \leq K$ *be a positive integer. Then,*

$$v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{J}}\right) \leq \max\Big(v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{I}}\right), v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)\Big). \tag{4.29}$$

Proof. Let $A = C^TC$ for some $C \in \mathbb{R}^{n\times n}$. Applying Theorem B.2 claim (ii), we have

$$v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{J}}\right) = v_{2,\mathrm{r}}\left(C_{:,\mathcal{I}}^T C_{:,\mathcal{J}}\right) \tag{4.30}$$

$$\leq v_{2,\mathrm{r}}\left(C_{:,\mathcal{I}}\right) v_{2,\mathrm{r}}\left(C_{:,\mathcal{J}}\right) \tag{4.31}$$

$$= \sqrt{v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{I}}\right) v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)} \tag{4.32}$$

□

Lemma 4.4 shows that, similar to the case of matrix volume, the submatrix of a SPSD matrix with maximal projective volume can be chosen to be principle, and in Theorem 4.14 we show that a principle submatrix with **close-to-maximal** projective volume among "nearby" principle submatrices generates a nicely bounded LRA.

Lemma 4.5. *Let* $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbb{R}^{n\times n}$ *be a SPSD matrix, where* $A_{11} \in \mathbb{R}^{K\times K}$. *Let* $C = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}$, *for* $0 < r \leq K$,

$$A - C(A_{11})_r^+ C^T \tag{4.33}$$

is a SPSD matrix.

Proof. Without loss of generality assume that A_{11} is non-singular and let $(A_{11})_\perp = A_{11} - (A_{11})_r$. Notice that

$$A - C(A_{11})_r^+ C^T = (A - CA_{11}^{-1}C^T) + C(A_{11})_\perp^+ C^T, \tag{4.34}$$

and the two terms on the right hand side are SPSD. □

Theorem 4.14. *Let* $A \in \mathbb{R}^{n\times n}$ *be a SPSD matrix, and* ξ *be a positive number. Let* $\mathcal{I} \subset [n]$ *be an index set, and* $|\mathcal{I}| = K$, *and assume that* $r < K < n$. *Suppose for all* $\mathcal{J} \subset [n]$, *where* $|\mathcal{J}| = K$ *and* $\mathcal{J}$ *differs from* $\mathcal{I}$ *at one element, inequality* $(1+\xi)\, v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{I}}\right) \geq v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)$ *holds. Then,*

$$||A - A_{:,\mathcal{I}}(A_{\mathcal{I},\mathcal{I}})_r^+ A_{\mathcal{I},:}||_C \leq (1+\xi)\frac{K+1}{K-r+1}\sigma_{r+1}(A). \tag{4.35}$$

Proof. The theorem essentially follows from ([50, Theorem 7]), however we include a simplified proof for completeness.

Let $R = A - A_{:,\mathcal{I}}(A_{\mathcal{I},\mathcal{I}})_r^+ A_{\mathcal{I},:}$, and by Lemma 4.5 R is SPSD. Therefore the entry with maximum absolute value will be on the diagonal, i.e. $||R||_C = \max_j |R_{j,j}|$, and our proof will be in two parts: (1) $j \in \mathcal{I}$, and (2) $j \in [n] - \mathcal{I}$.

The first part is straightforward,

$$\max_{j\in\mathcal{I}} |R_{j,j}| = ||A_{\mathcal{I},\mathcal{I}} - A_{\mathcal{I},\mathcal{I}}\big(A_{\mathcal{I},\mathcal{I}}\big)_r^+ A_{\mathcal{I},\mathcal{I}}||_C \tag{4.36}$$

and the right hand side is clearly less than $\sigma_{r+1}(A_{\mathcal{I},\mathcal{I}}) \leq \sigma_{r+1}(A)$. Now we move to the second part.

For any $j \in [n] - \mathcal{I}$, let $W = A_{\mathcal{S},\mathcal{S}} = \begin{bmatrix} A_{\mathcal{I},\mathcal{I}} & b \\ b^T & A_{j,j} \end{bmatrix}$ be the submatrix of A where $\mathcal{S} = \mathcal{I} \cup \{j\}$. Then,

$$R_{j,j} = A_{j,j} - b^T (A_{\mathcal{I},\mathcal{I}})_r^+ b. \tag{4.37}$$

Let $\mathcal{J} = \underset{\mathcal{X}\subset\mathcal{S}:|\mathcal{X}|=K}{\arg\max}\ v_{2,\mathrm{r}}\left(A_{\mathcal{X},\mathcal{X}}\right)$, and by Lemma 4.4 $A_{\mathcal{J},\mathcal{J}}$ has the maximal r-projective volume among all submatrices of size K in W. Combine this

with Corollary 4.12.2 and Theorem 4.13, we have

$$|R_{j,j}| \leq \frac{v_{2,\mathrm{r}}\left(W\right)}{v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{I}}\right)}\sigma_{r+1}(W) \tag{4.38}$$

$$\leq \frac{v_{2,\mathrm{r}}\left(W\right)}{v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)}\frac{v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)}{v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{I}}\right)}\sigma_{r+1}(A) \tag{4.39}$$

$$\leq (1+\xi)\frac{v_{2,\mathrm{r}}\left(W\right)}{v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)}\sigma_{r+1}(A) \tag{4.40}$$

$$\leq (1+\xi)\frac{K+1}{K-r+1}\sigma_{r+1}(A) \tag{4.41}$$

□

Next, we show that the upper bound of the number of iterations is sub-linear in n. In order to achieve this, we recall the well-known bound for the volume of the greedily selected column submatrix from [32], and the rest of the arguments follows naturally.

Lemma 4.6. *[32, Theorem 7.2], [16, Theorem 10] Let $C \in \mathbb{R}^{m\times n}$ be a matrix and $r \leq K < n$ be positive integers. Let $\mathcal{I}$ be the output of Algorithm 4.3 with input C and K, then*

$$v_{2,\mathrm{r}}\left(C_{:,\mathcal{I}}\right) \geq 2^{-r(r-1)/2}n^{-r/2}\, v_{2,\mathrm{r}}\left(C\right) \tag{4.42}$$

Following from arguments similar to that of Theorem 4.12, we have the volume of the initial submatrix obtained by Algorithm 4.4, $v_{2,\mathrm{r}}\left(A_{\mathcal{I},\mathcal{I}}\right)$, is

no less than $2^{-r(r-1)}n^{-r}$ of the projective volume of A, and the number of iterations is bounded by $O(r^2 + r \log n)$.

Algorithm 4.6: Main Algorithm

Input: SPSD matrix $A \in \mathbb{R}^{n \times n}$, positive integers K and r where $r \leq K < n$, and positive number ξ.
Output: $\mathcal{I}$

```
I ← Alg.4.4(A, K)
while TRUE do
  J ← Alg.4.5(A, I, r, ξ)
  if J = I then
    BREAK
  else
    I ← J
  end if
end while
return I.
```

4.4.4 Complexity Analysis

In this subsection, we analyze the time complexity of the Main Algorithm 4.6 in the case of both $r = K$ and $r < K$. The cost of finding the initial set $\mathcal{I}_0$ through Alg. 4.4 is $O(nK^2)$. Let t denote the number of iterations, and let $c(r, K)$ denote the cost of Algorithm 4.5 with parameters r, K. We have the complexity $O(nK^2 + t \cdot c(r, K))$.

In the case of $r = K$, Corollary 4.12.2 indicates that $t = O(r \log r)$. Within Alg. 4.5, potentially nr comparisons of $v_2\left(A_{\mathcal{I},\mathcal{I}}\right)$ and $v_2\left(A_{\mathcal{J},\mathcal{J}}\right)$ are

needed. Notice that $\mathcal{I}$ and $\mathcal{J}$ differs at most at one index, therefore if we compute $v_2\left(A_{\mathcal{J},\mathcal{J}}\right)$ through small rank update of $A_{\mathcal{I},\mathcal{I}}$ instead of computing from scratch, we could save a factor of r. Therefore, we have $c(r,r) = O(r^3 n)$, and time complexity of the Main Algorithm is $O(nr^4 \log r)$.

In the case of $r < K$, Lemma 4.6 implies that $t = O(r^2 + r \log n)$, and if $v_{2,\mathrm{r}}\left(A_{\mathcal{J},\mathcal{J}}\right)$ is computed through SVD, then $c(r,K) = O(K^4 n)$, and the time complexity of the Main Algorithm is $O(r^2 K^4 n + rK^4 n \log n)$.

Chapter 5

CUR LRA via Double-Sided Least Square Regression

Algorithm 5.1: Sublinear-time algorithm for approximate solution of $\min_Z \|A - CZR\|_F$

Input: $A \in \mathbb{R}^{m \times n}, C \in \mathbb{R}^{m \times d_1}, R \in \mathbb{R}^{d_2 \times n}$, and $\epsilon < 1$ a positive number.

Output: $Z \in \mathbb{R}^{d_1 \times d_2}$.;

Compute probabilities $p_{i,j}$ implicitly for $i \in [m]$ and $j \in [n]$ using Eqn. (5.2).

$c \leftarrow 3200 d_1^2 d_2^2 \epsilon^{-2}$

Initialize $Y \in \mathbb{R}^c$ and $W \in \mathbb{R}^{c \times d_1 d_2}$.

for $t = 1, 2, \ldots, c$ **do**

pick index pair (i_t, j_t) with probability p_{i_t, j_t}.

Set $Y_t = \frac{1}{\sqrt{c p_{i_t, j_t}}} A_{i_t, j_t}$

Set $W_t = \frac{1}{\sqrt{c p_{i_t, j_t}}} C_{i_t} \otimes R_{j_t}^T$

end for

Solve the least squares problem $Z = \arg\min_X ||Y - WX||_F$

return Reshaped $Z \in \mathbb{R}^{d_1 \times d_2}$.

Portions of this chapter previously appeared in our work [44].

5.1 Randomized Algorithm for Double-Sided Least Squares Problem

Algorithm 5.1 takes as input matrices $A \in \mathbb{R}^{m\times n}, C \in \mathbb{R}^{m\times d_1}$, $R \in \mathbb{R}^{d_2\times n}$ assuming $d_1, d_2 \leq \min(m, n)$, and a positive constant ϵ. The algorithm returns a matrix $Z \in \mathbb{R}^{d_1\times d_2}$ such that $CZR \approx A$, which can be interpreted as a reconstruction of A using C and R. The optimal solution to

$$\min_{Z} ||A - CZR||_F \tag{5.1}$$

is $Z_{opt} = C^+AR^+$, which requires to access the entire matrix A. However, a near-optimal solution can be obtained by using very few elements in A sampled from a non-uniform probability distribution constructed with C and R.

Specifically, Algorithm 5.1 first computes the top left-singular vectors U of C, V of R^T, and probability distribution

$$p_{i,j} = \frac{||U_i||_F^2||V_j||_F^2}{d_1 d_2} \text{ for } i \in [m], j \in [n]. \tag{5.2}$$

Then, we sample independently $c = \Theta(d_1^2 d_2^2 \epsilon^{-2})$ index pairs $\big\{(i_t, j_t)|t \in [c]\big\}$ from the probability distribution (5.2). According to the sampled index pairs, construct vector $Y \in \mathbb{R}^c$ and matrix $W \in \mathbb{R}^{c\times d_1 d_2}$, such that for all $t \in [c]$

$$Y_t = \frac{1}{\sqrt{cp_{i_t,j_t}}} A_{i_t,j_t} \quad \text{and} \quad W_t = \frac{1}{\sqrt{cp_{i_t,j_t}}} C_{i_t} \otimes R_{j_t}^T. \tag{5.3}$$

Finally, Z is computed, reshaped into a $d_1 \times d_2$ matrix

$$Z = \arg\min_x ||Y - WX||_F. \tag{5.4}$$

With probability of success greater or equal to 0.7 (this probability can be improved by increasing the number of the sampled index pairs), Algorithm 5.1 computes a near-optimal solution Z with $\|A - CZR\|_F^2 \leq (1+\epsilon)\|A - CZ_{opt}R\|_F^2$ using no more than $\Theta(d_1^2 d_2^2 \epsilon^{-2})$ elements from A, and at essentially sublinear computational cost.

5.2 Guarantee for Algorithm 5.1

Before we present our analysis of Algorithm 5.1 in Theorem 5.2, we first introduce a powerful supporting theorem regarding the quality of approximation for column/row sampling.

Theorem 5.1. *(Adapted from Thm. 5, Alg. Exactly(c) in [19]) Let $B \in \mathbb{R}^{m\times n}$ be a matrix of rank less or equal to k, $A \in \mathbb{R}^{m\times p}$, $0 < \epsilon < 1$, and let $Z_{opt} = \arg\min_X ||A - BX||_F = B^+A$. Let U be the top k left singular vectors of B and define any probability distribution*

$$p_i \geq \frac{\beta||U_i||_F^2}{k} \text{ for all } i \in [m] \tag{5.5}$$

for some $0 < \beta \leq 1$. Let $c = 3200k^2\epsilon^{-2}\beta^{-1}$, $Y \in \mathbb{R}^{c\times p}$ and $W \in \mathbb{R}^{c\times n}$ be two random matrices with independent rows, such that for all $t \in [c], i \in [m]$, the

Y_t and W_t equal to $\frac{1}{\sqrt{cp_i}}A_i$ and $\frac{1}{\sqrt{cp_i}}B_i$ respectively with probability p_i. Then, with probability no less than 0.7, for $Z = \arg\min_X ||Y - WX||_F$, we have

$$\begin{aligned} ||A - BZ||_F \leq & (1+\epsilon)||A - BZ_{opt}||_F \\ = & (1+\epsilon)\min_X ||A - BX||_F. \end{aligned} \tag{5.6}$$

Expected(c) sampling scheme from [19] provides a better asymptotic bound $O(k \log k\epsilon^{-2})$ on the number of required samples to achieve inequality (5.6). However, the constant factor of the bound is less obvious. Now we present the theoretical guarantee for Algorithm 5.1.

Theorem 5.2. *Assuming $d_1, d_2 \leq \min(m, n)$, and let $A \in \mathbb{R}^{m\times n}, C \in \mathbb{R}^{m\times d_1}$, $R \in \mathbb{R}^{d_2\times n}$ be three matrices, and $\epsilon < 1$ be any positive constant. Let Z be the output of Algorithm 5.1 with the above inputs, then with probability no less than 0.7, we have*

$$||A - CZR||_F \leq (1+\epsilon)\min_X ||A - CXR||_F. \tag{5.7}$$

Proof. If C(or R) is not a full rank matrix, we can locate and discard the extra columns(or rows) by performing algorithms such as rank-reveling QR factorization, and this can be done without losing precision in reconstructing A. Therefore, without loss of generality, assume that C and R be full rank matrices, and admit SVD decomposition $C = U_C\Sigma_C V_C^T = U_C S_C$, and $R = U_R\Sigma_R V_R^T = S_R V_R^T$. For simplicity, we name U_C and V_R as U and V,

respectively. Therefore,

$$||A - U\hat{Z}V^T||_F = ||A - CZR||_F, \tag{5.8}$$

where $\hat{Z} = S_C Z S_R$.

The element on the i-th row and j-th column of UXV^T is

$$\left(UXV^T\right)_{i,j} = \sum_{a=1}^{d_1}\sum_{b=1}^{d_2} U_{i,a}X_{a,b}V_{j,b}, \tag{5.9}$$

and is equivalent to the inner product of $U_i \otimes V_j$ and $\vec{X}$. Here $\vec{M}$ denotes the vectorization of a matrix $M \in \mathbb{R}^{m\times n}$ such that $\vec{M} = [M_{1,1}, M_{1,2}, \ldots, M_{m,n}]^T$. Therefore

$$||A - UXV^T||_F = ||\vec{A} - \overrightarrow{UXV^T}|| \tag{5.10}$$

$$= ||\vec{A} - (U \otimes V)\vec{X}||. \tag{5.11}$$

Define $f : [m] \times [n] \to [mn]$ to be a bijection between the indices of a matrix and the indices of its vectorization, such that $f(i,j) = (i-1)n + j$ for all $i \in [m]$, and $j \in [n]$. Since both U and V are orthogonal matrices, $U \otimes V$ is also orthogonal, and that

$$||(U \otimes V)_{f(i,j)}||_F^2 = \sum_{a=1}^{d_1}\sum_{b=1}^{d_2} U_{i,a}^2 V_{j,b}^2 \tag{5.12}$$

Draw independently with replacement $c = 3200 d_1^2 d_2^2 \epsilon^{-2}$ random index pairs,

$\{(i_t, j_t)|t \in [c]\}$, from probability distribution

$$p_{i,j} = \text{Prob}\big\{i_t = i, j_t = j\big\} \tag{5.13}$$

$$= \frac{||(U \otimes V)_{f(i,j)}||_F^2}{d_1 d_2}. \tag{5.14}$$

Then construct sample vector $Y \in \mathbb{R}^c$ and matrix $W \in \mathbb{R}^{c \times d_1 d_2}$, such that for all $t \in [c]$,

$$Y_t = \frac{1}{\sqrt{cp_{i_t,j_t}}} A_{i_t,j_t} \tag{5.15}$$

and

$$W_t = \frac{1}{\sqrt{cp_{i_t,j_t}}} U_{i_t} \otimes V_{j_t}. \tag{5.16}$$

Solve the following regression problem

$$\bar{Z} = \arg\min_{\vec{X}} ||Y - W\vec{X}||, \tag{5.17}$$

By applying Theorem 5.1 with the A, B replaced with $\vec{A}$, $U \otimes V$ respectively, and setting $\beta = 1$, it can be easily shown that the $\bar{Z}$ computed above satisfies the following inequality with probability no less than 0.7,

$$||\vec{A} - (U \otimes V)\bar{Z}|| \leq (1 + \epsilon) \min_{\vec{X}} ||\vec{A} - (U \otimes V)\vec{X}||. \tag{5.18}$$

Finally, reshape $\bar{Z}$ to $\hat{Z} \in \mathbb{R}^{d_1 \times d_2}$, and let $Z = S_C^{-1} \hat{Z} S_R^{-1}$, then we have

$$||A - CZR||_F = ||A - U\hat{Z}V^T||_F \tag{5.19}$$

$$= ||\vec{A} - \overrightarrow{U\hat{Z}V^T}|| \tag{5.20}$$

$$= ||\vec{A} - (U \otimes V)\bar{Z}|| \tag{5.21}$$

$$\leq (1+\epsilon) \min_{\vec{X}} ||\vec{A} - (U \otimes V)\vec{X}|| \tag{5.22}$$

$$= (1+\epsilon) \min_{X} ||A - CXR||_F \tag{5.23}$$

□

5.3 Relative Error Bound on $||A - CZR||_F$

In this section, we provide a near-optimal error bound analysis on the CUR decomposition Algorithm 5.1 produces, assuming sufficiently many columns and rows are sampled according to appropriate probability distributions.

Then we show that Algorithm 5.1, under conditions specified in Corollary 5.3.1, decomposes low-coherence input matrices near-optimally without accessing all entries and recovers unobserved entries in the process.

Before presenting the theorems, we first introduce the notations we use throughout this subsection. Given matrix $A \in \mathbb{R}^{m \times n}$, and an integer k, $k \leq \min(m, n)$, and let U and V be the top k left and right singular vectors

of A. We let

$$s_i = ||U_i||_F^2 \text{ for all } i \in [m] \tag{5.24}$$

denote the rank k row leverage scores of A, and similarly let

$$t_j = ||V_j||_F^2 \text{ for all } j \in [n] \tag{5.25}$$

denote the rank k column leverage scores of A. It is obvious that $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} = k$. Therefore, $\{s_i\}$ and $\{t_j\}$ naturally form two probability distributions $p_i = s_i/k, i \in [m]$ and $q_j = t_j/k, j \in [n]$.

We adopt the definition in [69] and let $\mu_r(A)$ [1], $\mu_c(A)$ [2], and $\mu(A)$ [3] denote the rank k row coherence, the rank k column coherence, and the rank k coherence, respectively. Notice that $r/m \leq \max_i\{s_i\} \leq 1$, and similarly $r/n \leq \max_j\{t_j\} \leq 1$. Therefore, $1 \leq \mu(A) \leq \max(m, n)$. We call A a low coherence matrix if $\mu(A)$ is a small constant, and $\mu(A) \ll \min(m/r, n/r)$.

Theorem 5.3. *Given $A \in \mathbb{R}^{m\times n}$, let $k \leq \min(m, n)$ be an integer, $\epsilon \in (0, 1]$, and $c_0 = 3^2 \cdot 3200$ be constants. Assume that $d_1 \geq c_0 k^2 \epsilon^{-2}$ columns are sampled with replacement according to probability distribution constructed with the column leverage scores of A, and construct $C \in \mathbb{R}^{m\times d_1}$ such that C consists of the sampled columns. Further assume that $d_2 \geq c_0 d_1^2 \epsilon^{-2}$ rows*

[1] $\mu_r(A) = \max_i\{ms_i/k\}$
[2] $\mu_c(A) = \max_j\{nt_j/k\}$
[3] $\mu(A) = \max\big(\mu_r(A), \mu_c(A)\big)$

are sampled with replacement according to probability distribution constructed with the row leverage scores, and construct $R \in \mathbb{R}^{d_2 \times n}$*, such that* R *consists of the sampled rows. Let* Z *be the output of Algorithm 5.1 with inputs* A*,* C*,* R*, and* $\epsilon/8$*, then with positive probability,*

$$||A - CZR||_F \leq (1+\epsilon)||A - A_k||_F.$$

Proof.

$$||A - CZR||_F \leq (1+\frac{\epsilon}{8})||A - CC^+AR^+R||_F \tag{5.26}$$

$$\leq (1+\frac{\epsilon}{8})(1+\frac{\epsilon}{3})||A - CC^+A||_F \tag{5.27}$$

$$\leq (1+\frac{\epsilon}{8})(1+\frac{\epsilon}{3})^2||A - A_k||_F \tag{5.28}$$

$$\leq (1+\epsilon)||A - A_k||_F \tag{5.29}$$

The first inequality is true due to Theorem 5.2, and the second and third inequalities are true due to Theorem 4 and Theorem 3 from [19] with $\epsilon/3$. All three inequalities have probability of success no less than 0.7, therefore taking union bound of the failure probability, inequality (5.29) holds with probability no less than 0.1. Notice that we can reduce the failure probability to δ by increasing the number of sampled columns, rows, and elements by $O(\log 1/\delta)$ times. □

Remark 5.1. *We can also achieve relative error CUR decomposition ap-*

plying Algorithm 5.1 with C and R sampled using Expected(c) from [19] or the sampling scheme provided in [7]. The aforementioned two sampling schemes provide superior asymptotic bounds on the required number of sampled columns/rows

that is ($d_1 = O(k \log k\epsilon^{-2}), d_2 = O(d_1 \log d_1 \epsilon^{-2})$ and $d_1, d_2 = O(k\epsilon^{-1})$, respectively). However, the constant factors on their bounds are less obvious.

Corollary 5.3.1. *Given $A \in \mathbb{R}^{m\times n}$, let $k \leq \min(m, n)$ be an integer, and let $\epsilon \in (0, 1]$ and $c_0 = 3^2 \cdot 3200$ be constants. Assume that the rank k coherence of A, $\mu(A) = \beta$, and that $d_1 \geq c_0 k^2 \beta \epsilon^{-2}$ columns are sampled with replacement uniformly, and construct $C \in \mathbb{R}^{m\times d_1}$, such that C consists of the sampled columns. Further assume that $d_2 \geq c_0 d_1^2 \epsilon^{-2}$ rows are sampled with replacement according to probability distribution constructed with the row leverage scores, and construct $R \in \mathbb{R}^{d_2\times n}$, such that R consists of the sampled rows. Let Z be the output of Algorithm 5.1 with inputs A, C, R, and $\epsilon/8$, then with positive probability,*

$$||A - CZR||_F \leq (1 + \epsilon)||A - A_k||_F.$$

Remark 5.2. *If $\mu(A) = \beta$ is a small constant, then by setting the column sampling probability distribution to uniform we have loss of accuracy by at most $1/\beta$, i.e., $p_j = 1/n \geq ||V_j||_F^2/k\beta$ for all $j \in [n]$, and this can be com-*

pensated by sampling β times more columns. In the case where the columns and rows are sampled independently, and given $O(k^2\beta\epsilon^{-2})$ columns and rows are sampled uniformly with replacement, the error bound deteriorate slightly to $(2+\epsilon)||A-A_k||_F$.

5.4 Algorithm Complexity

In this section, we confirm that given $m, n >> d_1, d_2$, Algorithm 5.1 achieves sublinear complexity.

In the sampling stage, the sampling probability distribution $p_{i,j}$ should be computed implicitly, otherwise storing $p_{i,j}$ would already require mn space, exceeding the claimed sublinear complexity. Fortunately,

$$\text{Prob}\big\{i_t = i, j_t = j\big\} = \frac{||(U \otimes V)_{f(i,j)}||_F^2}{d_1 d_2} \tag{5.30}$$

$$= \frac{||U_i||_F^2}{d_1}\frac{||V_j||_F^2}{d_2} \tag{5.31}$$

$$= \text{Prob}\big\{i_t = i\big\}\text{Prob}\big\{j_t = j\big\}. \tag{5.32}$$

In other words, in the sampling stage, we can simply sample the row(column) index first, and then independently sample the other. Therefore, the dominating computational cost, $O(md_1^2 + nd_2^2)$, will be the cost for computing top singular vectors, which can be achieved through QR(or SVD) factorization of the input matrices C and R.

Let $c = O(d_1^2 d_2^2 \epsilon^{-2})$ denote the number of samples required. The computational cost for constructing the down sampled problem $\min_X ||Y - WX||$ is $O(cd_1 d_2)$, and this problem can be solved in closed form as $W^+ Y$, whose cost is dominated by the cost, $O(cd_1^2 d_2^2)$, of computing the pseudo-inverse of W. In conclusion, the complexity of Algorithm 5.1 is $O(md_1^2 + nd_2^2 + d_1^4 d_2^4 \epsilon^{-2})$.

5.5 Numerical Experiments

To demonstrate the empirical applicability of the proposed algorithm, we evaluate it on six large-scale real-world data matrices, some of which can contain over one billion values. Although Theorem 5.2 requires a large overhead on the number of samples, our numerical experiments suggest that it often suffices to choose a reasonably small number of entries to solve for Z. The number of samples we pick for the experiments is not much greater than the number of entries necessary to guarantee a unique solution, and the number is extremely small compared to the size of the full matrix, resulting in a negligible amount of additional computations.

We implement the proposed CUR algorithm together with two relevant state-of-the-art CUR algorithms [19, 69] for comparisons. All tests are programmed using scripting language Python with highly optimized numerical linear algebra libraries NumPy [67] and SciPy [38]. SciPy sparse matrix mod-

ules are used to handle those sparse input matrices. All the experiments are run on a PC with Intel I7 3.5GHz CPU, 16GB RAM, and Windows operating system.

5.5.1 CUR Matrix Approximation on Low-Coherence Matrices

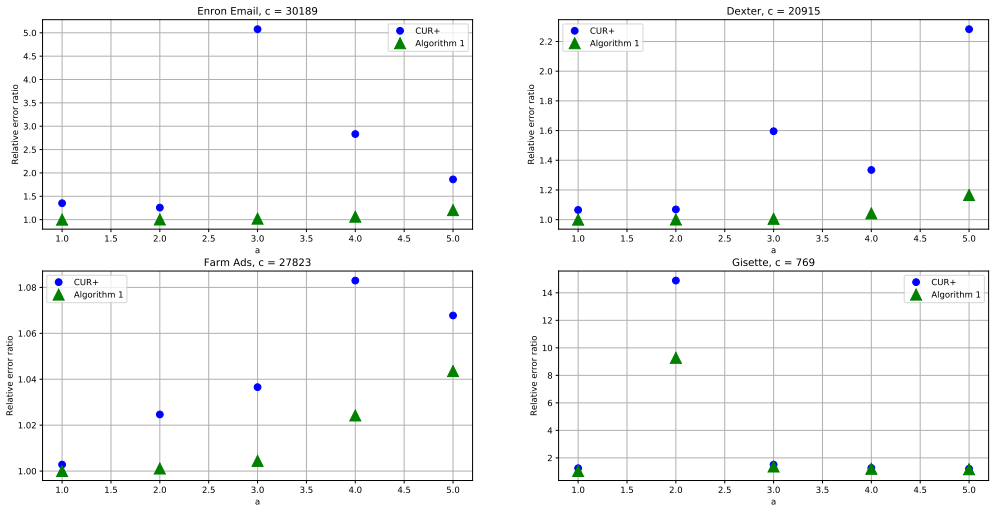

Figure 5.1: Relative error produced by Algorithm 5.1 and CUR+

In this subsection, we present the experimental results of the proposed algorithm on four benchmark data matrices for CUR matrix decomposition, which are widely used in previous work [68, 69].

The Enron Emails $(39,861 \times 28,102)$, Dexter $(20,000 \times 2,600)$, and Farm Ads $(54,877 \times 4,143)$ are textual data where in their matrix form, each row associates with one document, and each column associates with one word,

i.e., the element on the i-th row j-th column is the number of occurrence of word j in document i. Gisette $(13,500 \times 5,000)$ data consists of hand-written digits. In its matrix form, each row corresponds to one written digit, and each column corresponds to one feature.

All four data matrices have rapidly decaying singular values. Therefore, they are low numerical rank matrices suitable for CUR matrix approximation tasks. We refer readers to [68] for more details.

For each input data $A \in \mathbb{R}^{m \times n}$, we sample d_1 columns and d_2 rows uniformly. Let $C \in \mathbb{R}^{m \times d_1}$ be the matrix that consists of the sampled columns, and let $R \in \mathbb{R}^{d_2 \times n}$. Then we compute $Z \in \mathbb{R}^{d_1 \times d_2}$ as the return value of Algorithm 5.1 with inputs A, C, R, and c (i.e., number of samples). We compute the relative error of Z as

$$relative\ error = \frac{\|A - CZR\|_F}{\|A - CZ_{opt}R\|_F} \tag{5.33}$$

where,

$$Z_{opt} = argmin_X \|A - CXR\|_F \tag{5.34}$$

$$= C^+ A R^+, \tag{5.35}$$

for performance evaluation.

For comparison, we also include the relative error of Z_+, where

$$Z_+ = \arg\min_X \sum_{(i,j)\in S_+} (A_{i,j} - C_i X R^j)^2, \tag{5.36}$$

and S_+ is a subset of c matrix entries sampled uniformly without replacement. This is essentially the output of the CUR+ algorithm [69] applied with the same C, R, and c. The key difference between CUR+ algorithm and Algorithm 5.1 is that in Algorithm 5.1, Z is computed with equation

$$\min_{Z\in\mathbb{R}^{d_1\times d_2}} \sum_{(i,j)\in S} \frac{(A_{ij} - C_i Z R^j)^2}{p_{ij}|S|}, \tag{5.37}$$

where the summands are scaled, and S is a list of Ω matrix indices sampled independently with replacement according to a carefully constructed probability distribution.

In order to have comparable results, we follow the same experiment setting described in [69] by letting $d_1 = ar$, $d_2 = ad_1$, and $c = mnr^2/\text{nnz(A)}$, where nnz(A) represents the number of nonzero elements in A. We let $r = 10$, and $a = 1, 2, 3, 4, 5$. We run each test 10 times and report the mean relative error of Z and Z_+.

In this experiment, the relative errors produced by Algorithm 5.1 equal to approximately 1.0 consistently, indicating the output Z accurately approximates C^+AR^+ using only a small fraction of the entries in A. We notice that relative errors for both algorithms spike in tests on the Gisette Data with

$a = 2$. This is most likely caused by $c \approx d_1 d_2$, which may lead to a close to square down sampled regression problem that has a larger condition number. As a increases from 1 to 5, the relative error increases for both outputs. This is because the number of rows and columns sampled increases substantially, but the number of sampled elements stays fixed, making it harder to recover C^+AR^+ . However we observe that the relative error of Algorithm 5.1 behaves rather stable and only deteriorates slightly.

5.5.2 CUR with Leverage Score Sampling

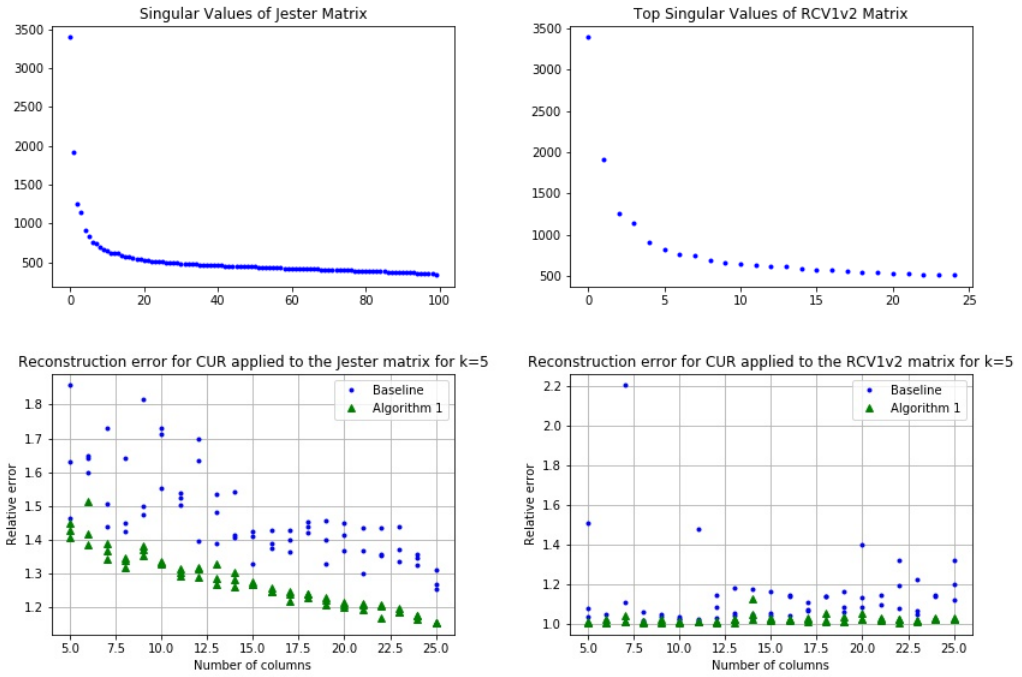

Figure 5.2: Singular Values of Jester and RCV1v2; Relative error produced by Algorithm 5.1 and Algorithm in [19].

In this section, we confirm that Algorithm 5.1 can improve the approximation level of the randomized CUR algorithm using leverage score sampling. We perform experiments on two large-scale matrices used in [19]:

- The so-called Jester joke dataset consists of anonymous ratings from the Jester Online Recommender System. It is developed by [24], and it is widely used as a benchmark for recommendation models.

- RCV1-v2: the LYRL2004 distribution of the RCV1-v2 text categorization test collection. This is a collection of "bag-of-words" vector representations of over 800,000 Reuters new stories from 1996 to 1997. We use the vector matrix from the training set provided by [42].

For both data matrices, we compute their rank 5 CUR approximation using the leverage score sampling for factor C and R, and Algorithm 5.1 for computing factor U. For comparison, we also compute the factor U as the pseudo-inverse of W, where W is the sub-block obtained by intersecting C and R. We define the *relative approximation error* as

$$\textit{relative error} = \frac{\|A - CUR\|_F}{\|A - A_5\|_F}. \tag{5.38}$$

Figure 5.2 displays the leading singular spectrum of both test matrices as well as the relative approximation errors for both Algorithm 5.1 and the

CUR algorithm in [19] with number of sampled columns c ranged from 5 to 25. The corresponding number of sampled rows is set to be $2c$, and the number of sampled entries is set to be 4 times the size of U. The test runs 3 times for each value of c.

The Jester matrix is a dense matrix of size $14,116 \times 100$ with entry values representing user ratings between ± 10.0. Its best rank-5 approximation A_5 is capable of capturing 81% of the matrix Frobenius norm. Using 5 columns and 10 rows, the algorithm developed in [19] produces CUR approximation with relative error about 1.5, and the error steadily decreases to about 1.3 as the number of columns and rows are increased to 25 and 50. In comparison, Algorithm 5.1 that uses the exactly same set of columns and rows constantly produces better CUR approximations, with relative error decreased to about 1.1 in the end.

The RCV1v2 matrix is a sparse $47,236 \times 23,149$ matrix with 0.16% of its entries being nonzero. The rank-5 relative approximation errors are quite close to 1 even for $c = r = 5$, and increasing c and r does not seem to further improve the approximation accuracy. Compared to the baseline CUR algorithm, Algorithm 5.1 has more stable performance and constantly produces lower relative error.

5.6 Summary

In this chapter, we propose a novel randomized sublinear-time algorithm that provides approximately optimal solution to the double-sided least squares problem with high probability. We present theoretical results that guarantee the solution of our method will be close to the optimal low-rank approximation with high probability of success. Numerical experiments are performed on various type of large datasets, demonstrating how existing superfast randomized CUR matrix algorithms can be improved by the proposed method to achieve better approximation quality while preserving low computational complexity.

Appendix A

Background on Matrix Computations

A.1 Basic Definitions

Recall in this section some basic definitions for matrix computations (cf. [1], [26]).

- W^T denote the transpose of an $m \times n$ matrix $W = (w_{ij})_{i,j=1}^{m,n}$. W is *orthogonal* if $W^T W = I_n$ or $WW^T = I_m$. W^* denote the Hermitian transpose of an $m \times n$ matrix $W = (w_{ij})_{i,j=1}^{m,n}$. W is *unitary* if $W^* W = I_n$ or $WW^* = I_m$.

- For a matrix $M = (m_{i,j})_{i,j=1}^{m,n}$ and two sets $\mathcal{I} \subseteq \{1, \ldots, m\}$ and $\mathcal{J} \subseteq$

Portions of this chapter previously appeared in our work [57].

$\{1, \ldots, n\}$, let $M_{\mathcal{I},:}$, $M_{:,\mathcal{J}}$, and $M_{\mathcal{I},\mathcal{J}}$ denote the submatrices $M_{\mathcal{I},:} := (m_{i,j})_{i\in\mathcal{I};j=1,\ldots,n}$, $M_{:,\mathcal{J}} := (m_{i,j})_{i=1,\ldots,m;j\in\mathcal{J}}$, and $M_{\mathcal{I},\mathcal{J}} := (m_{i,j})_{i\in\mathcal{I};j\in\mathcal{J}}$.

- Let I_k denote the $k \times k$ identity matrix. $O_{p,q}$ denotes the $p \times q$ matrix with all entries being zero. A $k \times k$ block diagonal matrix with diagonal blocks $B_1, \ldots, B_k$ is denoted by $\mathrm{diag}(B_1, \ldots, B_k) = \mathrm{diag}(B_j)_{j=1}^k$. A $1 \times k$ block matrix with blocks $B_1, \ldots, B_k$ is denoted by either $(B_1 \mid \ldots \mid B_k)$ or $(B_1, \ldots, B_k)$

- *Compact SVD* (or simply SVD) of a matrix W is defined by

$$W = U_W \Sigma_W V_W^T (\text{ or } \; W = S_W \Sigma_W T_W^* \;\text{ if }\; W \;\text{ is complex}). \tag{A.1}$$

  Let $\mathrm{rank}\, W = r \le \min(m, n)$, then

$$U_W^T U_W = V_W^T V_W = I_r (\text{ or } \; S_W^* S_W = T_W^* T_W = I_r), \tag{A.2}$$

  and $\Sigma_W := \mathrm{diag}(\sigma_j(W))_{j=1}^r$, where $\sigma_j(W)$ denotes the j-th largest singular value of W, and $\sigma_j(W) = 0$ for $j > r$.

- (see [26, Section 2.3.2 and Corollary 2.3.2]) Let $||\cdot|| = ||\cdot||_2$, $||\cdot||_F$, and $||\cdot||_C$ denote matrix Spectral, Frobenius, and Chebyshev norms

respectively. For $m \times n$ matrix $W = (W_{ij})_{i,j=1}^{m,n}$,

$$||W|| = \max_{||v||=1} ||Wv|| = \sigma_1(W), \tag{A.3}$$

$$||W||_F^2 := \sum_{i,j=1}^{m,n} |w_{ij}|^2 = \sum_{j=1}^{\operatorname{rank}(W)} \sigma_j^2(W), \tag{A.4}$$

$$||W||_C := \max_{i,j=1}^{m,n} |w_{ij}|, \tag{A.5}$$

$$||W||_C \le ||W|| \le ||W||_F \le \sqrt{mn}\ ||W||_C,\ ||W||_F^2 \le \min\{m,n\}\ ||W||^2. \tag{A.6}$$

- $W^+ = V_W \Sigma_W^{-1} U_W^T$ (or $W^+ := T_W \Sigma_W^{-1} S_W^*$) denotes the Moore–Penrose pseudo inverse of an $m \times n$ matrix W.

$$||W^+|| = 1/\sigma_r(W) \tag{A.7}$$

  for a matrix W of rank r.

- $\operatorname{rank}(M)$ denotes the *rank* of a matrix M. ϵ-$\operatorname{rank}(M)$ is defined as $\operatorname{argmin}_{|E| \le \epsilon |M|} \operatorname{rank}(M + E)$, and is called *numerical rank*, $\operatorname{nrank}(M)$, if ϵ is small in context.

- Let M_r denote the *rank-r truncation* of matrix M, obtained by setting $\sigma_j(M) = 0$ for $j > r$.

- Let M be a matrix of rank r, then $\kappa(M) = \frac{\sigma_1(M)}{\sigma_r(M)}$ is the spectral *condition number* of M, or equivalently $\kappa(M) = ||M||\ ||M^+||$.

A.2 Auxiliary Results

Lemma A.1. [The norm of the pseudo inverse of a matrix product.] *Suppose that* $A \in \mathbb{R}^{k\times r}$, $B \in \mathbb{R}^{r\times l}$ *and the matrices* A *and* B *have full rank* $r \leq \min\{k, l\}$. *Then* $|(AB)^+| \leq |A^+|\ |B^+|$.

Lemma A.2. (The norm of the pseudo inverse of a perturbed matrix, [6, Theorem 2.2.4].) *If* $\text{rank}(M + E) = \text{rank}(M) = r$, $\eta = ||M^+||\ ||E||$ *and* $\eta < 1$ *then*

$$\frac{1}{\sqrt{r}}||(M + E)^+|| \leq ||(M + E)^+|| \leq \frac{1}{1 - \eta}||M^+||.$$

Lemma A.3. (The impact of a perturbation of a matrix on its singular values, [26, Corollary 8.6.2].) *For* $m \geq n$ *and a pair of* $m \times n$ *matrices* M *and* $M + E$ *it holds that*

$$|\sigma_j(M + E) - \sigma_j(M)| \leq ||E|| \text{ for } j = 1, \ldots, n.$$

Theorem A.1. (The impact of a perturbation of a matrix on its top singular spaces, [26, Theorem 8.6.5].) *Let* $g =: \sigma_r(M) - \sigma_{r+1}(M) > 0$ and $||E||_F \leq 0.2g$. *Then for the left and right singular spaces associated with the* r *largest singular values of the matrices* M *and* $M + E$, *there exist orthogonal matrix*

bases $B_{r,\text{left}}(M)$, $B_{r,\text{right}}(M)$, $B_{r,\text{left}}(M+E)$, *and* $B_{r,\text{right}}(M+E)$ *such that*

$$\max\{||B_{r,\text{left}}(M+E) - B_{r,\text{left}}(M)||_F, ||B_{r,\text{right}}(M+E) - B_{r,\text{right}}(M)||_F\}$$
$$\leq 4\frac{||E||_F}{g}.$$

Remark A.1. *This theorem is especially useful in cases where* $||E||_F$ *is considerably less than* g*. For example, let* $||E|| \leq g/k$ *for* $k > 5$*, then the bound on the right hand side reduces to* $4/k$*, which approaches zero as* k *approaches infinity.*

A.3 Gaussian and Factor-Gaussian Matrices of Low Rank and Low Numerical Rank

Lemma A.4. [Orthogonal invariance of a Gaussian matrix.] *Let* k*,* m*, and* n *be three positive integers, where* $k \leq \min\{m, n\}$*, and let* G *be a* $m \times n$ *Gaussian matrix. For any orthogonal matrices* $U \in \mathbb{R}^{k\times m}$*,* $V \in \mathbb{R}^{n\times k}$*, the product matrices* UG *and* GV *have probability distribution of* $k \times n$ *and* $m \times k$ *Gaussian matrices respectively.*

Definition A.1. [Factor-Gaussian matrices.] *Let* $A \in \mathbb{R}^{m\times r}$*,* $B \in \mathbb{R}^{r\times n}$*, and* $C \in \mathbb{R}^{r\times r}$ *be well-conditioned matrices with rank* r*, assuming* $r \leq \min(m, n)$*. Let* $G_{m\times r}$ *and* $G_{r\times n}$ *be independent Gaussian matrices of size* $m \times r$ *and* $r \times n$ *respectively. Then we call* $AG_{r\times n}$*,* $G_{m\times r}B$*, and* $G_{m\times r}CG_{r\times n}$ **right**, **left**, *and*

two-sided *factor Gaussian matrix with rank* r*. Unless specific in context, we consider factor Gaussian matrix (with rank* r*) as random matrices distributed as* $G_{m\times r}G_{r\times n}$*. Notice that all above factor Gaussian matrices have rank* r *almost surely.*

Theorem A.2. *Any* $m \times n$ *two-sided factor Gaussian matrix with rank* r $G_{m\times r}RG_{r\times n}$ *defined as in Definition A.1 has the same distribution as* $G_{m\times r}\Sigma G_{r\times n}$ *for an appropriate diagonal matrix* Σ*, assuming that* $G_{m\times r}$ *and* $G_{r\times n}$ *are independent* $m \times r$ *and* $r \times n$ *Gaussian matrix.*

Proof. Let $R = U_R\Sigma_R V_R^T$ be SVD, and let $G_1 = G_{m\times r}U_R$ and $G_2 = V_R^T G_{r\times n}$. Recall that by orthogonal invariant property of Gaussian matrices, G_1 and G_2 are independent, and have the same distribution of $G_{m\times r}$ and $G_{r\times n}$ respectively. Therefore $G_{m\times r}RG_{r\times n} = G_1\Sigma_R G_2$, and the theorem follows. □

Definition A.2. *Define the* **relative norm of a perturbation of a Gaussian matrix** *as the ratio of the perturbation norm and the expected value of the norm of the matrix (estimated in Theorem A.3). Similarly define the* **relative norm of a perturbation of a factor Gaussian matrix**.

A.4 Norms of a Gaussian Matrix and Its Pseudo Inverse

$\Gamma(x) = \int_0^\infty \exp(-t)t^{x-1}dt$ denotes the Gamma function. $G_{p\times q}$ denotes a $p \times q$ Gaussian matrix for positive integers p and q.

Theorem A.3. [Norms of a Gaussian matrix. See [18, Theorem II.7] and our Definition 3.1.]

(i) $\text{Prob}\{||G_{m\times n}|| > t + \sqrt{m} + \sqrt{n}\} \le \exp(-t^2/2)$ *for* $t \ge 0$*, and furthermore* $\mathbb{E}(||G_{m\times n}||) \le \sqrt{m} + \sqrt{n}$.

(ii) $||G_{m\times n}||_F^2$ *has the same distribution as the* χ^2*-function with* mn *degrees of freedom, with expected value* $\mathbb{E}(||G_{m\times n}||_F^2) = mn$ *and probability density function* $\frac{2x^{mn-i}\exp(-x^2/2)}{2^{mn/2}\Gamma(mn/2)}$.

Theorem A.4. [Norms of the pseudo inverse of a Gaussian matrix (see [33, Proposition 10.4 and equations (10.3) and (10.4)] for claims (i) and (i), [61, Theorem 3.3] for claim (ii), and also Definition 3.1).]

(i) $\text{Prob}\{||G^+_{m\times n}||_F \ge t\sqrt{\frac{12n}{m-n}}\} \le 4t^{n-m}$ *and* $\text{Prob}\{||G^+_{m\times n}|| \ge t\frac{e\sqrt{m}}{m-n+1}\} \le t^{n-m-1}$ *for all* $t \ge 1$ *provided that* $m \ge n + 4$,

(ii) $\mathbb{E}(||G^+_{m\times n}||_F^2) = \frac{n}{m-n-1}$ *and* $\mathbb{E}(||G^+_{m\times n}||) \le \frac{e\sqrt{m}}{m-n}$ *if we assume that* $m \ge n + 2 \ge 4$,

(iii) $\text{Prob}\{||G^{+}_{n\times n}|| \geq x\} \leq \frac{\sqrt{n}}{x}$ *for* $n \geq 2$ *and all positive* x*, and furthermore*

$$\text{Prob}\left\{||(M_{n,n} + \lambda G_{n\times n})^{+}|| \geq x\right\} \leq \frac{2.35\sqrt{n}}{x\lambda} \tag{A.8}$$

for any $n \times n$ *matrix* $M_{n,n}$*, positive numbers* λ *.*

A.5 Supporting Lemma for Section 3.4

Lemma A.5. *Under the Assumption 3.2, fix positive number* $\xi < 1/4$*, then with probability no less than* $(1 - 2\sqrt{\xi})$*, we have*

$$\nu\mu \leq \xi^{-1}\theta. \tag{A.9}$$

Proof. By Theorem A.4 claim (ii), Theorem A.4 claim (iii), and Markov Inequality, we have

$$\text{Prob}\left\{\nu \geq \xi^{-0.5}(\sqrt{n} + \sqrt{r})\right\} \leq \sqrt{\xi} \tag{A.10}$$

and

$$\text{Prob}\left\{\mu \geq \xi^{-0.5}\frac{e\sqrt{l}}{l-r}\right\} \leq \sqrt{\xi}. \tag{A.11}$$

Therefore, with probability no less than $(1-2\sqrt{\xi})$, the following two inequalities $\nu \leq \xi^{-0.5}(\sqrt{n}+\sqrt{r})$ and $\mu \leq \xi^{-0.5}\frac{e\sqrt{l}}{l-r}$ hold. Notice that although μ and ν are dependent, our bound holds because union bound holds regardless of dependence. Deduce Lemma A.5. □

Appendix B

Results for Matrix Volume and r-Projective Volume

B.1 The Volume and r-Projective Volume of a Perturbed Matrix

Theorem B.1. *Suppose that W' and E are $k \times l$ matrices,* $\operatorname{rank}(W') = r \leq \min\{k, l\}$, *$W = W' + E$, and $||E|| \leq \epsilon$. Then*

$$\left(1 - \frac{\epsilon}{\sigma_r(W)}\right)^r \leq \prod_{j=1}^{r} \left(1 - \frac{\epsilon}{\sigma_j(W)}\right) \leq \frac{v_{2,r}(W)}{v_{2,r}(W')} \tag{B.1}$$

$$\leq \prod_{j=1}^{r} \left(1 + \frac{\epsilon}{\sigma_j(W)}\right) \leq \left(1 + \frac{\epsilon}{\sigma_r(W)}\right)^r. \tag{B.2}$$

If $\min\{k, l\} = r$, *then $v_2(W) = v_{2,r}(W)$, $v_2(W') = v_{2,r}(W')$, and*

$$\left(1 - \frac{\epsilon}{\sigma_r(W)}\right)^r \leq \frac{v_2(W)}{v_2(W')} = \frac{v_{2,r}(W)}{v_{2,r}(W')} \leq \left(1 + \frac{\epsilon}{\sigma_r(W)}\right)^r. \tag{B.3}$$

Portions of this chapter previously appeared in work [43].

Proof. Bounds (B.1) follow because a perturbation of a matrix within a norm bound ϵ changes its singular values by at most ϵ (see [26, Corollary 8.6.2]). Bounds (B.3) follow because $v_2(M) = v_{2,r}(M) = \prod_{j=1}^{r} \sigma_j(M)$ for any $k \times l$ matrix M with $\min\{k, l\} = r$, in particular for $M = W'$ and $M = W = W' + E$. □

If the ratio $\frac{\epsilon}{\sigma_r(W)}$ is small, then $\left(1 - \frac{\epsilon}{\sigma_r(W)}\right)^r = 1 - O\left(\frac{r\epsilon}{\sigma_r(W)}\right)$ and $\left(1 + \frac{\epsilon}{\sigma_r(W)}\right)^r = 1 + O\left(\frac{r\epsilon}{\sigma_r(W)}\right)$, which shows that the relative perturbation of the volume is amplified by at most a factor of r in comparison to the relative perturbation of the r largest singular values.

B.2 The Volume and r-Projective Volume of a Matrix Product

Theorem B.2. [See Examples B.1 and B.2 below.]

Suppose that $W = GH$ for an $m \times q$ matrix G and a $q \times n$ matrix H. Then

(i) $v_2(W) = v_2(G)v_2(H)$ if $q = \min\{m, n\}$; $v_2(W) = 0 \le v_2(G)v_2(H)$ if $q < \min\{m, n\}$.

(ii) $v_{2,r}(W) \le v_{2,r}(G)v_{2,r}(H)$ for $1 \le r \le q$,

(iii) $v_2(W) \le v_2(G)v_2(H)$ if $m = n \le q$.

The following examples show some limitations on the extension of the

theorem.

Example B.1. *If G and H are unitary matrices and if $GH = O$, then $v_2(G) = v_2(H) = v_{2,r}(G) = v_{2,r}(H) = 1$ and $v_2(GH) = v_{2,r}(GH) = 0$ for all $r \leq q$.*

Example B.2. *If $G = (1 \mid 0)$ and $H = \mathrm{diag}(1, 0)$, then $v_2(G) = v_2(GH) = 1$ and $v_2(H) = 0$.*

Proof. The theorem has been proved in [50]. Next we include an alternative proof.

We first prove claim (i).

Let $G = S_G \Sigma_G T_G^*$ and $H = S_H \Sigma_H T_H^*$ be SVDs such that Σ_G, T_G^*, S_H, Σ_H, and $U = T_G^* S_H$ are $q \times q$ matrices and S_G, T_G^*, S_H, T_H^*, and U are unitary matrices.

Write $V := \Sigma_G U \Sigma_H$. Notice that $\det(V) = \det(\Sigma_G)\det(U)\det(\Sigma_H)$. Furthermore $|\det(U)| = 1$ because U is a square unitary matrix. Hence $v_2(V) = |\det(V)| = |\det(\Sigma_G)\det(\Sigma_H)| = v_2(G)v_2(H)$.

Now let $V = S_V \Sigma_V T_V^*$ be SVD where S_V, Σ_V, and T_V^* are $q \times q$ matrices and where S_V and T_V^* are unitary matrices.

Observe that $W = S_G V T_H^* = S_G S_V \Sigma_V T_V^* T_H^* = S_W \Sigma_V T_W^*$ where $S_W = S_G S_V$ and $T_W^* = T_V^* T_H^*$ are unitary matrices. Consequently $W = S_W \Sigma_V T_W^*$

is SVD, and so $\Sigma_W = \Sigma_V$.

Therefore $v_2(W) = v_2(V) = v_2(G)v_2(H)$ unless $q < \min\{m, n\}$. This proves claim (i) because clearly $v_2(W) = 0$ if $q < \min\{m, n\}$.

Next prove claim (ii).

First assume that $q \leq \min\{m, n\}$ as in claim (i) and let $W = S_W \Sigma_W T_W^*$ be SVD.

In this case we have proven that $\Sigma_W = \Sigma_V$ for $V = \Sigma_G U \Sigma_H$, $q \times q$ diagonal matrices Σ_G and Σ_H, and a $q \times q$ unitary matrix U. Consequently $v_{2,r}(W) = v_{2,r}(\Sigma_V)$.

In order to prove claim (ii) in the case where $q \leq \min\{m, n\}$, it remains to deduce that

$$v_{2,r}(\Sigma_V) \leq v_{2,r}(G)v_{2,r}(H). \tag{B.4}$$

Notice that $\Sigma_V = S_V^* V T_V = S_V^* \Sigma_G U \Sigma_H T_V$ for $q \times q$ unitary matrices S_V^* and H_V.

Let $\Sigma_{r,V}$ denote the $r \times r$ leading submatrix of Σ_V, and so $\Sigma_{r,V} = \widehat{G}\widehat{H}$ where $\widehat{G} := S_{r,V}^* \Sigma_G U$ and $\widehat{H} := \Sigma_H T_{r,V}$ and where $S_{r,V}$ and $T_{r,V}$ denote the $r \times q$ leftmost unitary submatrices of the matrices S_V and T_V, respectively.

Observe that $\sigma_j(\widehat{G}) \leq \sigma_j(G)$ for all j because $\widehat{G}$ is a submatrix of the $q \times q$ matrix $S_V^* \Sigma_G U$, and similarly $\sigma_j(\widehat{H}) \leq \sigma_j(H)$ for all j. Therefore $v_{2,r}(\widehat{G}) = v_2(\widehat{G}) \leq v_{2,r}(G)$ and $v_{2,r}(\widehat{H}) = v_2(\widehat{H}) \leq v_{2,r}(H)$. Also notice that

$v_{2,r}(\Sigma_{r,V}) = v_2(\Sigma_{r,V})$.

Furthermore $v_2(\Sigma_{r,V}) \le v_2(\widehat{G})v_2(\widehat{H})$ by virtue of claim (i) because $\Sigma_{r,V} = \widehat{G}\widehat{H}$.

Combine the latter relationships and obtain (B.4), which implies claim (ii) in the case where $q \le \min\{m, n\}$.

Next we extend claim (ii) to the general case of any positive integer q.

Embed a matrix H into a $q \times q$ matrix $H' := (H \mid O)$ banded by zeros if $q > n$. Otherwise write $H' := H$. Likewise embed a matrix G into a $q \times q$ matrix $G' := (G^T \mid O)^T$ banded by zeros if $q > m$. Otherwise write $G' := G$.

Apply claim (ii) to the $m' \times q$ matrix G' and $q \times n'$ matrix H' where $q \le \min\{m', n'\}$.

Obtain that $v_{2,r}(G'H') \le v_{2,r}(G')v_{2,r}(H')$.

Substitute equations $v_{2,r}(G') = v_{2,r}(G)$, $v_{2,r}(H') = v_{2,r}(H)$, and $v_{2,r}(GH) = v_{2,r}(G'H')$, which hold because the embedding keeps invariant the singular values and therefore keeps invariant the volumes of the matrices G, H, and GH. This completes the proof of claim (ii), which implies claim (iii) because $v_2(V) = v_{2,n}(V)$ if V stands for G, H, or GH and if $m = n \le q$. □

Appendix C

Small Family of Hard Input

Any sublinear cost LRA algorithm fails on the following small families of LRA inputs.

Example C.1. *Let $\Delta_{i,j}$ denote a $m \times n$ matrix such that all entries are zero except that its (i,j)-th entry is 1. Then any matrix $W = \sum_{t=1}^{r} \Delta_{i_t,j_t}$ where $r \ll \min\{m,n\}$ is a low rank matrix, and any sublinear deterministic algorithm fails to compute LRA of W within reasonable error bound for such input.*

Such an input is hard for sublinear randomized algorithm as well. If the algorithm reads only $1/d$ of all entries from $\Delta_{i,j}$, then it fails to approximate $\Delta_{i,j}$ with probability at least $1 - 1/d$.

Portions of this chapter previously appeared in our work [57].

Bibliography

[1] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. *LAPACK Users' guide*, vol. 9. Siam, 1999.

[2] Bebendorf, M. Approximation of boundary element matrices. *Numerische Mathematik 86*, 4 (2000), 565–589.

[3] Bebendorf, M. Adaptive cross approximation of multivariate functions. *Constructive approximation 34*, 2 (2011), 149–179.

[4] Bebendorf, M., and Rjasanow, S. Adaptive low-rank approximation of collocation matrices. *Computing 70*, 1 (2003), 1–24.

[5] Ben-Israel, A. A volume associated with mxn matrices. *Linear algebra and its applications 167* (1992), 87–111.

[6] Björck, Å. *Numerical methods in matrix computations*, vol. 59. Springer, 2015.

[7] Boutsidis, C., and Woodruff, D. P. Optimal cur matrix decompositions. *SIAM Journal on Computing 46*, 2 (2017), 543–589.

[8] Businger, P., and Golub, G. H. Linear least squares solutions by householder transformations. *Numerische Mathematik 7*, 3 (1965), 269–276.

[9] Carroll, J. D., and Chang, J.-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika 35*, 3 (1970), 283–319.

[10] Chan, T. F. Rank revealing qr factorizations. *Linear algebra and its applications 88* (1987), 67–82.

[11] Chan, T. F., and Hansen, P. C. Computing truncated singular value decomposition least squares solutions by rank revealing qr-factorizations. *SIAM Journal on Scientific and Statistical Computing 11*, 3 (1990), 519–530.

[12] Chan, T. F., and Hansen, P. C. Some applications of the rank revealing qr factorization. *SIAM Journal on Scientific and Statistical Computing 13*, 3 (1992), 727–741.

[13] Chandrasekaran, S., and Ipsen, I. C. On rank-revealing factorisations. *SIAM Journal on Matrix Analysis and Applications 15*, 2 (1994), 592–622.

[14] Chen, Z., and Dongarra, J. J. Condition numbers of gaussian random matrices. *SIAM Journal on Matrix Analysis and Applications 27*, 3 (2005), 603–620.

[15] Cichocki, A., Lee, N., Oseledets, I., Phan, A.-H., Zhao, Q., Mandic, D. P., et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning 9*, 4-5 (2016), 249–429.

[16] Çivril, A., and Magdon-Ismail, M. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science 410*, 47-49 (2009), 4801–4811.

[17] Cortinovis, A., Kressner, D., and Massei, S. Mathicse technical report: On maximum volume submatrices and cross approximation for symmetric semidefinite and diagonally dominant matrices. Tech. rep., MATHICSE, 2019.

[18] Davidson, K. R., and Szarek, S. J. Local operator theory, random matrices and banach spaces. *Handbook of the geometry of Banach spaces 1*, 317-366 (2001), 131.

[19] Drineas, P., Mahoney, M. W., and Muthukrishnan, S. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications 30*, 2 (2008), 844–881.

[20] Edelman, A. Eigenvalues and condition numbers of random matrices. *SIAM Journal on Matrix Analysis and Applications 9*, 4 (1988), 543–560.

[21] Edelman, A., and Sutton, B. D. Tails of condition number distributions. *SIAM journal on matrix analysis and applications 27*, 2 (2005), 547–560.

[22] Fernando, K., and Nicholson, H. Discrete double-sided karhunen-loève expansion. In *IEE Proceedings D-Control Theory and Applications* (1980), vol. 127, IET, pp. 155–160.

[23] Gantmakher, F. R. *The theory of matrices*, vol. 131. American Mathematical Soc., 1959.

[24] Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval 4*, 2 (2001), 133–151.

[25] Golub, G. Numerical methods for solving linear least squares problems. *Numerische Mathematik 7*, 3 (1965), 206–216.

[26] Golub, G. H., and Loan, C. F. V. *Matrix computations*, 4 ed. Johns Hopkins University Press, 2013.

[27] Goreinov, S. A., Oseledets, I. V., Savostyanov, D. V., Tyrtyshnikov, E. E., and Zamarashkin, N. L. How to find a good submatrix. In *Matrix Methods: Theory, Algorithms And Applications: Dedicated to the Memory of Gene Golub*. World Scientific, 2010, pp. 247–256.

[28] Goreinov, S. A., and Tyrtyshnikov, E. E. The maximal-volume concept in approximation by low-rank matrices. *Contemporary Mathematics 280* (2001), 47–52.

[29] Goreinov, S. A., and Tyrtyshnikov, E. E. Quasioptimality of skeleton approximation of a matrix in the chebyshev norm. In *Doklady Mathematics* (2011), vol. 83, Springer, pp. 374–375.

[30] Goreinov, S. A., Tyrtyshnikov, E. E., and Zamarashkin, N. L. A theory of pseudoskeleton approximations. *Linear algebra and its applications 261*, 1-3 (1997), 1–21.

[31] Goreinov, S. A., Zamarashkin, N. L., and Tyrtyshnikov, E. E. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes 62*, 4 (1997), 515–519.

[32] Gu, M., and Eisenstat, S. C. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM Journal on Scientific Computing 17*, 4 (1996), 848–869.

[33] Halko, N., Martinsson, P.-G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review 53*, 2 (2011), 217–288.

[34] Harbrecht, H., Peters, M., and Schneider, R. On the low-rank approximation by the pivoted cholesky decomposition. *Applied numerical mathematics 62*, 4 (2012), 428–440.

[35] Harshman, R. Foundations of the parafac procedure: Model and conditions for an explanatory factor analysis. *Technical Report UCLA Working Papers in Phonetics 16, University of California, Los Angeles, Los Angeles, CA* (1970).

[36] Hong, Y. P., and Pan, C.-T. Rank-revealing qr-factorizations and the singular value decomposition. *Mathematics of Computation 58*, 197 (1992), 213–232.

[37] Hwang, T.-M., Lin, W.-W., and Yang, E. K. Rank revealing lu factorizations. *Linear algebra and its applications 175* (1992), 115–141.

[38] Jones, E., Oliphant, T., and Peterson, P. {SciPy}: Open source scientific tools for {Python}, 2014.

[39] Khoromskij, B., and Veit, A. Efficient computation of highly oscillatory integrals by using qtt tensor approximation. *Computational Methods in Applied Mathematics 16*, 1 (2016), 145–159.

[40] Kishore Kumar, N., and Schneider, J. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra 65*, 11 (2017), 2212–2244.

[41] Knuth, D. E. Semi-optimal bases for linear dependencies. *Linear and Multilinear Algebra 17*, 1 (1985), 1–4.

[42] Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research 5*, Apr (2004), 361–397.

[43] Luan, Q., and Pan, V. Y. Cur lra at sublinear cost based on volume maximization. In *International Conference on Mathematical Aspects of Computer and Information Sciences* (2019), Springer, pp. 105–121.

[44] Luan, Q., and Zhao, L. Efficient cur matrix decomposition via relative-error double-sided least squares solving. (in press) Accepted by 32th International Conference on Tools with Artificial Intelligence.

[45] Mahoney, M. W., and Drineas, P. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences 106*, 3 (2009), 697–702.

[46] Mahoney, M. W., et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning 3*, 2 (2011), 123–224.

[47] Musco, C., and Woodruff, D. P. Sublinear time low-rank approximation of positive semidefinite matrices. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017), IEEE, pp. 672–683.

[48] Oseledets, I., and Tyrtyshnikov, E. Tt-cross approximation for multidimensional arrays. *Linear Algebra and its Applications 432*, 1 (2010), 70–88.

[49] Osinsky, A. Probabilistic estimation of the rank 1 cross approximation accuracy. *arXiv preprint arXiv:1706.10285* (2017).

[50] Osinsky, A., and Zamarashkin, N. L. Pseudo-skeleton approximations with better accuracy estimates. *Linear Algebra and its Applications 537* (2018), 221–249.

[51] Pan, C.-T. On the existence and computation of rank-revealing lu factorizations. *Linear Algebra and its Applications 316*, 1-3 (2000), 199–222.

[52] Pan, V. Y. Some recent and new techniques for superfast (sublinear cost) low rank approximation. *arXiv preprint arXiv:1812.11406* (2018).

[53] Pan, V. Y., and Luan, Q. Refinement of low rank approximation of a matrix at sub-linear cost, 2019.

[54] Pan, V. Y., Luan, Q., Svadlenka, J., and Zhao, L. Superfast low-rank approximation and least squares regression, 2016.

[55] Pan, V. Y., Luan, Q., Svadlenka, J., and Zhao, L. Superfast cur matrix algorithms, their pre-processing and extensions, 2017.

[56] Pan, V. Y., Luan, Q., Svadlenka, J., and Zhao, L. Cur low rank approximation of a matrix at sub-linear cost, 2019.

[57] Pan, V. Y., Luan, Q., Svadlenka, J., and Zhao, L. Sublinear cost low rank approximation via subspace sampling. In *International Conference on Mathematical Aspects of Computer and Information Sciences* (2019), Springer, pp. 89–104.

[58] Pan, V. Y., Qian, G., and Yan, X. Random multipliers numerically stabilize gaussian and block gaussian elimination: proofs and an extension to low-rank approximation. *Linear Algebra and Its Applications 481* (2015), 202–234.

[59] Pan, V. Y., and Zhao, L. New studies of randomized augmentation and additive preprocessing. *Linear Algebra and its Applications 512* (2017), 256–305.

[60] Pan, V. Y., and Zhao, L. Numerically safe gaussian elimination with no pivoting. *Linear Algebra and its Applications 527* (2017), 349–383.

[61] Sankar, A., Spielman, D. A., and Teng, S.-H. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM Journal on Matrix Analysis and Applications 28*, 2 (2006), 446–476.

[62] Song, Z., Woodruff, D. P., and Zhong, P. Low rank approximation with entrywise l 1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (2017), ACM, pp. 688–701.

[63] Tropp, J. A., Yurtsever, A., Udell, M., and Cevher, V. Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications 38*, 4 (2017), 1454–1485.

[64] Tyrtyshnikov, E. Mosaic-skeleton approximations. *Calcolo 33*, 1-2 (1996), 47–57.

[65] Tyrtyshnikov, E. Incomplete cross approximation in the mosaic-skeleton method. *Computing 64*, 4 (2000), 367–380.

[66] Tyrtyshnikov, E., Goreinov, S., and Zamarashkin, N. Pseudo-skeleton approximations. *Doklay Akdemii Nauk 343*, 2 (1995), 151–152.

[67] Van Der Walt, S., Colbert, S. C., and Varoquaux, G. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering 13*, 2 (2011), 22.

[68] Wang, S., and Zhang, Z. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *The Journal of Machine Learning Research 14*, 1 (2013), 2729–2769.

[69] Xu, M., Jin, R., and Zhou, Z.-H. Cur algorithm for partially observed matrices. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37* (2015), JMLR. org, pp. 1412–1421.

[70] Zamarashkin, N., and Osinsky, A. New accuracy estimates for pseudoskeleton approximations of matrices. In *Doklady Mathematics* (2016), vol. 94, Springer, pp. 643–645.