Subspace-Orbit Randomized-Based Decomposition for Low-Rank Matrix Approximations

Maboud F. Kaloorazi
CETUC, Pontifical Catholic University
of Rio de Janeiro (PUC-Rio), Brazil
Email: kaloorazi@cetuc.puc-rio.br

Rodrigo C. de Lamare
CETUC, PUC-Rio, Brazil
Department of Electronics, University of York, UK
Email: delamare@cetuc.puc-rio.br

***Abstract*—In this paper we introduce a novel matrix decomposition algorithm termed Subspace-Orbit Randomized Singular Value Decomposition (SOR-SVD). It is computed by using random sampling techniques to give a low-rank approximation to an input matrix. Given a large and dense data matrix of size $m \times n$, SOR-SVD requires a few passes through data to compute a rank-k approximation in $O(mnk)$ floating-point operations. Furthermore, SOR-SVD can utilize advanced computer architectures and, as a result, it can be optimized for maximum efficiency. The SOR-SVD algorithm is simple, accurate, and provably correct, and outperforms previously reported techniques in terms of accuracy and efficiency.**

***Index Terms*—Matrix decomposition, low-rank approximation, randomized algorithms, numerical linear algebra.**

I. Introduction

Computing a low-rank approximation to a given data matrix is a fundamental task in numerical linear algebra and its applications. Such a low-dimensional representation of a matrix can provide a significant reduction in computational costs as well as memory requirements. Matrices with low-rank structures arise frequently in numerous applications such as latent variable graphical modeling, [1], ranking and collaborative filtering, [2], background subtraction [3], [4], system identification [5], IP network anomaly detection [6], [7], biometrics [8], sensor and multichannel signal processing [9], and statistical process control and multidimensional fault identification [10].

Traditional algorithms for constructing a low-rank approximation to a matrix, such as singular value decomposition (SVD) [11] and the rank-revealing QR factorization [12] can become computationally prohibitive for large data sets. Recently developed low-rank approximation algorithms based on random sampling techniques have been shown to be surprisingly efficient, accurate and robust, and are known to outperform their traditional counterparts in many practical situations [13]–[18]. The power of randomized methods lies in the fact that they can be optimized for maximum efficiency on modern architectures.

The methods in [13], [14], [19], [20] first sample columns of a data matrix with a probability proportional to either their magnitudes or leverage scores, representing the matrix in a compressed form. The submatrix is then used for further computations using algorithms such as the SVD and pivoted QR decomposition to obtain the final low-rank approximation. Sarlós [15] proposed a different method based on the well-known Johnson-Lindenstrauss (JL) lemma [21]. He showed that random linear combinations of rows can render a good approximation to a low-rank matrix. The works in [22], [23] also construct a low-rank approximation based on subspace embedding. Rokhlin et al. [16] propose to apply a random Gaussian embedding matrix in order to reduce the dimension of the data matrix, and a low-rank approximation is then given through computations of the reduced-sized matrix. Halko et al. [17] first form a low-dimensional subspace through a random linear combinations of matrix columns. Further computations on the reduced-sized matrix gives a low-rank approximation. Gu [18] uses a slightly modified version of the randomized algorithms of [17] to improve subspace iteration methods, and presents a new error analysis. The work in [4] proposes a rank-revealing decomposition based on randomized sampling, and applies it for solving the robust PCA problem.

In this paper, we propose a randomized decompositional approach called subspace-orbit randomized singular value decomposition (SOR-SVD) to compute a rank-k approximation to an input matrix. SOR-SVD requires a few passes over the data for a large and dense matrix of size $m \times n$, and is computed in $O(mnk)$ floating-point operations (flops). The main operations of SOR-SVD involve matrix-matrix multiplication and the QR decomposition, and due to recently developed Communication-Avoiding QR (CAQR) algorithms [24], which are optimal in terms of communication costs, it can be optimized for peak machine performance on modern computational platforms. We provide, without proof due to lack of space, theoretical lower bounds on the singular values and upper bounds on the error of the low-rank approximation for the SOR-SVD algorithm. We also experimentally verify that the Frobenius norm error bound provided is empirically sharp for one class of low-rank matrices.

We structure the remainder of this paper as follows. In Section II, we discuss prior works and the problem we are interested in solving. In Section III, we describe our proposed approach, which also includes a variant that uses the power iteration in detail. In Section IV, we present and discuss our experimental results, and our conclusion is given in Section V.

II. Problem Statement and Prior Works

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, where $m \geq n$, with numerical rank k, its SVD is defined as follows:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \begin{bmatrix}\mathbf{U}_k & \mathbf{U}_0\end{bmatrix} \begin{bmatrix}\mathbf{\Sigma}_k & 0 \\ 0 & \mathbf{\Sigma}_0\end{bmatrix} \begin{bmatrix}\mathbf{V}_k & \mathbf{V}_0\end{bmatrix}^T, \quad (1)$$

where $\mathbf{U}_k \in \mathbb{R}^{m \times k}$, $\mathbf{U}_0 \in \mathbb{R}^{m \times n-k}$ have orthonormal columns, $\mathbf{\Sigma}_k \in \mathbb{R}^{k \times k}$ and $\mathbf{\Sigma}_0 \in \mathbb{R}^{n-k \times n-k}$ are diagonal

ISBN 978-90-827970-1-5 © EURASIP 2018

containing the singular values, i.e., $\mathbf{\Sigma}_k = \text{diag}(\sigma_1, ..., \sigma_k)$ and $\mathbf{\Sigma}_0 = \text{diag}(\sigma_{k+1}, ..., \sigma_n)$, and $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ and $\mathbf{V}_0 \in \mathbb{R}^{n\times n-k}$ have orthonormal columns. The SVD constructs the optimal rank-k approximation $\hat{\mathbf{A}} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ to $\mathbf{A}$ [25], [26], i.e.,

$$\begin{aligned} \|\mathbf{A} - \hat{\mathbf{A}}\|_2 &= \sigma_{k+1}, \\ \|\mathbf{A} - \hat{\mathbf{A}}\|_F &= \sqrt{\sigma_{k+1}^2 + ... + \sigma_n^2}, \end{aligned} \tag{2}$$

where $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the spectral norm and the Frobenius norm, respectively. However, for large matrices, computing the SVD is expensive and, furthermore, standard techniques for its computation are challenging to parallelize [17], [27], [28]. As a result, alternative computationally efficient, accurate and robust algorithms which can easily lend themselves to a parallel implementation are desired. Throughout this paper we focus on the matrix $\mathbf{A}$ defined above.

Halko et al. [17] developed several randomized algorithms for low-rank matrix approximation. One of their basic algorithms for the matrix $\mathbf{A}$ and integers $k \leq \ell < n$ and q is computed as described in Alg. 1.

Algorithm 1 Randomized SVD (R-SVD)

Input: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, integers k, ℓ and q.
Output: A rank-ℓ approximation.
1: Draw a random matrix $\mathbf{\Omega} \in \mathbb{R}^{n\times l}$;
2: Compute $\mathbf{Y} = (\mathbf{A}\mathbf{A}^T)^q\mathbf{A}\mathbf{\Omega}$;
3: Compute a QR decomposition $\mathbf{Y} = \mathbf{Q}\mathbf{R}$;
4: Compute $\mathbf{B} = \mathbf{Q}^T\mathbf{A}$;
5: Compute an SVD $\mathbf{B} = \widetilde{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^T$;
6: $\mathbf{A} \approx (\mathbf{Q}\widetilde{\mathbf{U}})\mathbf{\Sigma}\mathbf{V}^T$.

Here q is the number of steps of a power method [16], [17]. Another approach proposed in [17, Section 5.5] is a *single-pass* algorithm, i.e., it requires only one pass through data, to compute a low-rank approximation. For the matrix $\mathbf{A}$, the decomposition, which we call two-sided randomized SVD (TSR-SVD), is computed as described in Alg. 2.

Algorithm 2 Two-Sided Randomized SVD (TSR-SVD)

Input: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, integers k and ℓ.
Output: A rank-ℓ approximation.
1: Draw random matrices $\mathbf{\Omega}_1 \in \mathbb{R}^{n\times\ell}$ and $\mathbf{\Omega}_2 \in \mathbb{R}^{m\times\ell}$;
2: Compute $\mathbf{Y}_1 = \mathbf{A}\mathbf{\Omega}_1$ and $\mathbf{Y}_2 = \mathbf{A}^T\mathbf{\Omega}_2$ in a single pass through $\mathbf{A}$;
3: Compute QR decompositions $\mathbf{Y}_1 = \mathbf{Q}_1\mathbf{R}_1$, $\mathbf{Y}_2 = \mathbf{Q}_2\mathbf{R}_2$;
4: Compute $\mathbf{B}_{\text{approx}} = \mathbf{Q}_1^T\mathbf{Y}_1(\mathbf{Q}_2^T\mathbf{\Omega}_1)^\dagger$;
5: Compute an SVD $\mathbf{B}_{\text{approx}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}$;
6: $\mathbf{A} \approx (\mathbf{Q}_1\widetilde{\mathbf{U}})\widetilde{\mathbf{\Sigma}}(\mathbf{Q}_2\widetilde{\mathbf{V}})^T$.

In Alg. 2, $\mathbf{B}_{\text{approx}}$ is an approximation to $\mathbf{B} = \mathbf{Q}_1^T\mathbf{A}\mathbf{Q}_2$, and the dagger $\dagger$ denotes the pseudo-inverse.

The TSR-SVD, however, substantially degrades the quality of the approximation, compared to the R-SVD, making it unsuitable for use in practice. The reason behind is, chiefly, poor approximate bases drawn from the row space of $\mathbf{A}$, i.e., $\mathbf{Q}_2$. Furthermore, the authors do not provide error bounds, neither upper bounds on the error of the low-rank approximation nor lower bounds on the singular values, for the TSR-SVD algorithm. This work addresses these issues.

This work develops a randomized algorithm for low-rank approximation that with comparable flops *i)* outperforms the TSR-SVD in terms of accuracy, and *ii)* can utilize advanced computer architectures better than the R-SVD.

III. Subspace-Orbit Randomized Singular Value Decomposition

In this section, we present a randomized algorithm termed subspace-orbit randomized SVD (SOR-SVD) [29] that computes a low-rank approximation of a given matrix using randomization. We also present a version of SOR-SVD with power method, which improves the performance of the algorithm at an extra cost.

Given the matrix $\mathbf{A}$, and an integer $k \leq \ell < n$, SOR-SVD is computed by taking the following seven steps:

1) Generate a standard Gaussian matrix $\mathbf{\Omega} \in \mathbb{R}^{n\times\ell}$,
2) Compute the matrix product:

$$\mathbf{T}_1 = \mathbf{A}\mathbf{\Omega}, \tag{3}$$

The matrix $\mathbf{T}_1 \in \mathbb{R}^{m\times\ell}$ is formed through linear combinations of columns of $\mathbf{A}$ by $\mathbf{\Omega}$. $\mathbf{T}_1$, in fact, is a projection onto the subspace spanned by columns of $\mathbf{A}$.
3) Compute the matrix product:

$$\mathbf{T}_2 = \mathbf{A}^T\mathbf{T}_1, \tag{4}$$

The matrix $\mathbf{T}_2 \in \mathbb{R}^{n\times\ell}$ is constructed by linear combinations of rows of $\mathbf{A}$ by $\mathbf{T}_1$. $\mathbf{T}_2$, in fact, is a projection onto the subspace spanned by rows of $\mathbf{A}$.
4) Compute QR decompositions of $\mathbf{T}_1$ and $\mathbf{T}_2$:

$$\mathbf{T}_1 = \mathbf{Q}_1\mathbf{R}_1 \quad \text{and} \quad \mathbf{T}_2 = \mathbf{Q}_2\mathbf{R}_2, \tag{5}$$

The matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are approximate bases for the range of $\mathbf{A}$ and the range of $\mathbf{A}^T$, respectively.
5) Compute the matrix product:

$$\mathbf{M} = \mathbf{Q}_1^T\mathbf{A}\mathbf{Q}_2, \tag{6}$$

$\mathbf{M} \in \mathbb{R}^{\ell\times\ell}$ is formed by compression of $\mathbf{A}$ via left and right multiplications by orthonormal bases.
6) Compute the rank-k truncated SVD of $\mathbf{M}$:

$$\mathbf{M}_k = \widetilde{\mathbf{U}}_k\widetilde{\mathbf{\Sigma}}_k\widetilde{\mathbf{V}}_k, \tag{7}$$

7) Form SOR-SVD-based low-rank approximation of $\mathbf{A}$:

$$\hat{\mathbf{A}}_{\text{SOR}} = (\mathbf{Q}_1\widetilde{\mathbf{U}}_k)\widetilde{\mathbf{\Sigma}}_k(\mathbf{Q}_2\widetilde{\mathbf{V}}_k)^T, \tag{8}$$

where $\mathbf{Q}_1\widetilde{\mathbf{U}}_k \in \mathbb{R}^{m\times k}$ and $\mathbf{Q}_2\widetilde{\mathbf{V}}_k \in \mathbb{R}^{n\times k}$ are approximations to the k leading left and right singular vectors of $\mathbf{A}$, respectively, and $\widetilde{\mathbf{\Sigma}}_k$ contains an approximation to the k leading singular values of $\mathbf{A}$.

The SOR-SVD, described in its basic form, requires three passes over data, for a matrix stored-out-of-core, but it can be modified to revisit the data only once. To this end, the matrix $\mathbf{M}$ (6) can be approximated as follows: both sides of the currently unknown equation $\mathbf{M} = \mathbf{Q}_1^T\mathbf{A}\mathbf{Q}_2$ are postmultiplied by $\mathbf{Q}_2^T\mathbf{\Omega}$, obtaining $\mathbf{M}\mathbf{Q}_2^T\mathbf{\Omega} = \mathbf{Q}_1^T\mathbf{A}\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{\Omega}$. Having defined $\mathbf{A} \approx \mathbf{A}\mathbf{Q}_2\mathbf{Q}_2$ and $\mathbf{T}_1 = \mathbf{A}\mathbf{\Omega}$, then $\mathbf{M}_{\text{approx}} = \mathbf{Q}_1^T\mathbf{T}_1(\mathbf{Q}_2^T\mathbf{\Omega})^\dagger$.

Unlike TSR-SVD, SOR-SVD projects $\mathbf{A}$ onto a subspace spanned by its rows using a sketch of $\mathbf{A}$. This significantly

ISBN 978-90-827970-1-5 © EURASIP 2018

improves the approximate basis for the range of $\mathbf{A}^T$ and, moreover, results in tighter theoretical bounds.

The SOR-SVD may be fairly accurate for matrices whose singular values display some decay, but for matrices with slowly decaying singular spectrum, it may produce a poor approximation compared to that of the SVD. Thus, we incorporate q steps of a power iteration [17] to improve the accuracy of the algorithm in these circumstances. Given $\mathbf{A}$, $k \leq \ell < n$ and q, the resulting algorithm is described in Alg. 3.

Algorithm 3 The SOR-SVD algorithm with Power Method

Input: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, integers k, ℓ and q.
Output: A rank-k approximation.

1: Draw a standard Gaussian matrix $\mathbf{T}_2 \in \mathbb{R}^{n\times \ell}$;
2: **for** $i = 1$: $q+1$ **do**
3: Compute $\mathbf{T}_1 = \mathbf{A}\mathbf{T}_2$;
4: Compute $\mathbf{T}_2 = \mathbf{A}^T\mathbf{T}_1$;
5: **end for**
6: Compute QR decompositions $\mathbf{T}_1 = \mathbf{Q}_1\mathbf{R}_1$, $\mathbf{T}_2 = \mathbf{Q}_2\mathbf{R}_2$;
7: Compute $\mathbf{M} = \mathbf{Q}_1^T\mathbf{A}\mathbf{Q}_2$ or $\mathbf{M}_{\text{approx}} = \mathbf{Q}_1^T\mathbf{T}_1(\mathbf{Q}_2^T\mathbf{T}_2)^\dagger$;
8: Compute the rank-k truncated SVD $\mathbf{M}_k = \widetilde{\mathbf{U}}_k\widetilde{\mathbf{\Sigma}}_k\widetilde{\mathbf{V}}_k$ or $\mathbf{M}_{\text{approx-k}} = \widetilde{\mathbf{U}}_k\widetilde{\mathbf{\Sigma}}_k\widetilde{\mathbf{V}}_k$;
9: Form the SOR-SVD-based low-rank approximation of $\mathbf{A}$: $\hat{\mathbf{A}}_{\text{SOR}} = (\mathbf{Q}_1\widetilde{\mathbf{U}}_k)\widetilde{\mathbf{\Sigma}}_k(\mathbf{Q}_2\widetilde{\mathbf{V}}_k)^T$.

Note that when the power method is employed a non-updated $\mathbf{T}_2$ must be used in order to form $\mathbf{M}_{\text{approx}}$.

The accuracy of singular values and the low-rank approximation of $\mathbf{A}$ computed by the SOR-SVD algorithm, as stated in the next subsection, depends strongly on the ratio $\frac{\sigma_{\ell-p+1}}{\sigma_j}$ for $j = 1, ..., k$, and on $\frac{\sigma_{\ell-p+1}}{\sigma_k}$, respectively, where $2 \leq p \leq \ell - k$ is an oversampling parameter. The power method decreases the extra factors in the error bounds by driving down the aforesaid ratios exponentially fast.

A. Deterministic Error Bounds

Our analysis of SOR-SVD was inspired by the work in [18]. Let the matrix $\mathbf{A}$ have an SVD defined in (1), $\mathbf{\Omega} \in \mathbb{R}^{n\times\ell}$ (or $\mathbf{T}_2$ in Alg. 3) be a standard Gaussian matrix, and $2 \leq p \leq \ell - k$, where ℓ is the number of samples and p is an oversampling parameter. Decompose $\mathbf{\Omega}$ such as

$$\widetilde{\mathbf{\Omega}} = \mathbf{V}^T\mathbf{\Omega} = [\widetilde{\mathbf{\Omega}}_1^T \quad \widetilde{\mathbf{\Omega}}_2^T]^T \tag{9}$$

where $\widetilde{\mathbf{\Omega}}_1$ and $\widetilde{\mathbf{\Omega}}_2$ have $\ell - p$ and $n - \ell + p$ rows, respectively. The error bounds for SOR-SVD depend on the properties of $\widetilde{\mathbf{\Omega}}_1$ and $\widetilde{\mathbf{\Omega}}_2$. The following theorem sets forth lower bounds on approximated singular values.

Theorem 1: Suppose that the matrix $\mathbf{A}$ has an SVD defined in (1), $2 \leq p + k \leq \ell$, and $\hat{\mathbf{A}}_{\text{SOR}}$ is computed by SOR-SVD. Assume that $\widetilde{\mathbf{\Omega}}_1$ is full row rank, then for $j = 1, ..., k$, we have

$$\sigma_j \geq \sigma_j(\hat{\mathbf{A}}_{\text{SOR}}) \geq \frac{\sigma_j}{\sqrt{1 + \|\widetilde{\mathbf{\Omega}}_2\|_2^2\|\widetilde{\mathbf{\Omega}}_1^\dagger\|_2^2\left(\frac{\sigma_{\ell-p+1}}{\sigma_j}\right)^{\Delta}}}, \tag{10}$$

where $\Delta = 4$ for the basic form of the algorithm, and $\Delta = 4q + 4$ when the power method is used, i.e., Alg. 3.

The following theorem establishes upper bounds on the error of the low-rank approximation.

Theorem 2: With the notation of Theorem 1, and $\varrho = 2, F$, the low-rank approximation error must satisfy

$$\begin{aligned}\|\mathbf{A} - \hat{\mathbf{A}}_{\text{SOR}}\|_\varrho \leq & \|\mathbf{\Sigma}_0\|_\varrho + \sqrt{\frac{\alpha^2\|\widetilde{\mathbf{\Omega}}_2\|_2^2\|\widetilde{\mathbf{\Omega}}_1^\dagger\|_2^2}{1 + \beta^2\|\widetilde{\mathbf{\Omega}}_2\|_2^2\|\widetilde{\mathbf{\Omega}}_1^\dagger\|_2^2}} \\ & + \sqrt{\frac{\eta^2\|\widetilde{\mathbf{\Omega}}_2\|_2^2\|\widetilde{\mathbf{\Omega}}_1^\dagger\|_2^2}{1 + \tau^2\|\widetilde{\mathbf{\Omega}}_2\|_2^2\|\widetilde{\mathbf{\Omega}}_1^\dagger\|_2^2}},\end{aligned} \tag{11}$$

where $\alpha = \sqrt{k}\frac{\sigma_{\ell-p+1}^2}{\sigma_k}$, $\beta = \frac{\sigma_{\ell-p+1}^2}{\sigma_1\sigma_k}$, $\eta = \sqrt{k}\sigma_{\ell-p+1}$ and $\tau = \frac{\sigma_{\ell-p+1}}{\sigma_1}$. And when the power method is used, Alg. 3, $\alpha = \sqrt{k}\frac{\sigma_{\ell-p+1}^2}{\sigma_k}\left(\frac{\sigma_{\ell-p+1}}{\sigma_k}\right)^{2q}$, $\beta = \frac{\sigma_{\ell-p+1}^2}{\sigma_1\sigma_k}\left(\frac{\sigma_{\ell-p+1}}{\sigma_k}\right)^{2q}$, $\eta = \dfrac{\sigma_k}{\sigma_{\ell-p+1}}\alpha$ and $\tau = \dfrac{1}{\sigma_{\ell-p+1}}\beta$.

B. Average-Case Error Bounds

Since $\mathbf{\Omega}$ is a random Gaussian matrix, we provide the average-case error bounds for the SOR-SVD algorithm.

Theorem 3: With the notation of Theorem 1, and $\gamma_j = \frac{\sigma_{\ell-p+1}}{\sigma_j}$ for $j = 1, ..., k$, we have

$$\mathbb{E}(\sigma_j(\hat{\mathbf{A}}_{\text{SOR}})) \geq \frac{\sigma_j}{\sqrt{1 + \nu^2\gamma_j^{\Delta}}}, \tag{12}$$

where $\mathbb{E}$ denotes expected value, $\nu = \nu_1\nu_2$, $\nu_1 = \sqrt{n - \ell + p} + \sqrt{\ell} + 7$, and $\nu_2 = \frac{4\mathrm{e}\sqrt{\ell}}{p+1}$. The exponent $\Delta = 4$ for the basic form of the algorithm, and $\Delta = 4q + 4$ when the power method is used, Alg. 3.

Theorem 4: With the notation of Theorem 1, for the basic form of SOR-SVD, we have

$$\mathbb{E}\|\mathbf{A} - \hat{\mathbf{A}}_{\text{SOR}}\|_\varrho \leq \|\mathbf{\Sigma}_0\|_\varrho + (1 + \gamma_k)\sqrt{k}\nu\sigma_{\ell-p+1}, \tag{13}$$

and when the power method is used, Alg. 3, we have

$$\mathbb{E}\|\mathbf{A} - \hat{\mathbf{A}}_{\text{SOR}}\|_\varrho \leq \|\mathbf{\Sigma}_0\|_\varrho + (1 + \gamma_k)\sqrt{k}\nu\sigma_{\ell-p+1}\gamma_k^{2q}, \tag{14}$$

where γ_k and ν are defined in Theorem 3.

C. Computational Complexity

The cost of an algorithm involves both arithmetic, i.e., the number of flops, and communication, i.e., data movement either between different levels of a memory hierarchy or between processors [24]. On multicore and accelerator-based computers, for a data matrix which is too large to fit in fast memory, the communication cost becomes substantially more expensive compared to the arithmetic [24], [30]. The power of randomized algorithms lies in the fact that they operate on a compressed version of the data matrix rather than a matrix itself and, moreover, they can be organized to exploit modern computational platforms better than their classical counterparts.

To decompose $\mathbf{A}$, the simple version of SOR-SVD incurs the following costs: Step 1 costs $O(n\ell)$, Step 2 costs $O(mn\ell)$, Step 3 costs $O(mn\ell)$, Step 4 costs $O(m\ell^2 + n\ell^2)$, Step 5 costs $O(mn\ell + m\ell^2)$ (if $\mathbf{M}$ is approximated by $\mathbf{M}_{\text{approx}}$, this step costs $O(m\ell^2 + n\ell^2 + \ell^3)$), Step 6 costs $O(\ell^3)$, Step 7 costs $O(m\ell k + n\ell k)$. The dominant cost of Step 1-7 occurs when multiplying $\mathbf{A}$ and $\mathbf{A}^T$ with the corresponding matrices. Thus

$$C_{\text{SOR-SVD}} = O(mn\ell). \tag{15}$$

ISBN 978-90-827970-1-5 © EURASIP 2018

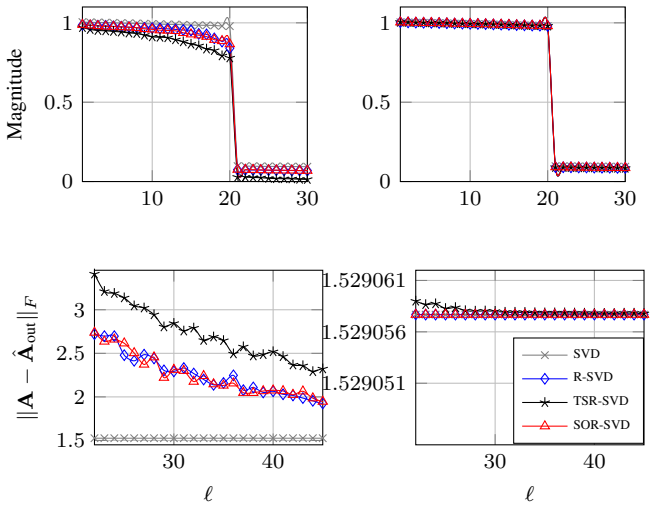

Fig. 1: Top: Comparison of singular values. Bottom: Comparison of low-rank approximation errors. $q = 0$ (left), and $q = 2$ (right).

The sample size parameter ℓ is typically close to the minimal rank k. The simple form of SOR-SVD requires either three or two passes (when $\mathbf{M}$ is approximated by $\mathbf{M}_{\text{approx}}$) through data to factor $\mathbf{A}$. When the power method is incorporated (Alg. 3), SOR-SVD requires either $(2q+3)$ or $(2q+2)$ passes (when $\mathbf{M}$ is approximated by $\mathbf{M}_{\text{approx}}$) over the data with arithmetic costs of $(2q+3)C_{\text{SOR-SVD}}$ or $(2q+2)C_{\text{SOR-SVD}}$, respectively.

Except for matrix-matrix multiplications, which are easily parallelizable, SOR-SVD performs two QR decompositions on matrices of size $m \times \ell$ and $n \times \ell$, whereas R-SVD performs one QR decomposition on an $m \times \ell$ matrix. Recently, Demmel et al. [24] developed communication-avoiding QR decomposition algorithms that perform the computations with optimal communication costs. Thus, this step of both algorithms can be implemented efficiently. In addition, SOR-SVD performs an SVD on an $\ell \times \ell$ matrix $\mathbf{M}$ (or $\mathbf{M}_{\text{approx}}$), whereas the R-SVD performs an SVD on a $n \times \ell$ matrix $\mathbf{B}$ in Alg. 1. Standard techniques to compute an SVD, however, are challenging for parallelization [27], [28]. Given a large input matrix $\mathbf{A}$ for which a rank-k approximation to be computed, where $k \leq \ell \ll \min\{m, n\}$, $\mathbf{M}$ (or $\mathbf{M}_{\text{approx}}$) would be much smaller than $\mathbf{B}$. Considering the size of $\mathbf{M}$ (or $\mathbf{M}_{\text{approx}}$) and, further, having known that current advanced computers have hardware switches that are controlled in software [30], the SVD of the $\ell \times \ell$ matrix can be computed either *in-core* on a sequential processor or with minimum communication cost on parallel processors. Thus, this step of SOR-SVD can be executed efficiently. This significantly reduces the computational time of SOR-SVD, an advantage of the algorithm over R-SVD.

IV. Numerical Experiments

We evaluate the performance of the SOR-SVD algorithm through numerical tests, and compare it against the performance of the SVD, R-SVD, and TSR-SVD algorithms. We also experimentally investigate the tightness of the low-rank approximation error bound for the Frobenius norm provided in Theorem 2. The experiment is implemented in MATLAB.

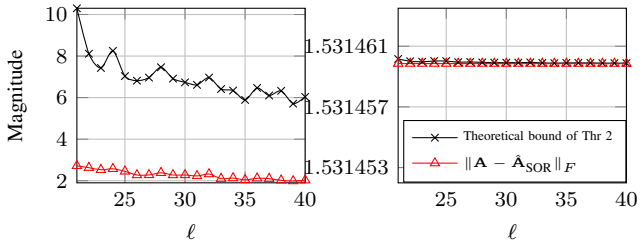

Fig. 2: Comparison of the Frobenius norm error of the SOR-SVD algorithm with the theoretical bound (Theorem 2). No power method, $q = 0$, (left), and $q = 2$ (right).

Due to space constraints, we only consider one type of low-rank matrices to assess the behavior of the SOR-SVD algorithm. For the sake of simplicity, we focus on square matrices.

The matrix of our test is a noisy rank-k matrix $\mathbf{A} \in \mathbb{R}^{1000 \times 1000}$, introduced by Stewart [31], generated as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T + 0.1\sigma_k\mathbf{E}$, where $\mathbf{U}$ and $\mathbf{V}$ are random orthonormal matrices, $\mathbf{\Sigma}$ is diagonal containing the singular values σ_is that decrease geometrically from 1 to 10^{-9}, $\sigma_{k+1} = ... = \sigma_{1000} = 0$, and $\mathbf{E}$ is a normalized Gaussian matrix. We set $k = 20$.

We first compare the singular values computed by the algorithms mentioned. We factor $\mathbf{A}$ using the randomized algorithms with $\ell = 38$, chosen randomly. All randomized algorithms require the same number of passes through data, either two or $2q + 2$. Fig. 1 (top) illustrates the results. The SOR-SVD uses a truncated SVD on the compressed matrix, however we show the results for the full SVD, i.e., a full SVD of the $\ell \times \ell$ matrix, for the sake of comparison. Judging from the figures, the SOR-SVD approximations to both leading and trailing singular values outperform those of the TSR-SVD.

We now compare the reconstruction error of the algorithms with respect to the Frobenius norm; we compute a rank-k approximation $\hat{\mathbf{A}}_{\text{out}}$ to $\mathbf{A}$ by varying the sample size parameter ℓ while the rank is fixed, and calculate the error:

$$e_k = \|\mathbf{A} - \hat{\mathbf{A}}_{\text{out}}\|_F. \tag{16}$$

Fig. 1 (bottom) illustrates the results, demonstrating that the R-SVD and SOR-SVD algorithms have similar behavior when $q = 0$, exceeding the performance of the TSR-SVD algorithm.

A. Empirical Evaluation of SOR-SVD Error Bound

To evaluate the tightness of the bound provided by Theorem 2, we fix the rank $k = 20$ for out test matrix and, by increasing the sample size parameter ℓ, we form $\hat{\mathbf{A}}_{\text{SOR}}$. A comparison between the theoretical bound and what is achieved in practice is shown in Fig. 2. The effect of the power method scheme can be easily seen from the figures; when $q = 2$, the theoretical bound closely tracks the error in the low-rank approximation of Alg. 3. We conclude that the theoretical error bound is empirically sharp.

V. Conclusion

In this paper we have proposed SOR-SVD, a randomized algorithm that computes a low-rank approximation of an input matrix. SOR-SVD is computationally efficient and outperforms TSR-SVD in terms of accuracy. SOR-SVD can exploit modern computational architectures better than R-SVD.

ISBN 978-90-827970-1-5 © EURASIP 2018

REFERENCES

[1] V. Chandrasekaran, P. Parrilo, and A. Willsky, "Latent variable graphical model selection via convex optimization," *The Ann. of Stat.*, vol. 40, no. 4, pp. 1935–1967, 2012.

[2] N. Srebro, N. Alon and T. Jaakkola, "Generalization error bounds for collaborative prediction with low-rank matrices," in *NIPS'04*, 2004, pp. 5–27.

[3] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao, "Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices," *Advances in Neural Information Processing Systems (NIPS)*, pp. 2080–2088, 2009.

[4] M. F. Kaloorazi and R. C. de Lamare, "Low-Rank and Sparse Matrix Recovery Based on a Randomized Rank-Revealing Decomposition," in *22nd Intl Conf. on Digital Signal Processing 2017, UK*, Aug 2017.

[5] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, "Hankel Matrix Rank Minimization with Applications to System Identification and Realization," *SIAM. J. Matrix Anal. & Appl.*, vol. 34, no. 3, pp. 946–977, Apr 2013.

[6] M. Mardani, G. Mateos, and G. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE Journal on Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 50–66, 2013.

[7] M. F. Kaloorazi and R. C. de Lamare, "Anomaly Detection in IP Networks Based on Randomized Subspace Methods," in *ICASSP 2017*, Mar 2017.

[8] B. Victor, K. Bowyer, and S. Sarkar, "An evaluation of face and ear biometrics," in *ICPR 2002*, vol. 1, 2002, pp. 429–432.

[9] R. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Processing Based on Joint and Iterative Interpolation, Decimation, and Filtering," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2503–2514, Jul 2009.

[10] R. Dunia and S. Qin, "A Subspace Approach to Multidimensional Fault Identification and Reconstruction," *AIChE J*, vol. 44, no. 8, pp. 1813–1831, Aug 1998.

[11] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Univ. Press, Baltimore, MD, (1996).

[12] T. F. Chan, "Rank revealing QR factorizations," *Linear Algebra and its Applications*, vol. 88–89, pp. 67–82, Apr 1987.

[13] A. Frieze, R. Kannan, and S. Vempala, "Fast monte-carlo algorithms for finding low-rank approximations," *J. ACM*, vol. 51, no. 6, pp. 1025–1041, Nov. 2004.

[14] P. Drineas, R. Kannan, and M. Mahoney, "Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 158–183, Jul 2006.

[15] T. Sarlós, "Improved approximation algorithms for large matrices via random projections," in *47th Ann. IEEE Symp. on Foundations of Computer Science. FOCS '06.*, vol. 1, Oct. 2006.

[16] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, vol. 31, no. 3, pp. 1100–1124, 2009.

[17] N. Halko, P.-G. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, Jun 2011.

[18] M. Gu, "Subspace Iteration Randomization and Singular Value Problems," *SIAM Journal on Scientific Computing*, vol. 37, no. 3, pp. A1139–A1173, 2015.

[19] A. Deshpande and S. Vempala, "Adaptive Sampling and Fast Low-Rank Matrix Approximation," *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, vol. 4110, pp. 292–303, 2006.

[20] M. Rudelson and R. Vershynin, "Sampling from large matrices: An approach through geometric functional analysis," *J. ACM*, vol. 54, no. 4, Jul. 2007.

[21] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Contemporary Mathematics*, vol. 26, 1984, pp. 189–206.

[22] J. Nelson and H. Nguyen, "Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings," in *Proc. of 54th Annual Symp. on FOCS '13*, 2013, pp. 117–126.

[23] K. Clarkson and D. Woodruff, "Low-rank approximation and regression in input sparsity time," *J. ACM*, vol. 63, no. 6, pp. 54:1–54:45, Jan. 2017.

[24] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou, "Communication-optimal Parallel and Sequential QR and LU Factorizations," *SIAM Journal on Scientific Computing*, vol. 34, no. 1, pp. A206–A239, 2012.

[25] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[26] L. Mirsky, "Symmetric gauge functions and unitarily invariant norms," *Quarterly Journal of Mathematics*, vol. 11, no. 1, pp. 50–59, 1960.

[27] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

[28] P.-G. Martinsson, G. Quintana-Orti, and N. Heavner, "randUTV: A blocked randomized algorithm for computing a rank-revealing UTV factorization," *Arxiv preprint arXiv:1703.00998*, 2017.

[29] M. F. Kaloorazi and R. C. de Lamare, "Subspace-Orbit Randomized Decomposition for Low-Rank Matrix Approximations," *IEEE Transactions on Signal Processing*, 2018.

[30] J. Dongarra, S. Tomov, P. Luszczek, J. Kurzak, M. Gates, I. Yamazaki, H. Anzt, A. Haidar, and A. Abdelfattah, "With Extreme Computing, the Rules Have Changed," *Computing in Science and Engineering*, vol. 19, no. 3, pp. 52–62, May 2017.

[31] G. W. Stewart, "The QLP Approximation to the Singular Value Decomposition," *SIAM Journal on Scientific Computing*, vol. 20, no. 4, pp. 1336–1348, 1999.

ISBN 978-90-827970-1-5 © EURASIP 2018