Fast communication

Alternating strategies with internal ADMM for low-rank matrix reconstruction

Kezhi Li \textsuperscript{a,b,\*}, Martin Sundin \textsuperscript{a}, Cristian R. Rojas \textsuperscript{a}, Saikat Chatterjee \textsuperscript{a}, Magnus Jansson \textsuperscript{a}

\textsuperscript{a} ACCESS Linnaeus Center, Electrical Engineering, KTH Royal Institute of Technology, S-100 44 Stockholm, Sweden
\textsuperscript{b} Medical Research Council, Imperial College London, White City, London, W12 0NN, United Kingdom

\textbf{A R T I C L E I N F O}

Article history:
Received 19 March 2015
Received in revised form 24 August 2015
Accepted 10 November 2015
Available online 27 November 2015

Keywords:
Low-rank matrix reconstruction
Alternating strategies
Least squares
ADMM

\textbf{A B S T R A C T}

This paper focuses on the problem of reconstructing low-rank matrices from under-determined measurements using alternating optimization strategies. We endeavour to combine an alternating least-squares based estimation strategy with ideas from the alternating direction method of multipliers (ADMM) to recover low-rank matrices with linear parameterized structures, such as Hankel matrices. The use of ADMM helps to improve the estimate in each iteration due to its capability of incorporating information about the direction of estimates achieved in previous iterations. We show that merging these two alternating strategies leads to a better performance and less consumed time than the existing alternating least squares (ALS) strategy. The improved performance is verified via numerical simulations with varying sampling rates and real applications.

\textcopyright 2015 Elsevier B.V. All rights reserved.

1. Introduction

The low-rank matrix reconstruction problem arises naturally in many fields, such as system identification \cite{1-4}, computer vision \cite{5,6} and quantum state tomography \cite{7}. Suppose an \(r\)-rank matrix \(X\) has size \(n_1 \times n_2\), \(r \leq \min(n_1, n_2)\); the objective is to recover \(X\) from the noisy measurements that satisfy the equation

\[ y = \mathcal{A}(X) + e, \]  \hspace{1cm} (1)

where \(y \in \mathbb{R}^{m \times 1}\) is the measurement vector, \(\mathcal{A}\) denotes a known sensing function \(\mathbb{R}^{n_1 \times n_2} \rightarrow \mathbb{R}^{m \times 1}\), and \(e\) is assumed to be zero-mean noise with known covariance \(E(ee^T) = C \in \mathbb{R}^{m \times m}\). Usually \(m < n_1 \times n_2\), that is, the number of coefficients of \(X\) is larger than the number of measurements, and hence \((1)\) is underdetermined. Specifically we consider the case where \(\mathcal{A}\) is a linear operator so that \((1)\) can be rewritten equivalently as the product of a matrix \(A\) and the vectorization of a low-rank \(X\)

\[ A(X) = \begin{bmatrix} \langle X, A_1 \rangle \\ \vdots \\ \langle X, A_m \rangle \end{bmatrix} = \text{vec}(X), \]  \hspace{1cm} (2)

where \(A \in \mathbb{R}^{m \times n_1 n_2}\) consists of vectorized \(A_i\), as its \(i\)th row, \(i = 1, \ldots, m\). This operator arises in many applications, for instance, in quantum state tomography \cite{7,8}; \(X\) then corresponds to the nearly pure density matrix of the unknown quantum state, and \(A\) represents a series of the measurement bases (e.g., tensor product of Pauli matrices).

Different techniques have been developed to solve such underdetermined problems, such as nuclear norm minimization \cite{9,10} or alternating approaches \cite{11-15}. Compared to the nuclear norm minimization \cite{9}, an alternating technique solution provides faster computation, higher accuracy and is hence useful for solving large-scale underdetermined problems based on different criteria such as maximum likelihood (ML) \cite{13} or least squares (LS)
Typically, alternating approaches provide locally optimal solutions. Each iteration leads to the best solution of a set of variables given another set of fixed variables found in the previous iteration. Here our hypothesis is that, if the updating directions of previous iterations are also considered in each iteration, the reconstruction will be improved in both aspects of accuracy and efficiency because the potential feasible set of solutions is narrowed in each iteration. In this regard, few relevant attempts have been made in matrix completion [11], or to update solutions using a gradient descent method [17,14].

In this paper we develop two algorithms based on the alternating technique. First we incorporate suitable modifications to the alternating least squares (ALS) algorithm of [12] to derive a new algorithm called alternating linear estimator (ALE) for low rank matrices with a linear structure. Then based on the ALS and ALE, we develop a novel algorithm called alternating direction least squares (ADLS) that endeavours to validate our hypothesis on updating directions by fusing two alternating strategies. It utilizes the alternating strategy with the help of an updating direction for structured matrix reconstruction. Inspired by the ALS, the proposed framework is based on running the LS estimation to update the low rank component matrices \( \mathbf{L}, \mathbf{R} \) and \( \mathbf{X} \) iteratively, where \( \mathbf{LR} = \mathbf{X} \). In our new approach, the new \( \mathbf{L}, \mathbf{R} \) are calculated by solving optimization problems involving the augmented Lagrangian to incorporate direction update knowledge. This method is able to push variables converging to solutions more efficiently, as in the standard alternating direction method of multipliers (ADMM) [18,11]. The new algorithm also inherits the capability of ALS of handling structured matrices, e.g., with Hankel structure. The simulation results are compared with the performance of ALS and Cramér–Rao bounds (CRBs). Besides the signal-to-reconstruction error ratio (SRER), we also compare their processing time to show the effectiveness and the efficiency of the proposed approach. Real applications in system identification and inpainting are also demonstrated.

This paper is organized as follows. We review the ALS method in Section 2. In Section 3 we propose the ALE algorithm for reconstructing low rank matrices with linear operators. Then by combining ALE and ADMM, the ADLS algorithm is proposed and analyzed in Section 4. Numerical simulations are shown in Section 5 and finally Section 6 concludes the paper.

Notations: Bold letters are used to denote a vector or a matrix. For vectors, \( \| \cdot \|_1, \| \cdot \|_2, \| \cdot \|_\infty \) represent the \( L_1, L_2 \) and \( L_\infty \) norms, respectively. For matrices, \( \mathbf{A}^T \) and \( \mathbf{A}^* \) denote the transpose and Moore–Penrose pseudoinverse of \( \mathbf{A} \). Moreover, \( \| \cdot \|_F \) represents the Frobenius norm and \( \| \mathbf{x} \|_W \) represents the set of \( r \) matrices. \( \text{vec}(\mathbf{A}) \) represents the column vector of concatenated columns of \( \mathbf{A} \), and \( \text{rank}(\mathbf{A}) = r \) denotes the Kronecker product, and \( \nabla f \) denotes the partial derivative of the function \( f \) with respect to \( \mathbf{X} \). Finally, we use p.s.d. as the short form for positive semidefinite.

### 2. ALS for low-rank matrix reconstruction

The alternating least-squares approach was developed in [12,13]. For an \( r \)-rank matrix \( \mathbf{X} \) satisfying (1) with noise covariance \( \mathbf{C} \), the weighted least-squares estimator is

\[
\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \| \mathbf{y} - \mathbf{A}(\mathbf{X}) \|^2_2. 
\]

To rewrite (3) in terms of the standard 2-norm, the measurements and sensing operator can be prewhitened by forming \( \mathbf{y} = \mathbf{C}^{-1/2} \mathbf{y} \) and \( \mathbf{A} = \mathbf{C}^{-1/2} \mathbf{A} \). Expressing \( \mathbf{X} = \mathbf{LR} \) where \( \mathbf{L} \in \mathbb{R}^{n_1 \times r} \) and \( \mathbf{R} \in \mathbb{R}^{r \times n_2} \), the square of residuals becomes

\[
\| \mathbf{y} - \mathbf{A}(\mathbf{X}) \|^2_2 = \| \mathbf{y} - \mathbf{A} \|^2_2 = \| \mathbf{y} - \mathbf{X} \|^2_2. 
\]

The cost function \( h(\mathbf{L}, \mathbf{R}) \) is minimized alternately by

\[
\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \| \mathbf{y} - \mathbf{A} \|^2_2, \\
\hat{\mathbf{L}} = \arg\min_{\mathbf{L}} \| \mathbf{y} - \mathbf{X} \|^2_2. 
\]

The iterations of estimating \( \hat{\mathbf{L}} \) and \( \hat{\mathbf{R}} \) continue until the residual \( \| \mathbf{y} - \mathbf{A}(\mathbf{X}) \|^2_2 \) no longer decreases. Specifically, we calculate the analytical solutions \( \text{vec}(\mathbf{R}) = [\mathbf{A}(\mathbf{L})]^{-1} \mathbf{y} \) given \( \mathbf{L} \) and \( \text{vec}(\mathbf{L}) = [\mathbf{A}(\mathbf{R}) \times \mathbf{I}_{n_2}]^{-1} \mathbf{y} \) given \( \mathbf{R} \). ALS is also capable of recovering structured low-rank matrices such as Hankel, Toeplitz, as well as p.s.d. matrices. In this case a projection step \( \mathbf{X} = \mathbf{P}(\mathbf{LR}) \) is added after updating \( \mathbf{R} \) using a “lift and project” approach, whose core steps are the truncated singular value decomposition with full explanation in [19]. Then a new \( \mathbf{R} \) is calculated by the least-squares estimator

\[
\mathbf{R} = \min_{\mathbf{R}} \| \mathbf{LR} - \mathbf{X} \|^2_2. 
\]

\( \mathbf{L} \) can be updated likewise. ALS has been shown to be effective for recovering low-rank matrices of large sizes.

### 3. Alternating linear estimator

In this section we develop a simple but efficient algorithm for low rank matrices with linear structure, called the alternating linear estimator (ALE). We assume that the low rank matrix has linear structure, which means that \( \mathbf{X} \in \mathbb{R}^{n_1 \times n_2} \) can be decomposed as

\[
\mathbf{X} = \mathbf{S}_\mathbf{f}(\mathbf{h}) 
\]

where \( \mathbf{h} \in \mathbb{R}^p \) is a parametrization of \( \mathbf{X} \) and \( \mathbf{S}_\mathbf{f} : \mathbb{R}^p \rightarrow \mathbb{R}^{n_1 \times n_2} \) is a linear map parameterizing the linear structure of \( \mathbf{X} \). Taking \( \mathbf{X} \) as a Hankel matrix for instance, \( \mathbf{h} \) can contain the first column and last row of \( \mathbf{X} \). We denote the pseudoinverse of \( \mathbf{S}_\mathbf{f} \) by \( \mathbf{T}_\mathbf{f} \), i.e.,

\[
\mathbf{T}_\mathbf{f}(\mathbf{X}) = \arg\min_{\mathbf{h}} \| \mathbf{X} - \mathbf{S}_\mathbf{f}(\mathbf{h}) \|^2_2. 
\]

The idea of ALE is to iteratively update \( \mathbf{L}, \mathbf{R} \) and apply the lift-and-project (or composite mapping) method to project the matrix to its linear structure. As an initial least squares estimate we set

\[
\mathbf{h}_0 = \arg\min_{\mathbf{h}} \| \mathbf{y} - \mathbf{A}(\mathbf{X}) \|^2_2. 
\]
We then apply lift-and-project to the estimate \( \mathbf{h}_0 \). The details of the algorithm are summarized in Algorithm 1.

### Algorithm 1. Alternating Linear Estimator.

**Input:** sensing operator \( \mathbf{A} \), measurements \( \mathbf{y} \), rank \( r \).

**Set:** residual bound \( \varepsilon \), max number of iterations \( k_{\text{max}} \).

**Initialize:** \( \mathbf{h}_0 \).

**Iterations:** \( k = 0, 1, \ldots, k_{\text{max}} \)

- \( L_{k+1} = \{ \text{first } r \text{ columns of } \mathbf{S}_r(\mathbf{h}_k) \} \)
- \( \mathbf{R}_{k+1} = \mathbf{A}^T(\mathbf{h}_k) \) and \( \mathbf{R}_{k+1} = \mathbf{A}^T(\mathbf{h}_k) \)
- \( \mathbf{h}_{k+1} = T_{\mu}(\mathbf{h}_k + \mathbf{R}_{k+1}) \)
- stop criterion
  - \( \| \mathbf{S}_r(\mathbf{h}_{k+1}) - \mathbf{R}_{k+1} \| \leq \varepsilon \)

**Output:** \( \mathbf{X} = \mathbf{L}_{k+1}\mathbf{R}_{k+1} \).

Since the algorithm only uses projections and mappings which can be performed iteratively, the algorithm is scalable and thus appropriate for large-scale problems. Simulations show that ALE has a better performance than ALS; see details in Section 5.2.

### 4. Alternating direction least-squares estimator

#### 4.1. ADMM embedded to alternating iteration

The ALS estimator leverages the low-rank constraints by forming \( \mathbf{X} = \mathbf{LR} \) and updates the factors iteratively. However, in ALS the changing directions of \( \mathbf{L}, \mathbf{R} \) are not recorded. On the other hand, Algorithm 1 complies with (9) only at the initial step, yet in the iterations it satisfies the low-rank and structured constraints alternatively. Hence the result might drift from the true solution since it does not impose Eq. (1) directly in iterations. To exploit the direction information and overcome the potential drifting problem, we adopt the alternating direction augmented Lagrangian method to strengthen the ALS algorithm [18]. The alternating direction method of multipliers (ADMM) is embedded in each iteration. ADMM is widely used in various applications to solve convex optimization problems effectively and can be used to construct distributed optimization algorithms [18,20]. To minimize \( f(\mathbf{R}) = \| \mathbf{Y} - \mathbf{A} \mathbf{1}_n \| \) and \( g(\mathbf{Z}) = \mu \| \mathbf{L} \mathbf{Z} - \mathbf{X} \|_{\text{F}}^2 \) where \( \mathbf{X} \) denotes the projection result on the structure \( \chi \) (like Hankel) and \( \mu \) is a parameter to balance the weights of \( f \) and \( g \), the problem can be rewritten as

\[
\begin{align*}
\text{minimize} & \quad f(\mathbf{R}) + g(\mathbf{Z}) \\
\text{subject to} & \quad \mathbf{R} = \mathbf{Z},
\end{align*}
\]

where \( f \) and \( g \) pursue solutions within measurement and low-rank constraints, respectively. To robustify the algorithm, an augmented Lagrangian is introduced. Combining the linear and quadratic terms, it becomes

\[
\mathbf{L}_d(\mathbf{R}, \mathbf{Z}, \mathbf{U}) = f(\mathbf{R}) + g(\mathbf{Z}) + \lambda \| \mathbf{R} - \mathbf{Z} + \mathbf{U} \|_{\text{F}}^2,
\]

where \( \lambda > 0 \) is a penalty parameter and \( \mathbf{U} \) is a scaled dual variable associated with the constraint \( \mathbf{R} = \mathbf{Z} \). The Lagrangian in (11) is related to the standard ADMM-Lagrangian in [18] by combining the variables contain \( \mathbf{R} \) and \( \mathbf{Z} \) into a quadratic term. The iteration of updating the right matrix can be formulated by minimizing the augmented Lagrangian over \( \mathbf{R} \) and \( \mathbf{Z} \) cyclically:

\[
\begin{align*}
\mathbf{R}_{k+1} &= \arg \min_{\mathbf{R}} L_d(\mathbf{R}, \mathbf{Z}_k, \mathbf{U}_k) \quad // \quad \mathbf{R} - \min \quad (12) \\
\mathbf{Z}_{k+1} &= \arg \min_{\mathbf{Z}} L_d(\mathbf{R}_{k+1}, \mathbf{Z}, \mathbf{U}_k) \quad // \quad \mathbf{Z} - \min \quad (13) \\
\mathbf{U}_{k+1} &= \mathbf{U}_k + \lambda(\mathbf{R}_{k+1} - \mathbf{Z}_{k+1}) \quad // \quad \text{dual – update} \quad (14)
\end{align*}
\]

Firstly, \( \mathbf{Z}_k \) and \( \mathbf{U}_k \) are fixed and we minimize the augmented Lagrangian over \( \mathbf{R} \). The second step is to pursue a solution to \( \mathbf{Z} \) under the constraints on \( \mathbf{X} \). At last the dual variable \( \mathbf{U}_k \) is updated in the third step. The algorithm runs iteratively until primal and dual residuals become smaller than the terminating bounds \( \| \mathbf{R}_k - \mathbf{Z}_{k-1} \|_2 \leq \varepsilon_{\text{pr}}, \| \lambda(\mathbf{Z}_k - \mathbf{Z}_{k-1}) \|_2 \leq \varepsilon_{\text{dual}} \). \( \mathbf{L} \) can be updated in a similar manner. Though \( f, g \) are general quadratic functions, the ADMM approach benefits from the use of updating directions of previous iterations, and it is capable of pursuing solutions satisfying constraints of measurements, low-rank and structure [18,20] at the same time.

#### 4.2. Alternating direction least-squares

The ADLS algorithm combines the features of the ALE and the ALS algorithm, and embeds ADMM loops to update \( \mathbf{L}, \mathbf{R}, \mathbf{Z}, \mathbf{U} \), respectively. Because functions \( f \) and \( g \) are convex and quadratic, the augmented Lagrangian over \( \mathbf{R}, \mathbf{Z}, \mathbf{U} \) can be minimized explicitly (see Appendix). In particular \( \mathbf{R} \) in (12) becomes

\[
\text{vec}(\mathbf{R}_{k+1}) = \left( \mathbf{P}^T_k \mathbf{P}_k + \lambda I \right)^{-1} \left( \mathbf{P}^T_k \mathbf{Y} + \lambda(\mathbf{Z}_k - \mathbf{U}_k) \right),
\]

where the lowercase letters represent the vectorized matrices, e.g. \( \mathbf{Z} = \text{vec}(\mathbf{Z}); \mathbf{P}_k = \mathbf{A}^T(\mathbf{I}_n \otimes \mathbf{S}_k) \) and \( k \) denotes the index of iterations; \( \mathbf{S} \) is initialized by \( \mathbf{L} \) and \( \mathbf{S}_0 = \mathbf{L}_0 \). For least squares \( g(\mathbf{Z}) \) the \( \mathbf{Z} - \min \) step in (13) is equal to

\[
\mathbf{Z}_{k+1} = \left( \mu \mathbf{S}_k^T \mathbf{S}_k + \lambda I \right)^{-1} \left( \mu \mathbf{S}_k^T \mathbf{X}_k + \lambda(\mathbf{R}_k + \mathbf{U}_k) \right).
\]

The ADMM formulation gives the following update rule for \( \mathbf{L} \):

\[
\text{vec}(\mathbf{L}_{k+1}) = \left( \mathbf{Q}_k^T \mathbf{Q}_k + \lambda I \right)^{-1} \left( \mathbf{Q}_k^T \mathbf{Y} + \lambda(\mathbf{S}_k - \mathbf{T}_k) \right),
\]

where \( \mathbf{Q}_k = \mathbf{A}^T(\mathbf{Z}_k^T + I_\mathbf{1}_n) \) and \( \mathbf{S}_k = \left( \mu \mathbf{X}_k(\mathbf{Z}_k^T + \lambda(\mathbf{L}_k + \mathbf{T}_k)) \right)^{-1} \left( \mu \mathbf{Z}_k^T \mathbf{X}_k^T + \lambda I \right)
\]

Similar to ALE, the structured \( \mathbf{X} \) is obtained in advance by projecting \( \mathbf{S} \mathbf{Z} \) to the constraints set \( \mathbf{x} \):

\[
\mathbf{X}_{k+1} = \mathcal{P}_x(\mathbf{S}_{k+1}^T(\mathbf{Z}_{k+1}^T)),
\]

where \( \mathcal{P}_x(\cdot) = \mathcal{S}_x(\mathcal{T}_x(\cdot)) \). Assuming that the low rank \( \mathbf{X} \) belongs to the set of Hankel matrices \( \mathcal{H} \), the pseudo-code of our algorithm ADLS is presented in Algorithm 2.

#### Algorithm 2. Alternating Direction Least-Squares.

**Input:** sensing operator \( \mathbf{A} \), measurements \( \mathbf{y} \), rank \( r \).

**Set:** residual bound \( \varepsilon \), weight \( \mu \), tunable parameters \( \lambda, \chi \), max number of iterations \( k_{\text{max}} \).

**Initialize:** Perform one iteration of Algorithm 1 to obtain \( \mathbf{h}_0, \mathbf{L}_0, \mathbf{R}_0, \mathbf{h}_1 = \mathbf{L}_0 \mathbf{R}_0, \mathbf{S}_0 = \mathbf{L}_0, \mathbf{U}_0 = \mathbf{0}, \mathbf{Z}_0 = \mathbf{R}_0 \).
\[ T_0 = 0. \]

**Iteration:** for \( k = 0, 1, \ldots, k_{\text{max}} \) do

1. inner loop to calculate the right matrix
   \( R_{k+1} \) is updated by (15);
2. inner loop to calculate the left matrix
   \( L_{k+1} \) is updated by (17);
3. update \( X \) with constraints
   \( X_{k+1} = S_{k+1} Z_{k+1} \);
4. stop criterion
   If \( \| S_{k+1} Z_{k+1} \|_\infty \leq \varepsilon \), break;

**Output:** \( X = S_{k+1} Z_{k+1} \).

In the final step of the stop criterion, the error bound \( \varepsilon \) is determined according to the problem's requirements. In numerical simulations, it is set to be the required SMNR fraction of the Frobenius norm of \( X \). Moreover, when the residual is no longer decreasing, the algorithm will jump out of the main loop and stop.

Algorithm 2 follows the steps to calculate \( R, L \) and \( X \) alternately, so it also belongs to the alternating strategies. It is noted that the three lines in each inner loop are actually run several times before concluding. In contrast to ALS or ALE, in ADLS each step is derived from the analytical solution to its augmented Lagrangian. Compared to ALS, Algorithm 2 uses the projection only once in each iteration, and it is able to achieve a balance between the measurement constraints by function \( f \) and the \( \chi \) constraints by function \( g \) by adjusting \( \mu \). Meanwhile compared to ALE, ADLS also incorporates function \( f \) in the updates so that it prevents the scenario that the output of \( SZ \) drifts away from the true solution to (3).

### 4.3. Discussion

ADLS, \( L \) and \( R \) are updated within an inner loop subject to the constraints rather than calculated directly in ALS. Thus the algorithm is strengthened in efficiency and robustness. When the constraint set \( \chi \) denotes the set of low rank and Hankel set, ADLS can be initialized by the parameterization \( h_0 \) as shown in Algorithm 2, as with any other structure which can be decomposed linearly. Otherwise ADLS is initialized by the singular value decomposition (SVD): \( U \Sigma V^T = \text{mat}_{n_1 \times n_2} (\bar{A} \bar{Y}) \), where \( L_0 = U \sqrt{\Sigma} \) and \( R_0 = \sqrt{\Sigma} V^T \) are rescaled with the square root of singular values to balance the norms of sub-matrices, and \( \Sigma \) is the matrix \( \Sigma \) truncated to the rth singular value. Generally the complexity of SVD for an \( n_1 \times n_2 \) matrix is \( O(n_1^2 + n_2^2) \). Since the SVD is used only once for the initialization, the proposed algorithm is computationally superior to other algorithms based on SVD in particular for large matrix.

ADLS is an advanced algorithm based on ALS. It also inherits many nice features from ALS, such as fast computation compared to methods based on nuclear norm minimization (usually it converges to a solution in less than 25 iterations), fitting for recovering low-rank matrices with certain structure, makes it ideal for various signal processing and system identification problems. For convex problems it has been proven that the residuals and the cost function converge to zero and an optimal value respectively as ADMM proceeds under mild assumptions. The rate of convergence is determined by the choices of \( \lambda, \lambda' \) [18]. In ADLS, \( \lambda, \lambda' \) need to be tuned carefully. Ideas from [18] may be adopted as guidance to tune parameters. In practice, we set \( \lambda, \lambda' \in [0, 1] \) based on numerical experiments. ADLS can also be extended to some special cases of non-convex \( \chi \) such as cardinality or Boolean constraints, in which \( P_\chi \) in (19) should be changed accordingly.

### 5. Numerical experiments

#### 5.1. Performance measure and Cramér–Rao bounds

Extensive simulations have been performed using MATLAB to test the recovery performances of unstructured and structured low-rank matrices contaminated with noise. Similar to the setting in [12], a Hankel matrix \( X \) is generated randomly by creating a matrix with i.i.d. \( N(0, 1) \) elements and fitting \( h \) using Prony’s method [21]. We use the signal-to-reconstruction error ratio (SRER), or namely the inverse of the normalized mean square error (NMSE) to measure the reconstruction result

\[
\text{SRER} = \frac{1}{\text{NMSE}} = \frac{E[\|X\|^2]}{E[\|X - \hat{X}\|^2]} \tag{20}
\]

versus increasing signal to measurement noise ratio (SMNR), \( \text{SMNR} = E[\|X\|^2]/E[\|e\|^2] \) where \( e \sim N(0, \sigma^2 I) \) is the noise. The results are also compared to the Cramér–Rao bound (CRB) defined for unbiased low-rank matrix estimators

\[
\text{CRB}(X) \leq E_{X,Y} \left[ \|X - \hat{X}(Y)\|^2 \right]. \tag{21}
\]

The expressions of CRB for unstructured or structured low rank matrices are derived in [13,22,12]. The CRB for the unstructured low-rank estimator is

\[
\text{CRB}(X) = \text{tr} \left( P^T A C^{-1} A P \right)^{-1} \tag{22}
\]

where \( P \triangleq [V_1 \otimes U_0, V_0 \otimes U_0, U_0] \). Submatrices are obtained from the left-singular vectors, \( U_0 = [u_1, \ldots, u_n] \) and \( U_1 = [v_{1,1}, \ldots, v_{1,n}] \), and right-singular vectors, \( V_0 = [v_{1,1}, \ldots, v_{1,n}] \) and \( V_1 = [v_{1,1}, \ldots, v_{1,n}] \), of \( X \in \mathcal{Y} \).

#### 5.2. Simulation results

Because the performance is evaluated by measurement errors as in (3), we compare the results to the algorithms with same objective functions [12] fairly but not to algorithms focusing on structure errors [3]. The reconstruction performance for low-rank matrices \( X \) with and without the prior knowledge of the Hankel structure is compared in Fig. 1. We choose Hankel structure because it is widely used and can be derived from a linear operator so that we can compare the ALS algorithm with others fairly. Without the prior knowledge, the performance of ADLS is drawn along with the curves of the ALS and the Cramér–Rao
Time versus $\xi$ fraction and observe their SRER. The sampling fraction is given by $\xi = \xi_1/(n_1n_2)$, where $X$ has size $80 \times 80$ and rank $r=4$. The reconstruction gain increases significantly when the matrix structure is taken into consideration. In detail, the performance of both ALE and ADLS is shown to be 2.5 and 0.5 dB better than the results of ALS with respect to Hankel structured and unstructured matrices, respectively. In particular, the curve of ADLS is close to the ALE curve when the SMNR is low, but reveals a better performance in terms of SRER when the SMNR is more than 15 dB.

Table 1

<table>
<thead>
<tr>
<th>SMNR (dB)</th>
<th>5</th>
<th>7.5</th>
<th>10</th>
<th>12.5</th>
<th>15</th>
<th>17.5</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADLS-hankel</td>
<td>0.256</td>
<td>0.245</td>
<td>0.250</td>
<td>0.254</td>
<td>0.251</td>
<td>0.255</td>
<td>0.251</td>
</tr>
<tr>
<td>ALE-hankel</td>
<td>0.065</td>
<td>0.063</td>
<td>0.063</td>
<td>0.060</td>
<td>0.063</td>
<td>0.066</td>
<td>0.062</td>
</tr>
<tr>
<td>ALS-hankel</td>
<td>0.592</td>
<td>0.556</td>
<td>0.570</td>
<td>0.580</td>
<td>0.601</td>
<td>0.669</td>
<td>0.659</td>
</tr>
<tr>
<td>$\xi$</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In order to evaluate the efficiency of the proposed methods, we record the processing time of each algorithm, and calculate their average processing time for each data point of Hankel matrix recovery for fair comparison. Each number is obtained by averaging the time for 500 runs. The time consumed corresponding to the results in Fig. 1(a) and (b) is shown in Table 1(a) and (b), respectively. In Table 1(a), the proposed ALE and ADLS algorithms only consume around one tenth and half of the time consumed by ALS for various noise levels while providing better performances. The different SMNRs do not affect the time significantly. In Table 1(b), on the other hand, the processing time increases when the sampling fraction $\xi$ increases from 0.1 to 0.5. In both tables the proposed methods reveal better performance and efficiency. Particularly ALE has less processing time but can only recover matrices with linear structure while ADLS can deal with more structures and achieve better SRER.

Moreover, ALE/ADLS are capable to solve system identification problems. In this example our objective is to recover the impulse responses of a low-order system by observing a small amount of noisy samples. Write input and output signals $u$ and $y$, respectively. The state–space model can be presented by $x(t+1) = Ax(t) + Bu(t), y(t) = Cx(t)$. In this case the extended observability matrix $O_n = [C; CA; \cdots; CA^{n-1}]$ and the extended controllability matrix $C_n = [B AB \cdots A^{n-1}B]$ give rise to a Hankel matrix $H_{n_1,n_2} = O_n C_n$ with elements $g_k$, $k = 1, \ldots, n_2+n_1$, where $g_k = CA^{k-1}B$ are the impulse response coefficients of the system. Since the system order is equal to the minimum rank of the controllability matrix and the observability matrix, the low-order assumption is usually true for most systems that are not so complicated. Then the system identification problem with known input and noisy outputs can be written as $y = A \cdot \text{vec}(H) + e$, where $y$ is the observed output samples, $A$ represents the sampling
matrices of the form

\[
A = \begin{bmatrix}
u(P) & u(P-1) & \ldots & u(1) \\
u(P+1) & u(P) & \ldots & \quad \quad \quad \quad \\
\quad \quad \quad \quad & \quad \quad \quad \quad & \quad \quad \quad \quad & \quad \quad \quad \quad \\
\quad \quad \quad \quad & \quad \quad \quad \quad & \quad \quad \quad \quad & \quad \quad \quad \quad \\
u(P+m) & \ldots & u(m+1) \\
\end{bmatrix} \quad \star S' \tag{23}
\]

and \( \mathbf{H} \) is the low-rank Hankel matrix that consists of impulse responses. \( \mathbf{S} \) is the matrix representation of the operator \( \mathcal{S} \) in Section 3. In the simulation we used \( N=81 \) impulse response samples, system order \( r=4 \), and we calculated the NMSE of the recovered responses with increasing number of observed output samples. The true system is the fourth order system with the transfer function

\[
\frac{18z - 1.06z^{-1} - 0.99z^{-2} + 0.97z^{-3} - 0.99z^{-4}}{(z - 0.97^2)(z - 0.97^2)(z - 0.97^2)(z - 0.97^2)}.
\]

The recovery results are shown in Fig. 2 where one can see that the proposed algorithms recover the impulse response precisely with a better accuracy comparing to LS and ALS.

Finally, ADLS is applicable to real problems incorporating low rank constraints. To show this we implement ADLS to solve an inpainting problem. Inpainting is the process of reconstructing lost or deteriorated parts of images and videos. Inpainting problems can be solved by finding regions of the image corresponding to local minima of the rank of the associated Hankel matrices [23], where in this scenario the Hankel matrix \( \mathbf{H} \) consists of observed pixels and missing pixels that need to be inpainted. We followed these settings and reconstructed an image using standard partial differential equation (PDE) interpolation [24], ALS and ADLS as shown in Fig. 3. Comparing the results from Fig. 3 we visually observe that both ADLS and ALS are better than the PDE solution. Further, ADLS is a little better than ALS in terms of the peak signal to noise ratio (PSNR) for the recovered images.

6. Conclusion

In this paper two novel algorithms are developed to strengthen the classical ALS algorithm. Firstly an efficient algorithm called alternating linear estimator was proposed for recovering low-rank structured matrices which can be decomposed linearly. Secondly by embedding ADMM steps into the iterations, the alternating direction least square algorithm was developed and analyzed. The algorithms are capable of recovering low-rank matrices in the general underdetermined setup as well as for structured matrices such as Hankel matrices. Simulations indicate that the proposed algorithms achieve performance that is closer to the Cramér–Rao bound than existing methods.
Acknowledgments

This work was partially supported by the Swedish Research Council under contract 621-2011-5847 and by the ACCESS Linnaeus Centre.

Appendix

Proposition 1. The updates of $R$ and $Z$ can be realized as (15) and (16), respectively.

Proof. Because $f$ and $g$ are differentiable, the primal feasibility of $R - Z = 0$ can be substituted with the dual feasibility: $\nabla_R L_\mu(R, Z, U) = 0$ and $\nabla_Z L_\mu(R, Z, U) = 0$. Specifically,

$$0 = \nabla_R (f(R) + \lambda \|R - Z + U\|^2_2) = 2P^T P \text{vec}(R) - 2P^T \gamma + 2 \lambda \text{vec}(Z - Z + U).$$

(24)

Move terms containing $R$ to one side of the equation,

$$\left(P^T P + \lambda I\right) \text{vec}(R) = P^T \gamma + 2 \lambda \text{vec}(Z - U),$$

(25)

from which (15) can be derived straightforwardly. For the $Z$–min step,

$$0 = \nabla_Z \left(g(Z) + \lambda \|R - Z + U\|^2_2\right) = \mu 2S^T (SZ - \hat{X}) - 2\lambda (R - Z + U).$$

(26)

Likewise, move terms containing $Z$ to one side of the equation,

$$(\mu S^T S + \lambda I) Z = \mu S^T \hat{X} + \lambda (R + U),$$

(27)

from which (16) can be derived.

The expressions of $L$ and $S$ in (17) and (18) can be obtained in a similar way.

References


