LOW-RANK MATRIX APPROXIMATION BASED ON INTERMINGLED RANDOMIZED DECOMPOSITION

Maboud F. Kaloorazi and Jie Chen

Center of Intelligent Acoustics and Immersive Communications (CIAIC)
School of Marine Science and Technology
Northwestern Polytechnical University, China
Emails: kaloorazi@nwpu.edu.cn, dr.jie.chen@ieee.org

ABSTRACT

This work introduces a novel matrix decomposition method termed Intermingled Randomized Singular Value Decomposition (InR-SVD), along with an InR-SVD variant powered by the power iteration scheme. InR-SVD computes a low-rank approximation to an input matrix by means of random sampling techniques. Given a large and dense $m \times n$ matrix, InR-SVD constructs a low-rank approximation with a few passes over the data in $O(mnk)$ floating-point operations, where k is much smaller than m and n. Furthermore, InR-SVD can exploit modern computational platforms and thereby being optimized for maximum efficiency. InR-SVD is applied to synthetic data as well as real data in image reconstruction and robust principal component analysis problems. Simulations show that InR-SVD outperforms existing approaches.

Index Terms— Matrix decomposition, randomized algorithms, low-rank approximation, image reconstruction, robust PCA.

1. INTRODUCTION

Low-rank approximation is constructing an approximation to an input matrix by one of lower rank. The compact low-rank representation captures most features of a high-dimensional matrix and thereby results in significant reduction in memory requirements and, more importantly, computational costs. Matrices with low-rank structure appear in many applications such as background subtraction [1–3], IP network anomaly detection [4, 5], latent variable graphical modeling [6], system identification [7], subspace clustering [8, 9] sensor and multichannel signal processing [10], subspace estimation over networks [11], and tensor decompositions [12].

Singular value decomposition (SVD) [13] and the rank-revealing QR (RRQR) decomposition [14,15] are the most commonly used deterministic algorithms for computing a low-rank approximation of a matrix. The bottlenecks of using these algorithms, however, are that (i) they are computationally expensive and (ii) standard techniques for their computation are challenging to parallelize in order to utilize modern architectures [16, 17]. Recently developed algorithms for low-rank approximations based on random sampling schemes have been shown to be remarkably efficient, highly accurate, robust, and are known to outperform traditional algorithms in many practical situations [16–19]. The power of randomized algorithms lies in that (i) they are computationally efficient, and (ii) their main operations can be optimized for maximum efficiency on modern architectures.

In this paper, we develop a randomized decompositional method termed intermingled randomized SVD (InR-SVD), which constructs a low-rank approximation to an input matrix. InR-SVD requires a few passes through data for a large and dense $m \times n$ matrix stored externally, and runs in $O(mnk)$ floating-point operations (flops), where $k \ll \min\{m, n\}$. The main operations of InR-SVD include matrix-matrix multiplication and the QR factorization. Due to recently developed Communication-Avoiding QR (CAQR) algorithms [20], which perform the computations with minimum communication costs, InR-SVD can be optimized for peak machine performance on advanced architectures. Through numerical examples we illustrate that InR-SVD provides highly accurate low-rank approximation, as accurate as the optimal SVD, to a given matrix. We further apply InR-SVD to treat an image reconstruction problem, as well as to solve the robust principal component analysis (robust PCA) problem, i.e., to decompose a matrix with grossly corrupted entries into a low-rank matrix plus a sparse matrix of outliers [21, 22].

We structure the rest of this paper as follows. In Section 2, we discuss the related works and the problem on which this work is focused. In Section 3, we describe the proposed InR-SVD method in detail. In Section 4, we develop an algorithm for robust PCA by using InR-SVD. In Section 5, we present and discuss our numerical experimental results. Conclusions are given in Section 6.

2. RELATED WORKS AND PROBLEM STATEMENT

Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, where $m \geq n$, with numerical rank k, its SVD [13] is defined as:

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

where $\mathbf{U}_k \in \mathbb{R}^{m\times k}$, $\mathbf{U}_0 \in \mathbb{R}^{m\times(n-k)}$, $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ and $\mathbf{V}_0 \in \mathbb{R}^{n\times(n-k)}$ 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 matrices containing the singular values, i.e., $\mathbf{\Sigma}_k = \mathrm{diag}(\sigma_1, ..., \sigma_k)$ and $\mathbf{\Sigma}_0 = \mathrm{diag}(\sigma_{k+1}, ..., \sigma_n)$. $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{A}_k + \mathbf{A}_0$, where $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$, and $\mathbf{A}_0 = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^T$. The SVD constructs the optimal rank-k approximation $\mathbf{A}_k$ to $\mathbf{A}$, [13] i.e.,

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

where $\|\cdot\|_2$ and $\|\cdot\|_F$ indicate the spectral norm and the Frobenius norm of a matrix, respectively. In this paper we focus on the matrix $\mathbf{A}$ defined. The SVD is highly accurate and provides detailed information on singular subspaces and singular values of a matrix. However, its computation is costly, e.g., $O(mn^2)$ for $\mathbf{A}$. Moreover,

[1]This work was supported in part by NSFC grants 61671382 and 61811530283, and 111 project (B18041).

standard techniques for its computation are challenging for parallelization on advanced computational envirenments [16, 17]. Partial SVD based on Krylov subspace methods, such as the Lanczos and Arnoldi algorithms, is an economic version of the SVD. Partial SVD, for instance, for $\mathbf{A}$ costs $O(mnk)$. However, it suffers from two drawbacks: (i) it is numerically unstable [13,23], and (ii) it does not lend itself to parallel implementations [16, 17]. The latter makes partial SVD unsuitable on modern computational devices. Another widely used algorithm for low-rank approximation is the RRQR decomposition [14]. RRQR for $\mathbf{A}$ takes the following form:

$$\mathbf{A\Pi} = \mathbf{QR} = \mathbf{Q}\begin{bmatrix}\mathbf{R}_{11} & \mathbf{R}_{12}\\ \mathbf{0} & \mathbf{R}_{22}\end{bmatrix}, \tag{3}$$

where $\mathbf{\Pi}$ is a permutation matrix, $\mathbf{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns, $\mathbf{R} \in \mathbb{R}^{n\times n}$ is upper triangular where $\mathbf{R}_{11} \in \mathbb{R}^{k\times k}$ is well-conditioned with $\sigma_{\min}(\mathbf{R}_{11}) = O(\sigma_k)$, and the ℓ_2-norm of $\mathbf{R}_{22} \in \mathbb{R}^{(n-k)\times(n-k)}$ is sufficiently small. RRQR gives a rank-k approximation in $O(mnk)$, however due to its pivoted strategy [13] is unsuitable to apply on modern computational platforms [16, 17, 20].

Recently, low-rank approximation algorithms based on randomized sampling techniques [16–19,24–26] have attracted considerable attention. These algorithms project a large data matrix onto a lower dimensional subspace by means of a random matrix, and apply deterministic methods on the smaller matrix to give an approximation of the matrix. Thus (i) they are computationally efficient, and (ii) they readily lend themselves to parallel implementation. Halko et al. [16] proposed *randomized SVD* (R-SVD) in which a smaller matrix is first formed by linear combinations of columns of the given matrix through randomization. The low-rank approximation is then given through the SVD of a reduced-size matrix. Another algorithm proposed in [16, Section 5.5], which we call two-sided randomized SVD (TSR-SVD), is a *single-pass* method, i.e., it needs only one pass through data. TSR-SVD captures most actions of an input matrix by forming a smaller matrix via linear combinations of both rows and columns of the matrix, and then applies the SVD for further computations. The TSR-SVD algorithm is given in Alg. 1.

Algorithm 1 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 $\mathbf{Y}_1 = \mathbf{A\Omega}_1$ 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 $\mathbf{B}_{\text{approx}} = \mathbf{U\Sigma V} \in \mathbb{R}^{\ell\times\ell}$;
6: $\mathbf{A} \approx (\mathbf{Q}_1\mathbf{U})\mathbf{\Sigma}(\mathbf{Q}_2\mathbf{V})^T$.

In Alg. 1, the dagger † denotes the Moore-Penrose pseudoinverse. The work in [27] proposed an algorithm termed subspace-orbit randomized SVD (SOR-SVD). SOR-SVD alternately projects the matrix onto its column and row space via a random Gaussian matrix. The matrix is then transformed into a lower dimensional subspace, and a truncated SVD follows to construct an approximation.

TSR-SVD gives poor approximation compared to the SVD due to the single-pass strategy. SOR-SVD has shown better performance than TSR-SVD. In this work, we develop a randomized algorithm for low-rank approximation that with comparable flops (i) outperforms TSR-SVD in terms of accuracy, and (ii) can utilize advanced computer architectures better than both TSR-SVD and SOR-SVD.

3. INTERMINGLED RANDOMIZED SINGULAR VALUE DECOMPOSITION

In this section, we present intermingled randomized SVD (InR-SVD). We also present a variant of InR-SVD with power iteration which improves the performance of the algorithm at an extra cost.

The idea behind InR-SVD is first to reduce the dimension of a high-dimensional input matrix through intermingling randomization and orthogonalization, and next to employ the deterministic SVD on the reduced-size data. Finally, the low-rank approximation is constructed by projecting the small data back to the original space. Given the matrix $\mathbf{A}$ and an integer $k \le \ell < n$, the basic form of InR-SVD is computed as follows:

1. Generate a standard Gaussian matrix $\mathbf{\Phi} \in \mathbb{R}^{n\times\ell}$.
2. Compute the matrix product:
$$\mathbf{B}_1 = \mathbf{A\Phi}. \tag{4}$$
The matrix $\mathbf{B}_1 \in \mathbb{R}^{m\times\ell}$ is formed through linear combinations of columns of $\mathbf{A}$ by $\mathbf{\Phi}$.
3. Compute a QR decomposition of $\mathbf{B}_1$:
$$\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1. \tag{5}$$
The matrix $\mathbf{Q}_1$ is an approximate basis for the range of $\mathbf{A}$.
4. Compute the matrix product:
$$\mathbf{B}_2 = \mathbf{A}^T\mathbf{Q}_1. \tag{6}$$
The matrix $\mathbf{B}_2 \in \mathbb{R}^{n\times\ell}$ is constructed by linear combinations of rows of $\mathbf{A}$ by means of orthonormal $\mathbf{Q}_1$.
5. Compute a QR decomposition of $\mathbf{B}_2$:
$$\mathbf{B}_2 = \mathbf{Q}_2\mathbf{R}_2. \tag{7}$$
The matrix $\mathbf{Q}_2$ is an approximate basis for the range of $\mathbf{A}^T$.
6. Compute the matrix product:
$$\mathbf{H} = \mathbf{Q}_1^T\mathbf{AQ}_2. \tag{8}$$
The matrix $\mathbf{H} \in \mathbb{R}^{\ell\times\ell}$ is formed by compression of $\mathbf{A}$ via left and right multiplications by the orthonormal bases.
7. Compute an SVD of $\mathbf{H}$:
$$\mathbf{H} = \mathbf{U\Sigma V}^T. \tag{9}$$
8. Form InR-SVD-based low-rank approximation of $\mathbf{A}$:
$$\hat{\mathbf{A}}_{\text{InR}} = (\mathbf{Q}_1\mathbf{U})\mathbf{\Sigma}(\mathbf{Q}_2\mathbf{V})^T, \tag{10}$$
where $\mathbf{Q}_1\mathbf{U} \in \mathbb{R}^{m\times\ell}$ and $\mathbf{Q}_2\mathbf{V} \in \mathbb{R}^{n\times\ell}$ are approximations to the ℓ leading left and right singular vectors of $\mathbf{A}$, respectively, and $\mathbf{\Sigma}$ contains the corresponding singular values.

Step 6 of the InR-SVD procedure requires $2mn\ell + 2m\ell^2$ flops and one pass through $\mathbf{A}$. However, computation of this step can be totally eliminated as follows:

$$\mathbf{H}^T = \mathbf{Q}_2^T\underbrace{\mathbf{A}^T\mathbf{Q}_1}_{\mathbf{B}_2} = \mathbf{Q}_2^T\mathbf{B}_2 = \mathbf{R}_2. \tag{11}$$

Thus $\mathbf{H} = \mathbf{R}_2^T$ (Note that $\mathbf{Q}_2^T\mathbf{Q}_2 = \mathbf{I}$ since $\mathbf{Q}_2$ is orthonormal).

The key differences between InR-SVD and competing TSR-SVD and SOR-SVD are that (i) InR-SVD uses an orthonormal basis

to project the input matrix onto its row space, while TSR-SVD uses a random matrix and SOR-SVD employs a compressed version of the matrix, (ii) both TSR-SVD and SOR-SVD approximate the compressed matrix of equation (8), even though it is obtained through different strategies, while InR-SVD obviates this step. Discarding the computation of step 6, as explained later, can substantially reduce the cost of InR-SVD compared to TSR-SVD and SOR-SVD.

InR-SVD provides fairly accurate approximations for matrices with decaying singular values. However in applications where the data matrix has a slowly decaying singular spectrum, it may produce poor approximations compared with the SVD. Therefore, we incorporate q steps of the power method [16, 19] to improve the performance of the algorithm in these situations. Given $\mathbf{A}$ and integers $k \leq \ell < n$ and q, the resulting algorithm is presented in Alg. 2.

Algorithm 2 InR-SVD with Power Method

Input: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, integers k, ℓ and q.
Output: A rank-ℓ approximation.
1: Generate a standard Gaussian matrix $\mathbf{\Phi} \in \mathbb{R}^{n\times \ell}$;
2: Compute $\mathbf{B}_1 = (\mathbf{A}\mathbf{A}^T)^q\mathbf{A}\mathbf{\Phi}$;
3: Compute QR decomposition $\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1$,
4: Compute $\mathbf{B}_2 = \mathbf{A}^T\mathbf{Q}_1$;
5: Compute QR decomposition $\mathbf{B}_2 = \mathbf{Q}_2\mathbf{R}_2$;
6: Compute an SVD $\mathbf{R}_2^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$; % See equation (11);
7: Form the low-rank approximation: $\hat{\mathbf{A}}_{\text{InR}} = (\mathbf{Q}_1\mathbf{U})\mathbf{\Sigma}(\mathbf{Q}_2\mathbf{V})^T$.

3.1. Computational Complexity

To factor $\mathbf{A}$, the simple form of InR-SVD incurs the following costs (we only consider high-order terms): Step 1 costs $n\ell$, Step 2 costs $2mn\ell$, Step 3 costs $2m\ell^2$, Step 4 costs $2mn\ell$, Step 5 costs $2n\ell^2$, Step 6 is averted, Step 7 costs $2\ell^3$, and Step 8 costs $2m\ell^2 + 2n\ell^2$. The dominant cost of Steps 1-8 occurs when $\mathbf{A}$ and $\mathbf{A}^T$ are multiplied by corresponding matrices. Thus

$$C_{\text{InR-SVD}} = O(mn\ell). \tag{12}$$

The sample size parameter ℓ, in general, is close to the minimal rank k. The simple version of InR-SVD needs two passes over $\mathbf{A}$ to give an approximation. While, InR-SVD of Alg. 2 needs $2q + 2$ passes over the data with arithmetic cost of $(2q + 2)C_{\text{InR-SVD}}$.

The cost of an algorithm is determined by both arithmetic, i.e., flop counts, and communication, i.e., transfering data between different levels of the memory hierarchy or between processors [20]. On high performance computing devices, for an input matrix stored externally, the cost of moving data becomes considerably more expensive than the arithmetic [20, 28]. InR-SVD carries out several matrix-matrix multiplications, however (unlike TSR-SVD and SOR-SVD) it does not compute the compressed matrix $\mathbf{H}$ in (8). Although matrix-matrix multiplication is readily parallelizable, computing this matrix may still impose a considerable cost on the algorithm when implemented on advanced computational environments. This is an advantage of InR-SVD over TSR-SVD and SOR-SVD. InR-SVD also performs two QR decompositions on $m \times \ell$ and $n \times \ell$ matrices. Demmel et al. [20] developed communication-avoiding QR methods that perform the computations with optimal communication costs. Thus, QR algorithms of InR-SVD can be implemented efficiently (this is also the case for TSR-SVD and SOR-SVD). InR-SVD further performs one SVD on a (structured) upper triangular $\ell\times\ell$ matrix $\mathbf{R}_2$ (7). The decomposition of this matrix can be done in a more efficient way compared to those of TSR-SVD and SOR-SVD (the corresponding matrices for latter methods, which are obtained differently from InR-SVD, do not have a special structure). This is another advantage of InR-SVD over TSR-SVD and SOR-SVD.

4. ROBUST PCA WITH INR-SVD

This section presents a new method to solve the robust PCA problem by employing InR-SVD. Robust PCA [21, 22] represents a grossly perturbed low-rank matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$ as a linear superposition of a clean low-rank matrix $\mathbf{L}$ and a sparse matrix of outliers $\mathbf{S}$ such as $\mathbf{D} = \mathbf{L} + \mathbf{S}$ by solving the following convex program:

$$\begin{aligned} &\text{minimize}_{(\mathbf{L},\mathbf{S})}\ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \\ &\text{subject to } \mathbf{D} = \mathbf{L} + \mathbf{S}, \end{aligned} \tag{13}$$

where $\|\mathbf{N}\|_* \triangleq \sum_i \sigma_i(\mathbf{N})$ refers to the nuclear norm of a matrix $\mathbf{N}$, $\|\mathbf{N}\|_1 \triangleq \sum_{ij} |\mathbf{N}_{ij}|$ refers to the ℓ_1-norm of $\mathbf{N}$, and $\lambda > 0$ is a tuning parameter. One efficient method to solve (13) is the method of augmented Lagrange multipliers (ALM) [29]. The ALM method yields the optimal solution, however its serious bottleneck is computing the costly SVD at each iteration to approximate the low-rank component $\mathbf{L}$ of $\mathbf{D}$ [22, 30]. To address this issue and to speed up the convergence of the ALM method, the work in [30] proposed a few techniques such as predicting the principal singular space dimension, a continuation scheme [31], and a truncated SVD by using PROPACK package [32]. The modified algorithm [30] considerably improved the convergence speed, however the truncated SVD [32] applied uses the Lanczos algorithm that (i) is unstable, and (ii) has poor performance on modern architectures, due to the limited data reuse in its operations [13,16,17,23]. To address this issue, by retaining the original objective function [21,22,30], we apply InR-SVD as a surrogate to the truncated SVD to solve the robust PCA problem. We adopt the continuation technique [30,31] that increases μ in each iteration. The proposed `ALM-InRSVD` method is given in Alg. 3.

Algorithm 3 Robust PCA solved by `ALM-InRSVD`

Input: Matrix $\mathbf{D}, \lambda, \mu, \mathbf{Y}_0 = \mathbf{S}_0 = 0, j = 0$.
Output: Low-rank plus sparse matrix.
1: **while** the algorithm does not converge **do**
2: Compute $\mathbf{L}_{j+1} = \mathcal{J}_{\mu_j^{-1}}(\mathbf{D} - \mathbf{S}_j + \mu_j^{-1}\mathbf{Y}_j)$;
3: Compute $\mathbf{S}_{j+1} = \mathcal{S}_{\lambda\mu_j^{-1}}(\mathbf{D} - \mathbf{L}_{j+1} + \mu_j^{-1}\mathbf{Y})$;
4: Compute $\mathbf{Y}_{j+1} = \mathbf{Y}_j + \mu_j(\mathbf{D} - \mathbf{L}_{j+1} - \mathbf{S}_{j+1})$;
5: Update $\mu_{j+1} = \max(\rho\mu_j, \bar{\mu})$;
6: **end while**
7: **return** $\mathbf{L}^*$ and $\mathbf{S}^*$.

In Alg. 3, for any matrix $\mathbf{N}$ with an InR-SVD such as $\mathbf{N} = \mathbf{U}_{\text{InR}}\mathbf{\Sigma}_{\text{InR}}\mathbf{V}_{\text{InR}}^T$, $\mathcal{J}_\delta(\mathbf{N}) = \mathbf{U}_{\text{InR}}\mathcal{S}_\delta(\mathbf{\Sigma}_{\text{InR}})\mathbf{V}_{\text{InR}}^T$ refers to an InR-SVD thresholding operator, where $\mathcal{S}_\delta(x) = \text{sgn}(x)\max(|x| - \delta, 0)$ is a shrinkage operator [33], λ, μ_0, $\bar{\mu}$, ρ, $\mathbf{Y}_0$, and $\mathbf{S}_0$ are initial values. The main operation of `ALM-InRSVD` is computing InR-SVD in each iteration, which is efficient in terms of flops, $O(mnk)$, and can be computed with minimum communication cost.

5. NUMERICAL EXPERIMENTS

In this section, we present simulations that evaluate the performance of InR-SVD for approximating an input matrix. We illustrate that InR-SVD furnishes highly accurate low-rank approximations, and compare InR-SVD against competing algorithms from the literature. We further employ InR-SVD for solving the robust PCA problem. The simulations were conducted in MATLAB.

5.1. Low-Rank Approximation

We compare the low-rank approximation constructed by our method against those of the SVD [13], RRQR [14], TSR-SVD [16], and

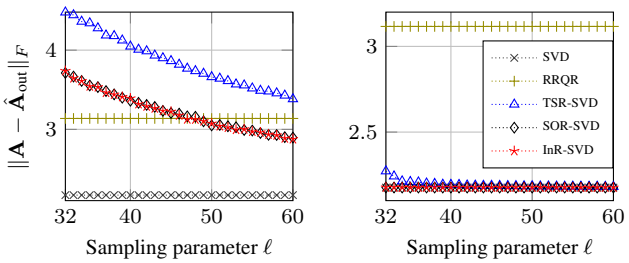

Fig. 1: Comparison of approximation errors. $q = 0$ (left), and $q = 1$ (right).

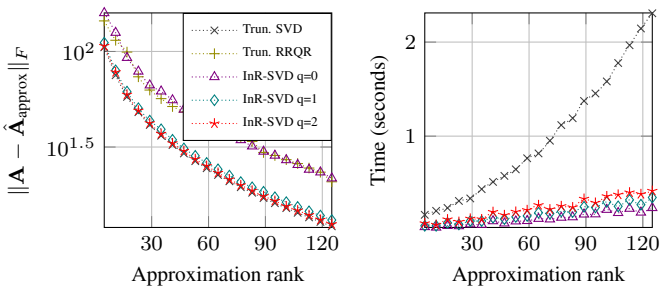

Fig. 3: Comparison of reconstruction errors and runtime.

SOR-SVD [27]. For the randomized algorithms considered, namely InR-SVD, TSR-SVD, and SOR-SVD, the results presented are averaged over 10 trials. Due to space constraints, we consider one type of low-rank matrices, and for simplicity we focus on a square matrix.

We construct a noisy rank-k matrix $\mathbf{A}$ of order 2000 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 consisting of singular values σ_is that decrease linearly from 1 to 10^{-12}, $\sigma_{k+1} = ... = \sigma_{2000} = 0$, and $\mathbf{E}$ is a normalized Gaussian matrix. We set $k = 30$. We construct a rank-k approximation $\hat{\mathbf{A}}_{\text{out}}$ to $\mathbf{A}$ by varying the sample size parameter ℓ with the rank being fixed, and calculate the error $\mathcal{E} = \|\mathbf{A} - \hat{\mathbf{A}}_{\text{out}}\|_F$. The results are shown in Fig. 1. It is observed that (i) when $q = 0$, InR-SVD and SOR-SVD show similar performances, while TSR-SVD shows the worst performance, (ii) when $q = 1$, the result by InR-SVD shows no loss of accuracy compared to the optimal SVD. In this case, RRQR has the poorest performance.

5.2. Image Reconstruction

We assess the quality of low-rank approximation constructed by InR-SVD through reconstructing a gray-scale image of a differential gear of size 1280×804, taken from [34]. We compare our results with those of widely used truncated RRQR and the truncated SVD of PROPACK package [32]. Fig. 2 shows the reconstructed images with $rank = 70$, and Fig. 3 displays the runtime and the approximation error defined as $\mathcal{E}_{approx} = \|\mathbf{A} - \hat{\mathbf{A}}_{\text{approx}}\|_F$, where $\hat{\mathbf{A}}_{\text{approx}}$ is the approximation by each algorithm. It is observed that with $q = 1$ or $q = 2$, the reconstructions (and also errors incurred) by InR-SVD match those of the optimal truncated SVD.

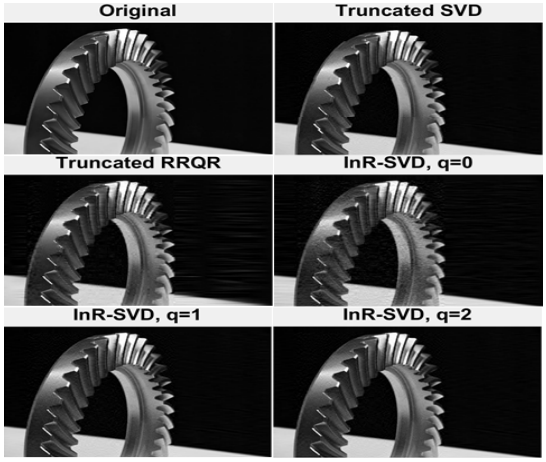

Fig. 2: low-rank image reconstruction with $rank = 70$.

5.3. Robust PCA

We examine the effectiveness of `ALM-InRSVD` of Alg. 3 in recovering low-rank and sparse components of synthetic data. We compare our results with those of the efficient inexact ALM method by [30], called `InexactALM` hereafter. We generate a matrix $\mathbf{D} = \mathbf{L} + \mathbf{S}$ as a linear combination of a rank-k matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ and a sparse error matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$. The matrix $\mathbf{L}$ is generated as $\mathbf{L} = \mathbf{U}\mathbf{V}^T$, where $\mathbf{U}$, $\mathbf{V} \in \mathbb{R}^{n\times k}$ are standard Gaussian matrices. The error matrix $\mathbf{S}$ has s non-zero entries independently drawn from the set {-100, 100}. We apply `ALM-InRSVD` and `InexactALM` to $\mathbf{D}$ to recover $\mathbf{L}$ and $\mathbf{S}$. The numerical results are summarized in Table 1, where the rank of $\mathbf{L}$ is set to $k = 0.05 \times n$ and $s = \|\mathbf{S}\|_0 = 0.05 \times n^2$.

In our simulation, we adopt the initial values suggested in [30]. The algorithms are stopped when $\|\mathbf{D} - \mathbf{L}^{\text{out}} - \mathbf{S}^{\text{out}}\|_F < 10^{-4}\|\mathbf{D}\|_F$ is met, where $(\mathbf{L}^{\text{out}}, \mathbf{S}^{\text{out}})$ is the pair of output of either algorithm. In the Table, $Time$ refers to the runtime in seconds, $Iter.$ refers to the number of iterations, and $\zeta = \|\mathbf{D} - \mathbf{L}^{\text{out}} - \mathbf{S}^{\text{out}}\|_F / \|\mathbf{D}\|_F$ is defined as the relative error.

Table 1: Numerical results for synthetic data recovery.

n	k	$\|\mathbf{S}\|_0$	Methods	$\hat{k}$	$\|\hat{\mathbf{S}}\|_0$	Time	Iter.	ζ
1000	50	5e4	`InexactALM`	50	5e4	7.8	9	3.1e-5
			`ALM-InRSVD`	50	5e4	3.0	9	4.1e-5
2000	100	2e5	`InexactALM`	100	2e5	40.7	9	4.1e-5
			`ALM-InRSVD`	100	2e5	17.7	9	4.7e-5
3000	150	45e4	`InexactALM`	150	45e4	120.9	9	5.1e-5
			`ALM-InRSVD`	150	45e4	57.9	9	5.4e-5

InR-SVD requires a prespecified rank ℓ to perform the factorization. We thus set $\ell = 2k$ as a random start. We also set $q = 1$. The results in Table 1 show that `ALM-InRSVD` successfully detects the exact rank k of the input matrices, and provides the exact optimal solution over 2 times faster than `InexactALM`.

6. CONCLUSION

In this paper, we presented InR-SVD for computing a low-rank approximation of an input matrix. Simulations show that InR-SVD provides approximations as accurate as those of the optimal SVD. InR-SVD is computationally more efficient than SVD, RRQR, and outperforms TSR-SVD in accuracy. InR-SVD can better exploit advanced architectures by leveraging higher levels of parallelism than SVD, RRQR, TSR-SVD and SOR-SVD. We further applied InR-SVD to reconstruct a low-rank image, as well as to solve the robust PCA problem via the ALM method. Simulations show that proposed `ALM-InRSVD` outperforms efficiently implemented `InexactALM`.

7. REFERENCES

[1] T. Bouwmans, S. Javed, H. Zhang, H. Lin, and R. Otazo, "On the Applications of Robust PCA in Image and Video Processing," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1427–1457, Aug 2018.

[2] 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 DSP 2017, UK*, Aug 2017.

[3] ——, "Compressed Randomized UTV Decompositions for Low-Rank Matrix Approximations," *IEEE Journal of Selected Topics in Signal Process.*, vol. 12, no. 6, pp. 1–15, Dec 2018.

[4] ——, "Anomaly Detection in IP Networks Based on Randomized Subspace Methods," in *ICASSP, USA*, Mar 2017, pp. 4222–4226.

[5] A. M. J. Niyaz Hussain and M. Priscilla, "A Survey on Various Kinds of Anomalies Detection Techniques in the Mobile Adhoc Network Environment," *Int. J. S. Res. CSE & IT.*, vol. 3, no. 3, pp. 1538–1541, 2018.

[6] V. Chandrasekaran, P. Parrilo, and A. Willsky, "Latent Variable Graphical Model Selection via Convex Optimization," *The Ann. of Stat.*, vol. 40, no. 4, p. 1935–1967, 2012.

[7] 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.

[8] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès, "Robust Subspace Clustering," *Annals of Statistics*, vol. 42, no. 2, pp. 669—-699, 2014.

[9] M. Rahmani and G. Atia, "Coherence Pursuit: Fast, Simple, and Robust Principal Component Analysis," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6260–6275, Dec 2017.

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

[11] J. Chen, C. Richard, and A. H. Sayed, "Multitask Diffusion Adaptation Over Networks With Common Latent Representations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 563–579, Apr 2017.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A Multilinear Singular Value Decomposition," *SIAM J. Matrix Anal. & Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

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

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

[15] M. Gu and S. C. Eisenstat, "Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization," *SIAM J. Sci. Comput.*, vol. 17, no. 4, p. 848–869, 1996.

[16] 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.

[17] M. Gu, "Subspace Iteration Randomization and Singular Value Problems," *SIAM J. Sci. Comput.*, vol. 37, no. 3, pp. A1139–A1173, 2015.

[18] 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.

[19] V. Rokhlin, A. Szlam, and M. Tygert, "A Randomized Algorithm for Principal Component Analysis," *SIAM. J. Matrix Anal. & Appl.*, vol. 31, no. 3, pp. 1100–1124, 2009.

[20] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou, "Communication-optimal Parallel and Sequential QR and LU Factorizations," *SIAM J. Sci. Comput.*, vol. 34, no. 1, p. A206–A239, 2012.

[21] V. Chandrasekaran, S. Sanghavi, P. a. Parrilo, and A. S. Willsky, "Rank-Sparsity Incoherence for Matrix Decomposition," *SIAM J. Opt.*, vol. 21, no. 2, pp. 572–596, 2009.

[22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, May 2011.

[23] D. Calvetti, L. Reichel, and D. Sorensen, "An Implicitly Restarted Lanczos Method for Large Symmetric Eigenvalue Problems," *ETNA*, vol. 2, pp. 1–21, 1994.

[24] J. Tropp, A. Yurtsever, M. Udell, and V. Cevher, "Practical Sketching Algorithms for Low-Rank Matrix Approximation," *SIAM J. Matrix Anal. & Appl.*, vol. 38, no. 4, pp. 1454–1485, 2017.

[25] M. F. Kaloorazi and R. C. de Lamare, "Subspace-Orbit Randomized-Based Decomposition for Low-Rank Matrix Approximations," in *26th European Signal Processing Conference (EUSIPCO)*, Sep 2018, pp. 2618–2622.

[26] M. F. Kaloorazi, *Low-Rank Matrix Approximations and Applications*, PhD Thesis, Pontifical Catholic University of Rio de Janeiro, Brazil (2018).

[27] M. F. Kaloorazi and R. C. de Lamare, "Subspace-Orbit Randomized Decomposition for Low-Rank Matrix Approximations," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4409–4424, Aug 2018.

[28] 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.

[29] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Method*, Academic Press, (1982).

[30] Z. Lin, R. Liu, and Z. Su, "Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation," in *NIPS*, no. 1, 2011, pp. 1–9.

[31] K. C. Toh and S. Yun, "An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Linear Least Squares Problems," *Pac. J. Optim.*, vol. 6, no. 3, pp. 615–640, 2010.

[32] R. M. Larsen, *Efficient Algorithms for Helioseismic Inversion*, PhD Thesis, University of Aarhus, Denmark (1998).

[33] E. Hale, W. Yin, and Y. Zhang, "Fixed-Point Continuation for ℓ_1-Minimization: Methodology and Convergence," *SIAM J. Opt.*, vol. 19, no. 3, pp. 1107–1130, 2008.

[34] J. A. Duersch and M. Gu, "Randomized QR with Column Pivoting," *SIAM J. Sci. Comput.*, vol. 39, no. 4, pp. C263–C291, 2017.