

# COMPRESSED RANDOMIZED UTV DECOMPOSITIONS FOR LOW-RANK MATRIX APPROXIMATIONS IN DATA SCIENCE

*Maboud F. Kaloorazi and Rodrigo C. de Lamare*

Centre for Telecommunications Studies (CETUC)  
Pontifical Catholic University of Rio de Janeiro, Brazil  
Department of Electronics, University of York, United Kingdom  
Emails: {kaloorazi, delamare}@cetuc.puc-rio.br

## ABSTRACT

In this work, a novel rank-revealing matrix decomposition algorithm termed Compressed Randomized UTV (CoR-UTV) decomposition along with a CoR-UTV variant aided by the power method technique is proposed. CoR-UTV computes an approximation to a low-rank input matrix by making use of random sampling schemes. Given a large and dense matrix of size  $m \times n$  with numerical rank  $k$ , where  $k \ll \min\{m, n\}$ , CoR-UTV requires a few passes over the data, and runs in  $O(mnk)$  floating-point operations. Furthermore, CoR-UTV can exploit modern computational platforms and can be optimized for maximum efficiency. CoR-UTV is also applied for solving robust principal component analysis problems. Simulations show that CoR-UTV outperform existing approaches.

**Index Terms**— Rank-revealing decompositions, low-rank approximations, randomized algorithms, robust PCA.

## 1. INTRODUCTION

Low-rank matrix approximations play an increasingly important role in signal processing and its applications. Such compact representations which retain the key features of a high-dimensional matrix provide a significant reduction in memory requirements, and more importantly, computational costs when the latter scales, e.g., according to a high-degree polynomial, with the dimensionality. Matrices with low-rank structures have found many applications in background subtraction [1, 2], system identification [3], IP network anomaly detection [4, 5], latent variable graphical modeling [6], subspace clustering [7, 8] and sensor and multichannel signal processing [9].

Singular value decomposition (SVD) [10] and the rank-revealing QR (RRQR) decomposition [11, 12] are among the most commonly used algorithms for computing a low-rank approximation of a matrix. On the other hand, a UTV decomposition [13] is a compromise between the SVD and the RRQR decomposition with the virtues of both: UTV (i) is more efficient than the SVD, and (ii) provides information on the numerical null space of the matrix [13]. Given a matrix  $\mathbf{A}$ , the UTV algorithm computes a decomposition  $\mathbf{A} = \mathbf{UTV}^T$ , where  $\mathbf{U}$  and  $\mathbf{V}$  have orthonormal columns, and  $\mathbf{T}$  is triangular (either lower or upper triangular). These deterministic algorithms, however, are computationally expensive for large data sets. Furthermore, standard techniques for their computation are challenging to parallelize in order to utilize advanced computer architectures [14, 15]. Recently developed algorithms for low-rank approximations based on random sampling schemes, however, have been shown to be remarkably efficient, highly accurate and robust, and are known to outperform existing algorithms in many practical situations [14–17]. 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.

This work presents a novel randomized rank-revealing method termed compressed randomized UTV (CoR-UTV) decomposition. Given a large and dense rank- $k$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , CoR-UTV computes a low-rank approximation  $\hat{\mathbf{A}}_{\text{CoR}}$  of  $\mathbf{A}$  such that

$$\hat{\mathbf{A}}_{\text{CoR}} = \mathbf{UTV}^T, \quad (1)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  have orthonormal columns, and  $\mathbf{T}$  is triangular (either lower or upper, whichever is preferred). CoR-UTV only requires a few passes through data, for a matrix stored externally, and runs in  $O(mnk)$  floating-point operations (flops). The operations of CoR-UTV involve matrix-matrix multiplication, the QR and RRQR decompositions. Due to recently developed Communication-Avoiding QR algorithms [18–20], which can perform the computations with optimal/minimum communication costs, CoR-UTV can be optimized for peak machine performance on modern architectures. We illustrate, through numerical examples, that CoR-UTV is rank-revealer and provides a highly accurate low-rank approximation to a given matrix. Furthermore, we apply CoR-UTV to solve the robust principal component analysis (robust PCA) problem [21, 22], i.e., to decompose a given matrix with grossly corrupted entries into a low-rank matrix plus a sparse matrix of outliers.

The rest of this paper is structured as follows. In Section 2, we introduce the mathematical model of the data and discuss related works. In Section 3, we describe the proposed CoR-UTV method in detail. In Section 4, we develop an algorithm for robust PCA using CoR-UTV. In Section 5, we present and discuss simulation results and conclusions are given in Section 6.

## 2. MATHEMATICAL MODEL AND RELATED WORKS

Given a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , where  $m \geq n$ , with numerical rank  $k$ , its SVD [10] is defined as:

$$\mathbf{A} = \mathbf{U}_A \mathbf{\Sigma}_A \mathbf{V}_A^T = \underbrace{\begin{bmatrix} \mathbf{U}_k & \mathbf{U}_0 \end{bmatrix}}_{\mathbf{U}_A \in \mathbb{R}^{m \times n}} \underbrace{\begin{bmatrix} \mathbf{\Sigma}_k & 0 \\ 0 & \mathbf{\Sigma}_0 \end{bmatrix}}_{\mathbf{\Sigma}_A \in \mathbb{R}^{n \times n}} \underbrace{\begin{bmatrix} \mathbf{V}_k & \mathbf{V}_0 \end{bmatrix}^T}_{\mathbf{V}_A^T \in \mathbb{R}^{n \times n}}, \quad (2)$$

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.,  $\mathbf{\Sigma}_k = \text{diag}(\sigma_1, \dots, \sigma_k)$  and  $\mathbf{\Sigma}_0 = \text{diag}(\sigma_{k+1}, \dots, \sigma_n)$ .  $\mathbf{A}$  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  $\mathbf{A}$ , [10] i.e.,

$$\begin{aligned} \|\mathbf{A} - \mathbf{A}_k\|_2 &= \sigma_{k+1}, \\ \|\mathbf{A} - \mathbf{A}_k\|_F &= \sqrt{\sigma_{k+1}^2 + \dots + \sigma_n^2}, \end{aligned} \quad (3)$$

where  $\|\cdot\|_2$  and  $\|\cdot\|_F$  denote the spectral norm and the Frobenius norm, respectively. In this paper we focus on the matrix  $\mathbf{A}$  defined.

The SVD is highly accurate for computing singular subspaces and singular values. However, its computation is costly for large data sets. Moreover, standard techniques for its computation are challenging to parallelize in order to take advantage of modern processors [14, 15]. An economical version of the SVD is the partial SVD based on Krylov subspace methods, such as the Lanczos and Arnoldi algorithms, which constructs an approximate SVD of an input matrix, for instance  $\mathbf{A}$ , at a cost  $O(mnk)$ . However, the partial SVD suffers from two drawbacks: (i) it is numerically unstable [10, 23], and (ii) it does not lend itself to parallel implementations [14, 15], which makes it unsuitable for modern architectures. Other approaches for low-rank matrix approximations include the RRQR [11] and the UTV decompositions [13]. Even though the QR with column pivoting (QRCP) and UTV decompositions provide highly accurate approximations to  $\mathbf{A}$ , they suffer from two drawbacks: (i) they are costly, i.e.,  $O(mn^2)$ , and (ii) methods for their computation are challenging to parallelize and hence they cannot exploit modern computational platforms [14, 15].

Recently developed algorithms for low-rank approximations based on randomization [14–17, 24] have attracted significant attention. The randomized algorithms project a large input matrix onto a lower dimensional space using a random matrix, and apply deterministic methods on the smaller matrix to give an approximation of the matrix. Hence (i) they are computationally efficient, and (ii) lend themselves to parallel implementation. Halko et al. [14] proposed *randomized SVD* (R-SVD) in which a smaller matrix is formed by linear combinations of columns of the given matrix. The low-rank approximation is then given through the SVD of a reduced-size matrix. Gu [15] applied a slightly modified version of the R-SVD algorithm to improve subspace iteration methods, and presents a new error analysis. Another algorithm proposed in [14, Section 5.5], which we call two-sided randomized SVD (TSR-SVD), is a *single-pass* method, i.e., it required only one pass through data. It captures most attributes of the data by means of forming the smaller matrix through linear combinations of both rows and columns of the given matrix, and then applies the SVD for further computations. The work in [25] proposed a randomized algorithm termed subspace-orbit randomized SVD (SOR-SVD). SOR-SVD alternately projects the matrix onto its column and row space. The matrix is then transformed into a lower dimensional space, and a truncated SVD follows in order to construct an approximation.

TSR-SVD gives poor approximation compared to the optimal SVD due to the single-pass strategy. SOR-SVD has shown better performance than TSR-SVD, however both methods apply the SVD on the reduced-size matrix. This computation may be burdensome in terms of communication cost [18] for large matrices. In this work, we develop 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 TSR-SVD as well as SOR-SVD.

### 3. COMPRESSED RANDOMIZED UTV DECOMPOSITIONS

In this section, we present a randomized rank-revealing decomposition algorithm termed compressed randomized UTV (CoR-UTV) decomposition, which computes a low-rank approximation of a given matrix. We focus on the matrix  $\mathbf{A}$  with  $m \geq n$ , where CoR-UTV, in the form of (1), produces an upper triangular middle matrix  $\mathbf{T}$ . The modifications required for a CoR-UTV for the case  $m < n$  that produces a lower triangular middle matrix  $\mathbf{T}$  is straightforward.

#### 3.1. Proposed CoR-UTV Decompositions

Given the matrix  $\mathbf{A}$  and an integer  $k \leq \ell < \min\{m, n\}$ , the basic version of CoR-UTV is computed as follows: using a random number generator, we form a matrix  $\mathbf{\Psi} \in \mathbb{R}^{n \times \ell}$  with entries drawn independent, identically distributed (i.i.d.) from the standard Gaussian distribution. We then compute the matrix product:

$$\mathbf{C}_1 = \mathbf{A}\mathbf{\Psi}, \quad (4)$$

where  $\mathbf{C}_1 \in \mathbb{R}^{m \times \ell}$  is, in fact, a projection onto the subspace spanned by columns of  $\mathbf{A}$ . Having  $\mathbf{C}_1$ , we form  $\mathbf{C}_2 \in \mathbb{R}^{n \times \ell}$ :

$$\mathbf{C}_2 = \mathbf{A}^T \mathbf{C}_1, \quad (5)$$

where  $\mathbf{C}_2$  is, in fact, a projection onto the subspace spanned by rows of  $\mathbf{A}$ . Using a QR decomposition, we factor  $\mathbf{C}_1$  and  $\mathbf{C}_2$  such that:

$$\mathbf{C}_1 = \mathbf{Q}_1 \mathbf{R}_1, \quad \text{and} \quad \mathbf{C}_2 = \mathbf{Q}_2 \mathbf{R}_2, \quad (6)$$

where  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  are approximate bases for  $\mathcal{R}(\mathbf{A})$  and  $\mathcal{R}(\mathbf{A}^T)$ , respectively. We now compress  $\mathbf{A}$  by left and right multiplications by the orthonormal bases obtained, forming the matrix  $\mathbf{D} \in \mathbb{R}^{\ell \times \ell}$ :

$$\mathbf{D} = \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_2, \quad (7)$$

We then compute a QRCP of  $\mathbf{D}$ :

$$\mathbf{D} = \tilde{\mathbf{Q}} \tilde{\mathbf{R}} \tilde{\mathbf{P}}^T. \quad (8)$$

The CoR-UTV-based low-rank approximation of  $\mathbf{A}$  is given by

$$\hat{\mathbf{A}}_{\text{CoR}} = \mathbf{U} \mathbf{T} \mathbf{V}^T, \quad (9)$$

where  $\mathbf{U} = \mathbf{Q}_1 \tilde{\mathbf{Q}} \in \mathbb{R}^{m \times \ell}$  and  $\mathbf{V} = \mathbf{Q}_2 \tilde{\mathbf{P}} \in \mathbb{R}^{n \times \ell}$  construct approximations to the  $\ell$  leading left and right singular vectors of  $\mathbf{A}$ , respectively, and  $\mathbf{T} = \tilde{\mathbf{R}} \in \mathbb{R}^{\ell \times \ell}$  is upper triangular with diagonals approximating the first  $\ell$  singular values of  $\mathbf{A}$ .

CoR-UTV requires three passes through data, for a matrix stored externally, but it can be altered to revisit the data only once. To this end, the compressed matrix  $\mathbf{D}$  (7) can be approximated as follows: both sides of the currently unknown  $\mathbf{D} = \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_2$  are postmultiplied by  $\mathbf{Q}_2^T \mathbf{\Psi}$ . Having defined  $\mathbf{A} \approx \mathbf{A} \mathbf{Q}_2 \mathbf{Q}_2^T$  and  $\mathbf{C}_1 = \mathbf{A} \mathbf{\Psi}$ , then  $\mathbf{D}_{\text{approx}} = \mathbf{Q}_1^T \mathbf{C}_1 (\mathbf{Q}_2^T \mathbf{\Psi})^\dagger$ .

CoR-UTV is accurate for matrices whose singular values display some decay, however in applications where the data matrix has a slowly decaying singular spectrum, it may produce a poor approximation compared to the SVD. Thus, we incorporate  $q$  steps of a power iteration [14, 17] to improve the accuracy of the algorithm in these circumstances. Given the matrix  $\mathbf{A}$ , and integers  $k \leq \ell < n$  and  $q$ , the resulting algorithm is described in Alg. 1.

---

**Algorithm 1** CoR-UTV with Power Method

---

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

- 1: Draw a standard Gaussian matrix  $\mathbf{C}_2 \in \mathbb{R}^{n \times \ell}$ ;
  - 2: **for**  $i = 1: q + 1$  **do**
  - 3:   Compute  $\mathbf{C}_1 = \mathbf{A}\mathbf{C}_2$ ;
  - 4:   Compute  $\mathbf{C}_2 = \mathbf{A}^T \mathbf{C}_1$ ;
  - 5: **end for**
  - 6: Compute QR decompositions  $\mathbf{C}_1 = \mathbf{Q}_1 \mathbf{R}_1$ ,  $\mathbf{C}_2 = \mathbf{Q}_2 \mathbf{R}_2$ ;
  - 7: Compute  $\mathbf{D} = \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_2$  or  $\mathbf{D}_{\text{approx}} = \mathbf{Q}_1^T \mathbf{C}_1 (\mathbf{Q}_2^T \mathbf{C}_2)^T$ ;
  - 8: Compute a QRCF  $\mathbf{D} = \tilde{\mathbf{Q}} \tilde{\mathbf{R}} \tilde{\mathbf{P}}^T$  or  $\mathbf{D}_{\text{approx}} = \tilde{\mathbf{Q}} \tilde{\mathbf{R}} \tilde{\mathbf{P}}^T$ ;
  - 9: Form the CoR-UTV-based low-rank approximation of  $\mathbf{A}$ :  
 $\hat{\mathbf{A}}_{\text{CoR}} = \mathbf{U} \mathbf{T} \mathbf{V}^T$ ;  $\mathbf{U} = \mathbf{Q}_1 \tilde{\mathbf{Q}}$ ,  $\mathbf{T} = \tilde{\mathbf{R}}$ ,  $\mathbf{V} = \mathbf{Q}_2 \tilde{\mathbf{P}}^T$ .
- 

### 3.2. 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 [18]. On multicore and accelerator-based computers, for a data matrix stored externally, the communication cost becomes substantially more expensive compared to the arithmetic [18, 26]. The randomized algorithms operate on a compressed version of the data matrix rather than a matrix itself and therefore can be organized to exploit modern computational environments better than their classical counterparts.

To decompose  $\mathbf{A}$ , the simple version of CoR-UTV incurs the following costs: Step 1 costs  $n\ell$ , Step 2 costs  $2mn\ell$ , Step 3 costs  $2mn\ell$ , Step 4 costs  $2m\ell^2 + 2n\ell^2$ , Step 5 costs  $m\ell^2 + 2mn\ell$  (if  $\mathbf{D}$  is approximated by  $\mathbf{D}_{\text{approx}}$ , this step costs  $2m\ell^2 + 2n\ell^2 + 3\ell^3$ ), Step 6 costs  $8/3\ell^3$ , Step 7 costs  $2m\ell^2 + 2n\ell$ . The dominant cost of Steps 1-7 occurs when multiplying  $\mathbf{A}$  and  $\mathbf{A}^T$  with the corresponding matrices. Thus

$$C_{\text{CoR-UTV}} = O(mn\ell). \quad (10)$$

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

In addition to matrix-matrix multiplications and QR decompositions, CoR-UTV performs one QRCF on an  $\ell \times \ell$  matrix, however TSR-SVD and SOR-SVD perform an SVD on the  $\ell \times \ell$  matrix. The SVD is more expensive than QRCF and, furthermore, recently developed QRCF algorithms based on randomization can perform the factorization with minimum communication costs [19, 20, 27], while standard techniques to compute an SVD are challenging for parallelization [14, 15]. Hence for large matrices to be factored on high performance computing architectures, where the compressed  $\ell \times \ell$  matrix does not fit into fast memory, the execution time to compute CoR-UTV can be substantially less than those of TSR-SVD and SOR-SVD. This is an advantage of CoR-UTV over TSR-SVD and SOR-SVD.

### 4. ROBUST PCA WITH CoR-UTV

This section describes how to solve the robust PCA problem using the proposed CoR-UTV method. Robust PCA [21, 22] represents an input low-rank matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  whose fraction of entries being corrupted, as a linear superposition of a low-rank matrix  $\mathbf{L}$  and

a sparse matrix of outliers  $\mathbf{S}$  such as  $\mathbf{M} = \mathbf{L} + \mathbf{S}$ , by solving the following convex program:

$$\begin{aligned} & \text{minimize}_{(\mathbf{L}, \mathbf{S})} \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ & \text{subject to } \mathbf{M} = \mathbf{L} + \mathbf{S}, \end{aligned} \quad (11)$$

where  $\|\mathbf{B}\|_* \triangleq \sum_i \sigma_i(\mathbf{B})$  is the nuclear norm of any matrix  $\mathbf{B}$ ,  $\|\mathbf{B}\|_1 \triangleq \sum_{ij} |\mathbf{B}_{ij}|$  is the  $\ell_1$ -norm of  $\mathbf{B}$ , and  $\lambda > 0$  is a tuning parameter. One efficient method to solve (11) is the method of augmented Lagrange multipliers (ALM) [28]. The ALM method yields the optimal solution, however its bottleneck is computing the costly SVD at each iteration to approximate the low-rank component  $\mathbf{L}$  of  $\mathbf{M}$  [22, 29]. To address this issue and to speed up the convergence of the ALM method, the work in [29] proposes a few techniques including predicting the principal singular space dimension, a continuation technique [30], and a truncated SVD by using PROPACK package [31]. The modified algorithm [29] substantially improves the convergence speed, however the truncated SVD [31] employed uses the Lanczos algorithm that (i) is unstable, and (ii) due to the limited data reuse in its operations, has very poor performance on modern architectures [10, 14, 15, 23].

To address this issue, by considering the original objective function proposed in [21, 22, 29], we apply CoR-UTV as a surrogate to the truncated SVD to solve the robust PCA problem. We adopt the continuation technique [29, 30], which increases  $\mu$  in each iteration. The proposed ALM-CoRUTV method is given in Alg. 2.

---

**Algorithm 2** Robust PCA solved by ALM-CoRUTV

---

**Input:** Matrix  $\mathbf{M}$ ,  $\lambda, \mu, \mathbf{Y}_0 = \mathbf{S}_0 = \mathbf{0}, j = 0$ .**Output:** Low-rank plus sparse matrix.

- 1: **while** the algorithm does not converge **do**
  - 2:   Compute  $\mathbf{L}_{j+1} = \mathcal{C}_{\mu_j^{-1}}(\mathbf{M} - \mathbf{S}_j + \mu_j^{-1} \mathbf{Y}_j)$ ;
  - 3:   Compute  $\mathbf{S}_{j+1} = \mathcal{S}_{\lambda \mu_j^{-1}}(\mathbf{M} - \mathbf{L}_{j+1} + \mu_j^{-1} \mathbf{Y}_j)$ ;
  - 4:   Compute  $\mathbf{Y}_{j+1} = \mathbf{Y}_j + \mu_j(\mathbf{M} - \mathbf{L}_{j+1} - \mathbf{S}_{j+1})$ ;
  - 5:   Update  $\mu_{j+1} = \max(\rho \mu_j, \bar{\mu})$ ;
  - 6: **end while**
  - 7: **return**  $\mathbf{L}^*$  and  $\mathbf{S}^*$ .
- 

In Alg. 2, for any matrix  $\mathbf{B}$  with a CoR-UTV decomposition described in Section 3,  $\mathcal{C}_\delta(\mathbf{B})$  refers to a CoR-UTV thresholding operator defined as:

$$\mathcal{C}_\delta(\mathbf{B}) = \mathbf{U}(:, 1:r) \mathbf{T}(1:r, :)\mathbf{V}^T, \quad (12)$$

where  $r$  is the number of diagonals of  $\mathbf{T}$  greater than  $\delta$ ,  $\mathcal{D}_\delta(\mathbf{B})$  refers to a singular value thresholding operator defined as  $\mathcal{D}_\delta(\mathbf{B}) = \mathbf{U}_B \mathcal{S}_\delta(\mathbf{\Sigma}_B) \mathbf{V}_B^T$ , where  $\mathcal{S}_\delta(x) = \text{sgn}(x) \max(|x| - \delta, 0)$  is a shrinkage operator [32],  $\lambda, \mu_0, \bar{\mu}, \rho, \mathbf{Y}_0$ , and  $\mathbf{S}_0$  are initial values. The main operation of ALM-CoRUTV is computing CoR-UTV in each iteration, which is efficient in terms of flops,  $O(mnk)$ , and can be computed with minimum communication costs.

### 5. NUMERICAL EXPERIMENTS

In this section, we present simulations that evaluate the performance of CoR-UTV for approximating a low-rank input matrix. We show that CoR-UTV provides highly accurate singular values and low-rank approximations, and compare CoR-UTV against competing algorithms from the literature. We also employ CoR-UTV for solving the robust PCA problem.

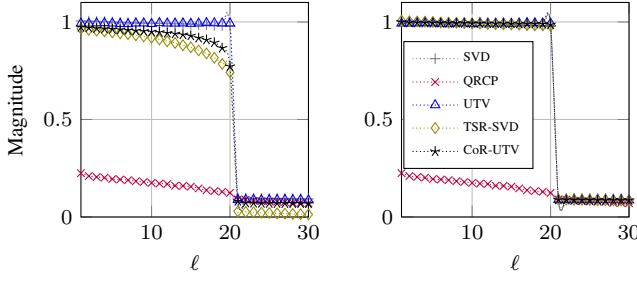


Fig. 1: Comparison of singular values.  $q = 0$  (left), and  $q = 2$  (right).

### 5.1. Rank-Revealing Property & Singular Values Estimation

We first show that CoR-UTV (i) is rank revealer, i.e., the gap in the singular value spectrum of the matrix is revealed, and (ii) provides highly accurate singular values. For the randomized algorithms considered, namely CoR-UTV, TSR-SVD, and SOR-SVD, the results presented are averaged over 20 trials. Each trial was run with the same input matrix with an independent draw of the test matrix. Due to space constraints, we only consider one class of low-rank matrices, and for simplicity we focus on a square matrix.

We construct a noisy rank- $k$  matrix  $\mathbf{A}$  of order  $10^3$  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  $\sigma_i$ s that decrease linearly from 1 to  $10^{-9}$ ,  $\sigma_{k+1} = \dots = \sigma_{10^3} = 0$ , and  $\mathbf{E}$  is a normalized Gaussian matrix. We set  $k = 20$ .

We compare the singular values of the matrix computed by CoR-UTV against those of competing methods such as the SVD [10], QRCP [11], UTV [33] and TSR-SVD [14]. For CoR-UTV and TSR-SVD, we arbitrarily set the sample size parameter to  $\ell = 2k$ . Both algorithms require the same number of passes over  $\mathbf{A}$ , either two or  $2q + 2$  when the power method is used, to perform a factorization.

The results are shown in Fig. 1. It is observed that (i) CoR-UTV strongly reveals the numerical rank  $k$ , (ii) with no power iterations ( $q = 0$ ), CoR-UTV provides very good approximations to singular values and outperforms TSR-SVD in approximating both leading and trailing singular values, (iii) with  $q = 2$ , CoR-UTV delivers singular values as accurate as the optimal SVD, (iv) QRCP only suggests the gap in the singular spectrum, and gives a fuzzy approximation to singular values of the matrix.

### 5.2. Low-Rank Approximation

We now compare the low-rank approximation constructed by our method against those of the SVD, QRCP, TSR-SVD, and SOR-SVD [25]. We construct a rank- $k$  approximation  $\hat{\mathbf{A}}_{\text{out}}$  to  $\mathbf{A}$  by varying the sample size parameter  $\ell$  with the rank fixed, and calculate the error:

$$e_k = \|\mathbf{A} - \hat{\mathbf{A}}_{\text{out}}\|_F. \quad (13)$$

The results are shown in Fig. 2. It is observed that (i) when  $q = 0$ , CoR-UTV and SOR-SVD show similar performances, while TSR-SVD shows the worst performance, (ii) when  $q = 2$ , the errors resulting from CoR-UTV show no loss of accuracy compared to the optimal SVD. In this case, QRCP has the poorest performance.

### 5.3. Robust Principal Component Analysis

Here, we examine the efficiency and efficacy of ALM-CoRUTV in Alg. 2 for recovering the low-rank and sparse components of data.

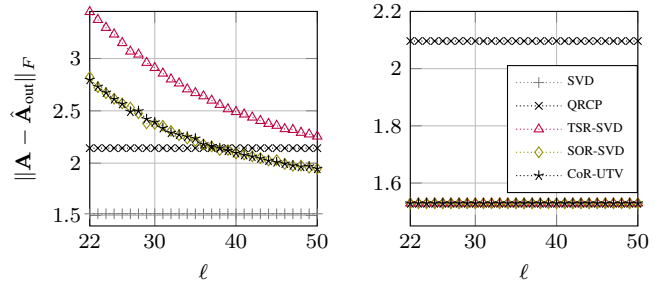


Fig. 2: Comparison of low-rank approximation errors.  $q = 0$  (left), and  $q = 2$  (right).

We compare the results obtained with those of the efficient inexact ALM method by [29], called *InexactALM* hereafter.

We form a rank- $k$  matrix  $\mathbf{M} = \mathbf{L} + \mathbf{S}$  as a linear combination of a low-rank 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  $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times k}$  have standard Gaussian distributed entries. The error matrix  $\mathbf{S}$  has  $s$  non-zero entries independently drawn from the set  $\{-80, 80\}$ . We apply the ALM-CoRUTV and *InexactALM* algorithms to  $\mathbf{M}$  to recover  $\mathbf{L}$  and  $\mathbf{S}$ . The numerical results are summarized in Table 1, where the rank of  $\mathbf{L}$   $r(\mathbf{L}) = 0.05 \times n$  and  $s = \|\mathbf{S}\|_0 = 0.05 \times n^2$ .

In our experiments, we adopt the initial values suggested in [29]. The algorithms are terminated when  $\|\mathbf{M} - \mathbf{L}^{\text{out}} - \mathbf{S}^{\text{out}}\|_F < 10^{-5}\|\mathbf{M}\|_F$  is satisfied, where  $(\mathbf{L}^{\text{out}}, \mathbf{S}^{\text{out}})$  is the pair of output of either algorithm. In the Table, *Time(s)* refers to the runtime in seconds, *Iter.* refers to the number of iterations, and  $\zeta = \|\mathbf{M} - \mathbf{L}^{\text{out}} - \mathbf{S}^{\text{out}}\|_F / \|\mathbf{M}\|_F$  refers to the relative error.

Table 1: Numerical results for synthetic matrix recovery.

$n$	$r(\mathbf{L})$	$\ \mathbf{S}\ _0$	Methods	$r(\mathbf{L}^*)$	$\ \mathbf{S}^*\ _0$	Time(s)	Iter.	$\zeta$
1000	50	5e4	InexactALM	50	5e4	4.1	12	2.1e-6
			ALM-CoRUTV	50	5e4	0.6	12	9.6e-6
2000	100	2e5	InexactALM	100	2e5	27.4	12	2.7e-6
			ALM-CoRUTV	100	2e5	3.7	12	8.3e-6
3000	150	45e4	InexactALM	150	45e4	75.6	12	3.1e-6
			ALM-CoRUTV	150	45e4	9.4	12	8.7e-6

CoR-UTV requires a prespecified rank  $\ell$  to perform the factorization. Thus, we set  $\ell = 2k$ , as a random start, and  $q = 1$ . The results in Table 1 show that ALM-CoRUTV detects the exact rank  $k$  of the input matrix, provides the exact optimal solution, and outperforms *InexactALM* in terms of runtime.

## 6. CONCLUSIONS

In this paper, we have presented CoR-UTV for computing low-rank approximations of an input matrix. Simulations show that CoR-UTV reveals the numerical rank more sharply than QRCP, and provides results as good as those of the optimal SVD. CoR-UTV is more efficient than SVD, QRCP, UTV, TSR-SVD and SOR-SVD in terms of cost. CoR-UTV can better exploit advanced computational platforms by leveraging higher levels of parallelism than all compared algorithms. We have also applied CoR-UTV to solve the robust PCA problem via the ALM method. Simulations show that ALM-CoRUTV outperforms efficiently implemented *InexactALM*.

## 7. REFERENCES

- [1] T. Bouwmans, S. Javed, H. Zhang, Z. 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] 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.
- [4] M. F. Kaloorazi and R. C. de Lamare, "Anomaly Detection in IP Networks Based on Randomized Subspace Methods," in *ICASSP*, 2017.
- [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, pp. 1935–1967.
- [7] 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.
- [8] T.-H. Oh, Y. Matsushita, Y.-W. Tai, and I. S. Kweon, "Fast randomized singular value thresholding for low-rank optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 376–391, Mar. 2017.
- [9] 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.
- [10] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Univ. Press, Baltimore, MD, (1996).
- [11] T. F. Chan, "Rank revealing QR factorizations," *Linear Algebra and its Applications*, vol. 88–89, pp. 67–82, Apr 1987.
- [12] M. Gu and S. C. Eisenstat, "Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization," *SIAM J. Sci. Comput.*, vol. 17, no. 4, pp. 848–869, 1996.
- [13] G. W. Stewart, *Matrix Algorithms: Volume 1: Basic Decompositions*, SIAM, Philadelphia, PA, (1998).
- [14] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, Jun 2011.
- [15] M. Gu, "Subspace Iteration Randomization and Singular Value Problems," *SIAM J. Sci. Comput.*, vol. 37, no. 3, pp. A1139–A1173.
- [16] 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.
- [17] 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.
- [18] 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, pp. A206–A239, 2012.
- [19] J. Demmel, L. Grigori, M. Gu, and H. Xiang, "Communication Avoiding Rank Revealing QR Factorization with Column Pivoting," *SIAM J. Matrix Anal. & Appl.*, vol. 36, no. 1, pp. 55–89, 2015.
- [20] J. A. Dueresch and M. Gu, "Randomized QR with Column Pivoting," *SIAM J. Sci. Comput.*, vol. 39, no. 4, pp. C263–C291, 2017.
- [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 Decomposition for Low-Rank Matrix Approximations," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4409–4424, Aug 2018.
- [26] J. Dongarra, S. Tomov, P. Luszczyk, 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.
- [27] P.-G. Martinsson, G. Quintana Ortí, N. Heavner, and R. van de Geijn, "Householder QR Factorization With Randomization for Column Pivoting (HQRPP)," *SIAM J. Sci. Comput.*, vol. 39, no. 2, pp. C96–C115, 2017.
- [28] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Method*, Academic Press, (1982).
- [29] 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.
- [30] 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.
- [31] R. M. Larsen, *Efficient algorithms for helioseismic inversion*, PhD Thesis, University of Aarhus, Denmark (1998).
- [32] E. Hale, W. Yin, and Y. Zhang, "Fixed-Point Continuation for  $\ell_1$ -Minimization: Methodology and Convergence," *SIAM J. Opt.*, vol. 19, no. 3, pp. 1107–1130, 2008.
- [33] R. D. Fierro, P. C. Hansen, and H. P. S. K., "UTV Tools: Matlab templates for rank-revealing UTV decompositions," *Numerical Algorithms*, vol. 20, pp. 165–194, 1999.