SIAM J. OPTIM.
Vol. 21, No. 1, pp. 57–81




# RECOVERING LOW-RANK AND SPARSE COMPONENTS OF MATRICES FROM INCOMPLETE AND NOISY OBSERVATIONS*

MIN TAO† AND XIAOMING YUAN‡

**Abstract.** Many problems can be characterized by the task of recovering the low-rank and sparse components of a given matrix. Recently, it was discovered that this nondeterministic polynomial-time hard (NP-hard) task can be well accomplished, both theoretically and numerically, via heuristically solving a convex relaxation problem where the widely acknowledged nuclear norm and $l_1$ norm are utilized to induce low-rank and sparsity. This paper studies the recovery task in the general settings that only a fraction of entries of the matrix can be observed and the observation is corrupted by both impulsive and Gaussian noise. We show that the resulting model falls into the applicable scope of the classical augmented Lagrangian method. Moreover, the separable structure of the new model enables us to solve the involved subproblems more efficiently by splitting the augmented Lagrangian function. Hence, some splitting numerical algorithms are developed for solving the new recovery model. Some preliminary numerical experiments verify that these augmented–Lagrangian-based splitting algorithms are easily implementable and surprisingly efficient for tackling the new recovery model.

**Key words.** matrix recovery, principal component analysis, sparse, low-rank, alternating direction method, augmented Lagrangian method

**AMS subject classifications.** 90C06, 90C22, 90C25, 90C59, 93B30

**DOI.** 10.1137/100781894

## 1. Introduction.

Many applications arising in various areas can be captured by the task of recovering the low-rank and sparse components of a given matrix (usually the matrix represents data obtained by observation), e.g., the model selection in statistics, matrix rigidity in computer science, and system identification in engineering [20, 21, 22, 38, 45, 48, 55]. It is well acknowledged that the nuclear norm (defined as the sum of all singular values) and the $l_1$ norm (defined as the sum of absolute values of all entries) are powerfully capable of inducing low-rank and sparsity, respectively. Hence, in [9], it has been shown that the task of recovering low-rank and sparse components, which is nondeterministic polynomial-time hard (NP-hard), can be accurately accomplished via solving the following nuclear-norm- and $l_1$–norm-involved convex relaxation problem:

$$\begin{array}{rl} \min_{A,E} & \|A\|_* + \tau\|E\|_1 \\ \textit{such that } (s.t.) & A + E = C, \end{array} \tag{1.1}$$

where $C \in \mathcal{R}^{m\times n}$ is the given matrix (data); the nuclear norm denoted by $\|\cdot\|_*$ is to induce the low-rank component of $C$ and the $l_1$ norm denoted by $\|\cdot\|_1$ is to induce the sparse component of $C$, and $\tau > 0$ is a constant balancing the low-rank and sparsity. This model has also been highlighted in [5] in the context of the so-called robust

*Received by the editors January 5, 2010; accepted for publication (in revised form) November 7, 2010; published electronically January 4, 2011.
http://www.siam.org/journals/siopt/21-1/78189.html

†Department of Mathematics, Nanjing University, Nanjing, China; School of Science, Nanjing University of Posts and Telecommunications, Nanjing, China (taom@njupt.edu.cn). This author was supported by the fund NY210049.

‡Department of Mathematics, Hong Kong Baptist University, Hong Kong, China (xmyuan@hkbu.edu.hk). This author was supported by the Hong Kong General Research fund 202610 and the NSFC grant 10701055.

principal component analysis (RPCA) where $C$ is a given high-dimensional matrix in $\mathcal{R}^{m\times n}$, $A$ is the underlying low-rank matrix representing the principal components, and $E$ is the corrupted data matrix which is sparse, yet its entries can be arbitrary in magnitude. As analyzed in [5], the model (1.1) is capable of recovering the principal components from grossly corrupted or outlying observation data, which is confronted in many applications such as the image processing and bioinformatics. Hence, it is a substantial improvement over the classical principal component analysis (PCA) which can only complete exact recovery from mildly (but could be densely) corrupted observation data, e.g., [33].

In [5], theoretical conditions to ensure the perfect recovery in the probabilistic sense via the convex programming (1.1) have been profoundly analyzed, with the default assumption that the dimensionality of data (i.e., $m$) is sufficiently high. This benefit brought by the high dimensionality was dubbed in [14] as the blessing of dimensionality. The curse accompanying the blessing of dimensionality, however, is that (1.1) becomes numerically challenging when the dimensionality is high. Although the semidefinite programming (SDP) reformulation of (1.1) is readily derived, the difficulty of high dimensionality excludes direct applications of some state-of-the-art yet generally purposed SDP solvers such as the popular interior–point-method packages SeDuMi [49] and SDPT3 [53]. This hardness, however, can be alleviated much by noticing the fact that (1.1) is well structured in the sense that the separable structure emerges in both the objective function and the constraint. Hence, we have no reason not to take advantage of this favorable structure for the benefits of algorithmic design. In fact, the thought of exploiting the favorable structure of (1.1), rather than treating it as a generic convex programming, has already inspired some interesting numerical algorithms for (1.1), e.g., the iterative thresholding algorithm [5], the accelerated proximal gradient algorithm in [40], and the alternating direction method proposed almost simultaneously and independently in [39, 59].

As pointed out in [5], in many real world applications, we need to consider the model (1.1) under more practical circumstances. First, only a fraction of entries of $C$ can be observed e.g., because of the experimental budget or reachability. We refer to the literature of the well-known matrix completion problem (e.g., [4, 6, 7, 8, 10, 34, 47]) for the justification of considering incomplete observation. In particular, let $\Omega$ be a subset of the index set of entries $\{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$. We assume that only those entries $\{C_{ij}, (i, j) \in \Omega\}$ can be observed. Note that it is reasonable to assume that $\Omega \supseteq \Gamma$, which is the support of the index set of nonzero entries of $E$. Adhering to the notation of [4, 6], we also summarize the incomplete observation information by the operator $P_\Omega : \mathcal{R}^{m\times n} \to \mathcal{R}^{m\times n}$, which is the orthogonal projection onto the span of matrices vanishing outside of $\Omega$ so that the $ij$-th entry of $P_\Omega(X)$ is $X_{ij}$ if $(i, j) \in \Omega$ and zero otherwise. Second, the observed data may be corrupted by both impulsive noise (sparse but large) and Gaussian noise (small but dense). Let $C$ be the superposition of the principal component matrix $A$, the impulsive noise matrix $E$, and the Gaussian noise matrix $F$; i.e., $C = A + E + F$. We assume that the Gaussian noise of the observed entries is small in the sense that $\|P_\Omega(F)\|_F \le \delta$, where $\delta > 0$ is the Gaussian noise level and $\|\cdot\|_F$ is the Frobenius norm. Then, to be broadly applicable, we are interested in the following concretely extended model of (1.1):

$$\begin{array}{rl} \min_{A,E} & \|A\|_* + \tau\|E\|_1 \\ s.t. & \|P_\Omega(C - A - E)\|_F \le \delta. \end{array} \tag{1.2}$$

Note that the model (1.2) includes many special and interesting cases, which capture diversified applications in many fields. It includes, for instance, the Gaussian-noisy

low-rank and sparse recovery with complete observation (or RPCA with Gaussian noise as in [5, 61]),

$$\begin{array}{rl} \min_{A,E} & \|A\|_* + \tau\|E\|_1 \\ s.t. & \|C - A - E\|_F \leq \delta, \end{array} \tag{1.3}$$

the Gaussian-noiseless low-rank and sparse recovery with incomplete observation (or RPCA without Gaussian noise as in [5]),

$$\begin{array}{rl} \min_{A,E} & \|A\|_* + \tau\|E\|_1 \\ s.t. & P_\Omega(A + E) = P_\Omega(C), \end{array} \tag{1.4}$$

and the Gaussian-noiseless low-rank and sparse recovery with complete observation (1.1). In particular, we emphasize that the model (1.2) recovers the convex relaxation problems of the noiseless (where $\tau = 0$, $E = 0$, and $\delta = 0$), the Gaussian-noisy (where $\tau = 0$, $E = 0$, and $\delta > 0$), and the grossly noisy (where $\delta = 0$) matrix completion problems, which have been well studied in the literature because of their wide applications in many fields; see, e.g., [5, 6, 7, 8, 19, 34, 35]. In fact, the profound work [7, 8] has inspired rapidly some efficient algorithms for solving the matrix completion problems, for example, the singular value thresholding algorithm [4], the fixed point continuation method [43], the refined interior-point method [42], the projected subgradient method [47], the approach based on the classical proximal point algorithm [41, 54], and the alternating direction method [10].

Despite the wide applicability and novelty, the model (1.2) is numerically challenging, also partially because of the high dimensionality. Just as (1.1), an easy reformulation of the constrained convex programming (1.2) falls perfectly in the applicable scope of the classical augmented Lagrangian method (ALM) which was contributed originally in [31, 46]; moreover, the favorable separable structure emerging in both the objective function and the constraints entails the idea of splitting the corresponding augmented Lagrangian function to derive more efficient numerical algorithms. As to be delineated later, we shall develop the alternating splitting augmented Lagrangian method (ASALM) and its variant (VASALM) for solving (1.2). Some preliminary numerical results show the fast and accurate recovery via implementing the proposing algorithms to solve (1.2). Thus, the validity of the model (1.2) for recovering low-rank and sparse components of a matrix from incomplete and noisy observation is verified empirically.

The rest of the paper is organized as follows. In section 2, we provide some preliminaries that are useful for the subsequent analysis. In section 3, we reform (1.2) into a more structured form, which inspires the augmented–Lagrangian-based methods to be developed. The optimality characterization of this reformulation is also derived for the convenience of subsequent analysis. In section 4, we apply directly the ALM to solve (1.2), without consideration of the special structure of (1.2). In section 5, we develop the general ALM by splitting the augmented Lagrangian function in the alternating order. Hence, the ASALM is derived for (1.2). In section 6, we present a variant of the ASALM, and prove its convergence. In section 7, we consider the possibility of extending the preceding augmented–Lagrangian-based methods to an alternative model of (1.2), i.e., the nuclear-norm- and $l_1$–norm-regularized least squares problems. In section 8, we report some numerical results of the proposing algorithms to verify the justification of the model (1.2) and the efficiency of the proposing algorithms. Finally, in section 9, we make some conclusions and discuss some topics for future work.

**2. Preliminaries.** In this section, we briefly review some well-known results in the literature that are useful for the subsequent analysis.

We first list some lemmas with respect to the shrinkage operators, which will be used at each iteration of the proposing augmented–Lagrangian-type methods to solve the generated subproblems.

LEMMA 2.1. *For $\mu > 0$ and $T \in \mathcal{R}^{m\times n}$, the solution of the problem*

$$\min_{S\in\mathcal{R}^{m\times n}} \mu\|S\|_1 + \frac{1}{2}\|S-T\|_F^2$$

*is given by $\mathcal{S}_\mu(T) \in \mathcal{R}^{m\times n}$, which is defined componentwisely by*

$$(S_\mu(T))_{ij} := \max\{abs(T_{ij}) - \mu, 0\} \cdot sign(T_{ij}), \tag{2.1}$$

*where $abs(\cdot)$ and $sign(\cdot)$ are the absolute value and sign functions, respectively.*

*Proof.* See, e.g., [3, 12, 57]. ∎

LEMMA 2.2. *Let $S_\mu$ be defined in (2.1), $Y \in \mathcal{R}^{m\times n}$ whose rank is $r$, and $\mu > 0$. The solution of the problem*

$$\arg\min_{X\in\mathcal{R}^{m\times n}} \left\{\mu\|X\|_* + \frac{1}{2}\|X-Y\|_F^2\right\}$$

*is given by $\mathcal{D}_\mu(Y) \in \mathcal{R}^{m\times n}$, which is defined by*

$$\mathcal{D}_\mu(Y) := U\,diag(\mathcal{S}_\mu(\Sigma))V^T, \tag{2.2}$$

*where $U \in \mathcal{R}^{m\times r}$, $V \in \mathcal{R}^{n\times r}$, and $\Sigma \in \mathcal{R}^{r\times r}$ are obtained by the singular value decomposition (SVD) of $Y$:*

$$Y = U\Sigma V^T, \qquad \text{and} \quad \Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_r).$$

*Proof.* See, e.g., [4, 32, 43]. ∎

**3. Reformulation and optimality.** In this section, we reform (1.2) into an augmented–Lagrangian-oriented form which is beneficial for the algorithmic design and convergence analysis later on, and derive the optimality characterization of this reformulation. More specifically, let $M := P_\Omega(C)$; then it is easy to see that (1.2) has the following reformulation:

$$\begin{array}{rl} \min_{A,E,Z} & \|A\|_* + \tau\|E\|_1 \\ s.t. & A + E + Z = M, \\ & Z \in \mathbf{B} := \{Z \in \mathcal{R}^{m\times n} |\ \|P_\Omega(Z)\|_F \le \delta\}. \end{array} \tag{3.1}$$

Let $\Lambda$ be the Lagrange multiplier associated with the linear constraint of (3.1) (or the variable of the dual problem of (3.1)). Then, it is obvious that the optimality condition of (3.1) can be characterized by some inclusions and variational inequalities (VI). More specifically, $(A^*, E^*, Z^*) \in \mathcal{R}^{m\times n} \times \mathcal{R}^{m\times n} \times \mathbf{B}$ is a solution of (3.1) if and only if there exists $\Lambda^* \in \mathcal{R}^{m\times n}$ that satisfies the following inclusions and VI:

$$\left\{\begin{array}{l} 0 \in \partial(\|A^*\|_*) - \Lambda^*, \\ 0 \in \tau\partial(\|E^*\|_1) - \Lambda^*, \\ \langle Z' - Z^*, -\Lambda^*\rangle \ge 0,\ \forall Z' \in \mathbf{B}, \\ A^* + E^* + Z^* - M = 0, \end{array}\right. \tag{3.2}$$

where $\partial(\cdot)$ denotes the subgradient operator of a convex function. Therefore, a solution of (3.2) also yields a solution of (3.1) (hence (1.2)). Throughout this paper, we assume that the solution set of (3.2), denoted by $\mathcal{W}^*$, is nonempty. Due to the monotonicity of the problem, $\mathcal{W}^*$ is convex (see Theorem 2.3.5 of [18]).

**4. The general augmented Lagrangian method.** In this section, we apply the classical ALM to solve (1.2), by treating it as a generic convex programming without consideration of its specific favorable structure.

The augmented Lagrangian function of (3.1) is

$$L_{\mathcal{A}}(A,E,Z,\Lambda,\beta) := \|A\|_*+\tau\|E\|_1-\langle\Lambda, A+E+Z-M\rangle+\frac{\beta}{2}\|A+E+Z-M\|_F^2, \tag{4.1}$$

where $\beta > 0$ is the penalty parameter for the violation of the linear constraint and $\langle\cdot\rangle$ denotes the standard trace inner product.

For $\rho > 1$ and $\beta_0 > 0$, with the given $\Lambda^k$, the classical ALM generates the new iterate $(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^{k+1})$ via the following computation.

**The $k$-th iteration of the ALM for (3.1):**
1. Compute

$$(A^{k+1}, E^{k+1}, Z^{k+1}) \in \arg\min_{A\in\mathcal{R}^{m\times n}, E\in\mathcal{R}^{m\times n}, Z\in\mathbf{B}} \mathcal{L}_{\mathcal{A}}(A,E,Z,\Lambda^k,\beta_k). \tag{4.2}$$

2. Update $\Lambda^{k+1}$ via $\Lambda^{k+1} = \Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M)$.
3. Update $\beta_{k+1}$ via $\beta_{k+1} = \rho \cdot \beta_k$.

Convergence of the proposed ALM for (1.2) is just a special scenario of that of the generic ALM, which can be found easily in [31, 46]. In particular, it is easy to prove that the sequence of the Lagrange multiplier $\{\Lambda^k\}$ is Fejèr monotone. Hence, the convergence of the ALM in terms of $\{\Lambda^k\}$ is implied immediately. In the following, we provide the proof for ensuring accurate recovery via the proposed ALM.

LEMMA 4.1. *The sequences $\{A^k\}, \{E^k\}, \{Z^k\}$, and $\{\Lambda^k\}$ generated by the proposed ALM are all bounded.*

*Proof.* It is easy to verify that the $k$-th iterate generated by the proposed ALM is characterized by the following system:

$$\begin{cases} 0 \in \partial(\|A^{k+1}\|_*) - [\Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M)], \\ 0 \in \partial(\tau\|E^{k+1}\|_1) - [\Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M)], \\ \langle Z' - Z^{k+1}, -[\Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M)]\rangle \geq 0, \quad \forall Z' \in \mathbf{B}, \\ \Lambda^{k+1} = \Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M). \end{cases} \tag{4.3}$$

Substituting the last equation of (4.3) into the first one, we get

$$\Lambda^{k+1} \in \partial(\|A^{k+1}\|_*),$$

which implies immediately that the sequence $\{\Lambda^k\}$ is bounded.

Note that the sequence of penalty parameter $\{\beta_k\}$ satisfies $\sum_{k=1}^{\infty} \beta_k^{-2}\beta_{k+1} < +\infty$. On the other hand, we have that

$$\begin{aligned} &\mathcal{L}_A(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^k, \beta_k) \leq \mathcal{L}_A(A^k, E^k, Z^k, \Lambda^k, \beta_k) \\ &= \mathcal{L}_A(A^k, E^k, Z^k, \Lambda^{k-1}, \beta_{k-1}) + \frac{1}{2}\beta_{k-1}^{-2}(\beta_{k-1} + \beta_k)\|\Lambda^k - \Lambda^{k-1}\|_F^2. \end{aligned}$$

Since

$$\sum_{k=1}^{\infty} \beta_{k-1}^{-2}(\beta_{k-1} + \beta_k) \leq 2\sum_{k=1}^{\infty} \beta_{k-1}^{-2}\beta_k < +\infty,$$

and recall the boundedness of $\{\Lambda^k\}$, we have that $\{\mathcal{L}_A(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^k, \beta^k)\}$ is upper bounded. Note that $\Lambda^k = \Lambda^{k-1} - \beta_{k-1}(A^k + E^k + Z^k - M)$. We then have

$$\|A^k\|_* + \tau\|E^k\|_1 = \mathcal{L}_A(A^k, E^k, Z^k, \Lambda^{k-1}, \beta_{k-1}) - \frac{1}{2\beta_{k-1}}(\|\Lambda^k\|_F^2 - \|\Lambda^{k-1}\|_F^2),$$

which is also upper bounded. Therefore, both $\{A^k\}$ and $\{E^k\}$ are bounded.

Moreover, the VI in (4.3) implies that (see, e.g., [18])

$$Z^{k+1} = P_{\mathbf{B}}[\frac{1}{\beta_k}\Lambda^k - (A^{k+1} + E^{k+1} - M)].$$

Thus, the boundedness of $\{Z^k\}$ follows immediately from the boundedness of $\{A^k\}$, $\{E^k\}$, $\{\Lambda^k\}$, and the definition of projection on $\mathbf{B}$ (see (3.1)). □

Now we are ready to present the convergence theorem of the proposed ALM.

THEOREM 4.2. *Let the sequence* $\{(A^k, E^k, Z^k, \Lambda^k)\}$ *be generated by the ALM for* (3.1). *Then, any accumulation point* $(A^*, E^*, Z^*, \Lambda^*)$ *of the sequence generated by the proposed ALM yields an optimal solution of* (3.1).

*Proof.* By an easy reformulation, we find that (4.3) implies that

$$\begin{cases} 0 \in \partial(\|A^{k+1}\|_*) - \Lambda^{k+1}, \\ 0 \in \partial(\tau\|E^{k+1}\|_1) - \Lambda^{k+1}, \\ \langle Z' - Z^{k+1}, -\Lambda^{k+1}\rangle \geq 0, \quad \forall Z' \in \mathbf{B}, \\ A^{k+1} + E^{k+1} + Z^{k+1} - M = \frac{1}{\beta_k}(\Lambda^k - \Lambda^{k+1}). \end{cases} \tag{4.4}$$

Recall the boundedness of $\{(A^k, E^k, Z^k, \Lambda^k)\}$ proved in Lemma 4.1. There exists an accumulation point of $\{(A^k, E^k, Z^k, \Lambda^k)\}$, and we denote it by $(A^*, E^*, Z^*, \Lambda^*)$. According to the last equation in (4.4) and the facts that $\{\Lambda^k\}$ is bounded and $\lim_{k\to\infty} \beta_k = +\infty$, we have

$$\lim_{k\to\infty} A^{k+1} + E^{k+1} + Z^{k+1} - M = 0. \tag{4.5}$$

Then, we can easily conclude that $(A^*, E^*, Z^*, \Lambda^*)$ satisfies the optimality characterization (3.2). Hence, the proof is completed. □

**5. The alternating splitting augmented Lagrangian method.** As we emphasized previously, the direct application of the ALM to the well-structured problem (3.1) treats it as a generic convex programming and ignores completely the favorable structure which could be very beneficial for designing efficient algorithms. In fact, the separable structure emerging in both the objective function and constraints in (3.1) enables us to derive more structure-exploited augmented–Lagrangian-based algorithms by splitting the augmented Lagrangian function in appropriate ways. In this and following sections, we focus on the way of splitting the augmented Lagrangian function in the spirit of the well-known alternating direction method (ADM). Hence, the minimization task of (4.2) is decomposed into three smaller ones which solve the variables $A^{k+1}$, $E^{k+1}$, and $Z^{k+1}$ separably in the consecutive order. With this splitting, the ASALM and its variant VASALM for solving (3.1) will be developed.

Recall that the ADM dates back to [24, 25, 26, 27] and is closely related to the Douglas–Rachford operator splitting method [13], and that it has attracted wide attention of many authors in various areas; see, e.g., [2, 11, 15, 16, 23, 28, 30, 36, 51, 52].

In particular, some novel and attractive applications of the ADM have been discovered very recently, e.g., the total-variation problem in image processing [17, 44, 51, 58], the covariance selection problem and semidefinite least square problem in statistics [29, 59], the SDP [50, 56], and the sparse and low-rank recovery problem in engineering [39, 60].

**5.1. Algorithm.** More specifically, let $\mathcal{L}_{\mathcal{A}}$ be defined in (4.1) and $\mathbf{B}$ be defined in (3.1); let $\beta > 0$. Then, with the given $(A^k, E^k, \Lambda^k)$, the ASALM generates the new iterate $(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^{k+1})$ via the following scheme:

$$\left\{\begin{array}{l} Z^{k+1} \in \arg\min_{Z\in\mathbf{B}} \mathcal{L}_{\mathcal{A}}(A^k, E^k, Z, \Lambda^k, \beta), \\ E^{k+1} \in \arg\min_{E\in\mathcal{R}^{m\times n}} \mathcal{L}_{\mathcal{A}}(A^k, E, Z^{k+1}, \Lambda^k, \beta), \\ A^{k+1} \in \arg\min_{A\in\mathcal{R}^{m\times n}} \mathcal{L}_{\mathcal{A}}(A, E^{k+1}, Z^{k+1}, \Lambda^k, \beta), \\ \Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M), \end{array}\right. \tag{5.1}$$

which can be easily written into the following more specific form:

$$\left\{\begin{array}{l} Z^{k+1} = \arg\min_{Z\in\mathbf{B}} \frac{\beta}{2}\|Z + A^k + E^k - \frac{1}{\beta}\Lambda^k - M\|_F^2, \\ E^{k+1} = \arg\min_{E\in\mathcal{R}^{m\times n}} \tau\|E\|_1 + \frac{\beta}{2}\|E + A^k + Z^{k+1} - \frac{1}{\beta}\Lambda^k - M\|_F^2, \\ A^{k+1} = \arg\min_{A\in\mathcal{R}^{m\times n}} \|A\|_* + \frac{\beta}{2}\|A + E^{k+1} + Z^{k+1} - \frac{1}{\beta}\Lambda^k - M\|_F^2, \\ \Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M). \end{array}\right. \tag{5.2}$$

As shown in (5.1) and (5.2), like the ADM, the ASALM decomposes the minimization task in (4.2) into three separable tasks. Moreover, the involved subproblems are solved in the consecutive order so that the latest iterative information is adopted whenever possible: the second subproblem uses the solution of the first subproblem and the third subproblem takes advantage of solutions of the first two subproblems in order to approximate the original ALM as much as possible. In this sense, the iterative scheme of the ASALM is in the spirit of the Gauss–Seidel type methods, and we believe that this fact contributes much to the promising numerical results of the ASALM to be reported.

The main fact making the proposing ASALM easily implementable is that all the generated subproblems in (5.2) have closed-form solutions. Now, we elaborate on the strategies of solving these subproblems at each iteration.

Recall the definition of $\mathbf{B}$ in (3.1). Then, it is easy to verify that the solution of the first subproblem involving $Z^{k+1}$ can be solved explicitly via

$$Z_{ij}^{k+1} = \left\{\begin{array}{ll} N_{ij}^k, & \text{if } (i,j) \notin \Omega; \\ \frac{\min\{\|P_\Omega(N^k)\|_F, \delta\}}{\|P_\Omega(N^k)\|_F} N_{ij}^k, & \text{if } (i,j) \in \Omega; \end{array}\right., \tag{5.3}$$

where $N^k = \frac{1}{\beta}\Lambda^k + M - A^k - E^k$. Note that the computation demanded for this subproblem is $O(mn)$.

For the second subproblem involving $E^{k+1}$ in (5.2), according to Lemma 2.1, we have that

$$E^{k+1} = \mathcal{S}_{\tau/\beta}\left(\frac{1}{\beta}\Lambda^k + M - A^k - Z^{k+1}\right),$$

where the shrinkage operator $S_{\tau/\beta}$ is defined in (2.1). Note that this subproblem also requires $O(mn)$ flops.

Last, according to Lemma 2.2, the third subproblem involving $A^{k+1}$ in (5.2) can also be solved explicitly via the following scheme:

$$A^{k+1} = \mathcal{D}_{1/\beta}(\frac{1}{\beta}\Lambda^k + M - Z^{k+1} - E^{k+1}),$$

where the nuclear–norm-involved shrinkage operator $\mathcal{D}_{1/\beta}$ is defined in (2.2). Note that one SVD is required by this subproblem, and this SVD actually dominates the main computation of each iteration of the proposed ASALM for solving (1.2).

Now, we are ready to present the algorithm of the ASALM for (3.1). Let $\beta > 0$ and $(A^k, E^k, \Lambda^k)$ be given. Then, the ASALM generates the new iterate via the following computation.

**The $k$-th iteration of the ASALM for (3.1):**
1. Compute $Z^{k+1}$ via

$$Z_{ij}^{k+1} = \begin{cases} N_{ij}^k, & \text{if } (i,j) \notin \Omega; \\ \frac{\min\{\|P_\Omega(N^k)\|_F, \delta\}}{\|P_\Omega(N^k)\|_F} N_{ij}^k, & \text{if } (i,j) \in \Omega \end{cases}$$

with $N^k = \frac{1}{\beta}\Lambda^k + M - A^k - E^k$.
2. Compute $E^{k+1}$ via $E^{k+1} = \mathcal{S}_{\tau/\beta}(\frac{1}{\beta}\Lambda^k + M - A^k - Z^{k+1})$.
3. Compute $A^{k+1}$ via $A^{k+1} = \mathcal{D}_{1/\beta}(\frac{1}{\beta}\Lambda^k + M - Z^{k+1} - E^{k+1})$.
4. Update $\Lambda^{k+1}$ via $\Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M)$.

*Remark.* The reason we decide to perform the alternating tasks in the order of $Z^k \to E^k \to A^k$ is that we actually allow partial SVD in the computation of $A^k$, as to be delineated in section 8. Hence, to prevent the error resulted by the partial SVD from affecting subsequent subproblems, we do not follow the conventional alternating order: $A^k \to E^k \to Z^k$. If full SVD is executed in the $A^k$-involved subproblem, then the numerical performance of the ASALM makes little difference for different alternating orders among the variables $\{A^k, E^k, Z^k\}$.

**5.2. Stopping criterion.** It is easy to verify that the iterate $(A^k, E^k, Z^k, \Lambda^k)$ generated by the proposed ASALM can be characterized by

$$\begin{cases} \langle Z - Z^{k+1}, -[\Lambda^k - \beta(A^k + E^k + Z^{k+1} - M)]\rangle \geq 0, \ \forall Z \in \mathbf{B}, \\ 0 \in \partial(\tau\|E^{k+1}\|_1) - [\Lambda^k - \beta(A^k + E^{k+1} + Z^{k+1} - M)], \\ 0 \in \partial\|A^{k+1}\|_* - [\Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M)], \\ \Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M), \end{cases} \tag{5.4}$$

which is equivalent to

$$\begin{cases} \langle Z - Z^{k+1}, -\Lambda^{k+1} + \beta(A^k - A^{k+1}) + \beta(E^k - E^{k+1})\rangle \geq 0, \ \forall Z \in \mathbf{B}, \\ 0 \in \partial(\tau\|E^{k+1}\|_1) - \Lambda^{k+1} + \beta(A^k - A^{k+1}), \\ 0 \in \partial(\|A^{k+1}\|_*) - \Lambda^{k+1}, \\ \Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M). \end{cases} \tag{5.5}$$

Recall the optimality condition (3.2). The characterization (5.5) immediately shows that the distance of the iterate $(A^{k+1}, E^{k+1}, Z^{k+1})$ to the solution set of (3.1) can be measured by the quantities $\beta(\|A^k - A^{k+1}\| + \|E^k - E^{k+1}\|)$ and $\frac{1}{\beta}\|\Lambda^k - \Lambda^{k+1}\|$. This inspires an easily implementable stopping criterion for implementing the proposed ASALM:

$$\min\{\beta(\|A^k - A^{k+1}\| + \|E^k - E^{k+1}\|), \frac{1}{\beta}\|\Lambda^k - \Lambda^{k+1}\|\} \leq \varepsilon. \tag{5.6}$$

**5.3. Toward convergence.** In this subsection, we first prove the boundedness of the iterate generated by the ASALM. For notational convenience, we define

$$\hat{\Lambda}^{k+1} := \Lambda^k - \beta_k(E^{k+1} + A^k + Z^{k+1} - M).$$

LEMMA 5.1. *Let the sequences $\{A^k\}, \{E^k\}, \{Z^k\}, \{\Lambda^k\}$ be generated by the proposed ASALM. Then, in the implementation of the proposed ASALM, if the penalty parameter $\beta$ is allowed to vary dynamically and the sequence $\{\beta_k\}$ is chosen appropriately such that $\sum_{k=1}^{\infty} \beta_k^{-2}\beta_{k+1} < +\infty$, then the sequences $\{A^k\}, \{E^k\}, \{Z^k\}, \{\Lambda^k\}, \{\hat{\Lambda}^k\}$ generated by the proposed ASALM are all bounded.*

*Proof.* We first reiterate that the penalty parameter $\beta$ is set to be fixed in the presentation of the algorithm of ASALM, for the sake of simplification. In fact, as used in the proposed ALM (and also [39]), we can adjust its value dynamically at each iteration by the principle $\beta_{k+1} := \rho \cdot \beta_k$, where $\rho \in (1, +\infty)$. It is easy to verify that this strategy of determining the penalty parameter satisfies the requirement on $\{\beta_k\}$ assumed in this lemma. Now, we start the proof, which is analogous to Lemma 4.1. Recall (5.4). We have

$$0 \in \tau\partial(\|E^{k+1}\|_1) - [\Lambda^k - \beta_k(E^{k+1} + A^k + Z^{k+1} - M)],$$

and

$$0 \in \partial(\|A^{k+1}\|_*) - [\Lambda^k - \beta_k(A^{k+1} + E^{k+1} + Z^{k+1} - M)],$$

which equivalently states that

$$\hat{\Lambda}^{k+1} \in \partial(\tau\|E^{k+1}\|_1), \quad \Lambda^{k+1} \in \partial(\|A^{k+1}\|_*).$$

From above facts, the sequences $\{\Lambda^k\}$ and $\{\hat{\Lambda}^k\}$ are bounded since the dual norms of $\|\cdot\|_*$ and $\|\cdot\|_1$ are $\|\cdot\|_2$ and $\|\cdot\|_\infty$, respectively.

It is assumed that $\{\beta_k\}$ satisfies $\sum_{k=1}^{\infty} \beta_k^{-2}\beta_{k+1} < +\infty$. In addition, we have that

$$\begin{aligned}
&\mathcal{L}_A(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^k, \beta_k) \leq \mathcal{L}_A(A^k, E^{k+1}, Z^{k+1}, \Lambda^k, \beta_k) \\
&\leq \mathcal{L}_A(A^k, E^{k+1}, Z^k, \Lambda^k, \beta_k) \leq \mathcal{L}_A(A^k, E^k, Z^k, \Lambda^k, \beta_k) \\
&= \mathcal{L}_A(A^k, E^k, Z^k, \Lambda^{k-1}, \beta_{k-1}) + \frac{1}{2}\beta_{k-1}^{-2}(\beta_{k-1} + \beta_k)\|\Lambda^k - \Lambda^{k-1}\|_F^2.
\end{aligned}$$

Since

$$\sum_{k=1}^{\infty} \beta_{k-1}^{-2}(\beta_{k-1} + \beta_k) \leq 2\sum_{k=1}^{\infty} \beta_{k-1}^{-2}\beta_k < +\infty,$$

and recall the boundedness of $\{\Lambda^k\}$, we have that $\{\mathcal{L}_A(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^k, \beta^k)\}$ is upper bounded. Recall that $\Lambda^k = \Lambda^{k-1} - \beta_{k-1}(A^k + E^k + Z^k - M)$. We then have

$$\|A^k\|_* + \tau\|E^k\|_1 = \mathcal{L}_A(A^k, E^k, Z^k, \Lambda^{k-1}, \beta_{k-1}) - \frac{1}{2\beta_{k-1}}(\|\Lambda^k\|_F^2 - \|\Lambda^{k-1}\|_F^2),$$

which is also upper bounded. Therefore, both $\{A^k\}$ and $\{E^k\}$ are bounded.

Moreover, recall (5.3). Then, the boundedness of $\{Z^k\}$ follows immediately from the boundedness of $\{A^k\}$, $\{E^k\}$, and $\{\Lambda^k\}$. □

Without more conditions, we must point out that the convergence of the proposed ASALM is ambiguous. In fact, although the convergence of the ADM for linearly constrained convex programming whose objective function is separable into two parts is well known, the convergence of the ADM for the more general case with three or more separable parts is still open in the literature. Nevertheless, for the proposed ASALM, when some restrictive conditions on $\{\beta_k\}$ are assumed, e.g., the condition proposed in [39] (i.e., $\lim_{k\to\infty} \beta_k(A^{k+1} - A^k) = 0$ and $\lim_{k\to\infty} \beta_k(E^{k+1} - E^k) = 0$), it is easy to derive the convergence of the ASALM. For succinctness, we omit the proof and refer to [39].

**6. A variant alternating splitting augmented Lagrangian method.** As we emphasized in the last section, convergence of the proposed ASALM can be proved theoretically under some restrictive condition on $\{\beta_k\}$, while the same convergence for the ASALM under mild assumptions on $\{\beta_k\}$ is still ambiguous. Hence, although empirically the numerical performance of the ASALM is shown to be dominatingly better (as to be reported), we still desire to seek some variants of the ASALM whose convergence is ensured without restrictive requirements on the penalty parameter. This desire inspires us to propose the following variant of the ASALM (denoted by VASALM).

**6.1. Algorithm.** More specifically, let $\mathbf{B}$ be defined in (3.1). Let $\beta > 0$, $\eta > 2$, and $(A^k, E^k, \Lambda^k)$ be given. We propose the VASALM which generates the new iterate $(A^{k+1}, E^{k+1}, Z^{k+1}, \Lambda^{k+1})$ via the following scheme:

$$\text{(6.1)} \qquad \begin{cases} Z^{k+1} \in \arg\min_{Z\in\mathbf{B}} \mathcal{L}_{\mathcal{A}}(A^k, E^k, Z, \Lambda^k, \beta), \\ \tilde{\Lambda}^k = \Lambda^k - \beta(A^k + E^k + Z^{k+1} - M), \\ E^{k+1} = \arg\min_{E\in\mathcal{R}^{m\times n}} \tau\|E\|_1 + \frac{\beta\eta}{2}\|E - E^k - \frac{1}{\beta\eta}\tilde{\Lambda}^k\|_F^2, \\ A^{k+1} = \arg\min_{A\in\mathcal{R}^{m\times n}} \|A\|_* + \frac{\beta\eta}{2}\|A - A^k - \frac{1}{\beta\eta}\tilde{\Lambda}^k\|_F^2, \\ \Lambda^{k+1} = \tilde{\Lambda}^k + \beta(E^k - E^{k+1}) + \beta(A^k - A^{k+1}). \end{cases}$$

Analogously, the generated subproblems of the VASALM are all easy to handle in the sense that they all have closed-form solutions. In fact, it is easy to verify that each iteration of the VASALM has the same computational complexity as that of the proposed ASALM. Thus, the advantages of the ASALM for easy implementation are preserved completely by the VASALM. For succinctness, we omit the elaboration on solving the subproblems of the VASALM, which is analogous to that in section 6.1. Instead, we directly present the algorithm of the VASALM for solving (3.1).

Let $\eta > 2$, $\beta > 0$, and $(A^k, E^k, \Lambda^k)$ be given. Then, the VASALM generates the new iterate $(Z^{k+1}, E^{k+1}, A^{k+1}, \Lambda^{k+1})$ via the following computation.

**The $k$-th iteration of the VASALM for (3.1):**

1. Compute $Z^{k+1}$ via

$$Z_{ij}^{k+1} = \begin{cases} N_{ij}^k, & \text{if } (i,j) \notin \Omega; \\ \frac{\min\{\|P_\Omega(N^k)\|_F, \delta\}}{\|P_\Omega(N^k)\|_F} N_{ij}^k, & \text{if } (i,j) \in \Omega \end{cases}$$

with $N^k = \frac{1}{\beta}\Lambda^k + M - A^k - E^k$.

2. Compute

$$\tilde{\Lambda}^k := \Lambda^k - \beta(A^k + E^k + Z^{k+1} - M). \tag{6.2}$$

3. Compute $E^{k+1}$ via $E^{k+1} = \mathcal{S}_{\tau/\beta\eta}(E^k + \frac{1}{\beta\eta}\tilde{\Lambda}^k)$.

4. Compute $A^{k+1}$ via $A^{k+1} = \mathcal{D}_{1/\beta\eta}(A^k + \frac{1}{\beta\eta}\tilde{\Lambda}^k)$.

5. Update $\Lambda^{k+1}$ via

$$\Lambda^{k+1} = \tilde{\Lambda}^k + \beta(E^k - E^{k+1}) + \beta(A^k - A^{k+1}). \tag{6.3}$$

*Remark.* Note that at each iteration of the VASALM, the subproblems involving $E^{k+1}$ and $A^{k+1}$ both need the solution of $Z^{k+1}$. Thus, in the implementation of the VASALM, the $Z^{k+1}$-related subproblem should precede the $E^{k+1}$- and $A^{k+1}$-related subproblems. This is the alternating characterization of the VASALM. On the other hand, the $E^{k+1}$- and $A^{k+1}$-related subproblems are eligible for parallel computation since they do not require the solution of the other subproblem in their own procedures of solution, and this is the parallel characterization of the VASALM. In these senses, the proposed VASALM is featured by the partially alternating and parallel fashion.

**6.2. Convergence.** In this subsection, we concentrate on the convergence of the proposed VASALM. We first prove some contractive properties of the sequence generated by the proposed VASALM, which play crucial roles in the coming convergence analysis. For the iterate $(Z^{k+1}, E^{k+1}, A^{k+1}, \Lambda^{k+1})$ generated by the VASALM from the given iterate $(Z^k, E^k, A^k, \Lambda^k)$, we also temporarily denote $\tilde{E}^k := E^{k+1}$ and $\tilde{A}^k := A^{k+1}$.

For convenience, we use the notations

$$W = \begin{pmatrix} Z \\ E \\ A \\ \Lambda \end{pmatrix}, \quad W^* = \begin{pmatrix} Z^* \\ E^* \\ A^* \\ \Lambda^* \end{pmatrix}, \quad V = \begin{pmatrix} E \\ A \\ \Lambda \end{pmatrix}, \quad \text{and} \quad V^* = \begin{pmatrix} E^* \\ A^* \\ \Lambda^* \end{pmatrix}.$$

For any positive integer $i$, we also use the notations

$$W^i = \begin{pmatrix} Z^i \\ E^i \\ A^i \\ \Lambda^i \end{pmatrix}, \quad V^i = \begin{pmatrix} E^i \\ A^i \\ \Lambda^i \end{pmatrix}, \quad \text{and} \quad \tilde{V}^i = \begin{pmatrix} \tilde{E}^i \\ \tilde{A}^i \\ \tilde{\Lambda}^i \end{pmatrix}.$$

Moreover, we define $\mathcal{V}^* := \{V^* | W^* \in \mathcal{W}^*\}$. Thus, under the blanket assumption that $\mathcal{W}^*$ is nonempty, we have that $\mathcal{V}^*$ is also nonempty.

LEMMA 6.1. *Let $I_m$ denote the identity matrix in $\mathcal{R}^{m\times m}$. Let $V^k$ and $\tilde{V}^k$ be defined as before. Then, we have the following identity:*

$$\langle V^k - \tilde{V}^k, G \cdot d(V^k, \tilde{V}^k)\rangle = \langle V^k - \tilde{V}^k, H \cdot (V^k - \tilde{V}^k)\rangle, \tag{6.4}$$

*where*

$$G = \begin{pmatrix} \eta\beta I_m & 0 & 0 \\ 0 & \eta\beta I_m & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}, \quad H = \begin{pmatrix} \eta\beta I_m & 0 & -\frac{1}{2} I_m \\ 0 & \eta\beta I_m & -\frac{1}{2} I_m \\ -\frac{1}{2} I_m & -\frac{1}{2} I_m & \frac{1}{\beta} I_m \end{pmatrix}, \tag{6.5}$$

*and*

$$d(V^k, \tilde{V}^k) = \begin{pmatrix} E^k - \tilde{E}^k \\ A^k - \tilde{A}^k \\ \Lambda^k - \tilde{\Lambda}^k - \beta(E^k - \tilde{E}^k) - \beta(A^k - \tilde{A}^k) \end{pmatrix}. \tag{6.6}$$

*Proof.* Elementary, thus omitted. □

*Remark.* With the notation of $d(V^k, \tilde{V}^k)$, we can easily see that the iterative scheme of the proposed VASALM is equivalent to the compact form, $V^{k+1} = V^k - d(V^k, \tilde{V}^k)$, which formally reveals that the VASALM can be explained as a descent type method where $d(V^k, \tilde{V}^k)$ acts the role of a descent direction, as further justified in the following lemma.

LEMMA 6.2. *Let $V^k$, $\tilde{V}^k$, $d(V^k, \tilde{V}^k)$, $G$, and $H$ be defined as before. Let $V^* \in \mathcal{V}^*$. Then, we have*

$$\langle V^k - V^*, G \cdot d(V^k, \tilde{V}^k) \rangle \geq \langle V^k - \tilde{V}^k, H \cdot (V^k - \tilde{V}^k) \rangle. \tag{6.7}$$

*Proof.* First, it follows from (6.2) and (6.3) that

$$Z^{k+1} + \tilde{E}^k + \tilde{A}^k - M = \frac{1}{\beta}(\Lambda^k - \tilde{\Lambda}^k) - (E^k - \tilde{E}^k) - (A^k - \tilde{A}^k). \tag{6.8}$$

Then, based on the optimality condition of (6.1), we have

$$\begin{cases} \langle Z' - Z^{k+1}, -\tilde{\Lambda}^k \rangle \geq 0, & \forall\, Z' \in \mathbf{B}; \\ \langle E' - \tilde{E}^k, \tau G_1 - \tilde{\Lambda}^k + \eta\beta(\tilde{E}^k - E^k) \rangle \geq 0, & \forall\, E' \in \mathcal{R}^{m\times n}; \\ \langle A' - \tilde{A}^k, G_2 - \tilde{\Lambda}^k + \eta\beta(\tilde{A}^k - A^k) \rangle \geq 0, & \forall\, A' \in \mathcal{R}^{m\times n}; \\ Z^{k+1} + \tilde{E}^k + \tilde{A}^k - M - [\frac{1}{\beta}(\Lambda^k - \tilde{\Lambda}^k) - (E^k - \tilde{E}^k) - (A^k - \tilde{A}^k)] = 0, \end{cases} \tag{6.9}$$

where $G_1 \in \partial\|\tilde{E}^k\|_1$ and $G_2 \in \partial\|\tilde{A}^k\|_*$.

On the other hand, based on the optimality condition of (3.2), we have

$$\begin{cases} \langle Z^{k+1} - Z^*, -\Lambda^* \rangle \geq 0, \\ \langle \tilde{E}^k - E^*, \tau S_1 - \Lambda^* \rangle = 0, \\ \langle \tilde{A}^k - A^*, S_2 - \Lambda^* \rangle = 0, \\ Z^* + E^* + A^* - M = 0, \end{cases} \tag{6.10}$$

where $S_1 \in \partial\|E^*\|_1$ and $S_2 \in \partial\|A^*\|_*$.

Setting $Z' = Z^*$, $E' = E^*$, and $A' = A^*$ in (6.9) and adding them into (6.10), we obtain

$$\begin{cases} \langle Z^* - Z^{k+1}, -(\tilde{\Lambda}^k - \Lambda^*) \rangle \geq 0, \\ \langle E^* - \tilde{E}^k, \tau(G_1 - S_1) - (\tilde{\Lambda}^k - \Lambda^*) + \eta\beta(\tilde{E}^k - E^k) \rangle \geq 0, \\ \langle A^* - \tilde{A}^k, (G_2 - S_2) - (\tilde{\Lambda}^k - \Lambda^*) + \eta\beta(\tilde{A}^k - A^k) \rangle \geq 0, \\ (Z^{k+1} + \tilde{E}^k + \tilde{A}^k - M) - [\frac{1}{\beta}(\Lambda^k - \tilde{\Lambda}^k) - (E^k - \tilde{E}^k) - (A^k - \tilde{A}^k)] = 0. \end{cases} \tag{6.11}$$

Note that the operator of the subgradient of a convex function is monotone. Hence, we have

$$\langle \tilde{E}^k - E^*, G_1 - S_1 \rangle \geq 0, \quad \langle \tilde{A}^k - A^*, G_2 - S_2 \rangle \geq 0. \tag{6.12}$$

In addition, recall the fact that $Z^* + E^* + A^* - M = 0$. We have the following identity:

$$\begin{aligned} &\langle Z^{k+1} - Z^*, -(\tilde{\Lambda}^k - \Lambda^*) \rangle + \langle \tilde{E}^k - E^*, -(\tilde{\Lambda}^k - \Lambda^*) \rangle + \langle \tilde{A}^k - A^*, -(\tilde{\Lambda}^k - \Lambda^*) \rangle \\ &\qquad + \langle \tilde{\Lambda}^k - \Lambda^*, Z^{k+1} + \tilde{E}^k + \tilde{A}^k - M \rangle = 0. \end{aligned} \tag{6.13}$$

Thus, according to (6.11)–(6.13), the definition of $G$, and $d(V^k, \tilde{V}^k)$ (see (6.5) and (6.6)), we obtain

$$\langle \tilde{V}^k - V^*, G d(V^k, \tilde{V}^k) \rangle \geq 0.$$

Therefore, it follows from Lemma 6.1 that

$$\begin{aligned} \langle V^k - V^*, G d(V^k, \tilde{V}^k) \rangle &\geq \langle V^k - \tilde{V}^k, G d(V^k, \tilde{V}^k) \rangle \\ &= \langle V^k - \tilde{V}^k, H \cdot (V^k - \tilde{V}^k) \rangle, \end{aligned}$$

which is the assertion of this lemma. □

The following theorem indicates that the sequence generated by the proposed VASALM is Fejèr monotone with respect to $\mathcal{V}^*$, and hence implies the convergence.

THEOREM 6.3. *Let $G$ be defined in* (6.5) *and $V^* \in \mathcal{V}^*$ and the sequence $\{V^k\}$ be generated by the proposed VASALM. Then, the sequence $\{V^k\}$ satisfies*

$$\|V^{k+1} - V^*\|_G^2 \leq \|V^k - V^*\|_G^2 - \left\{ (\eta - 2)\beta \big[ \|E^k - \tilde{E}^k\|_F^2 + \|A^k - \tilde{A}^k\|_F^2 \big] + \frac{1}{\beta} \|\Lambda^k - \tilde{\Lambda}^k\|_F^2 \right\}, \tag{6.14}$$

*where*

$$\|V^k - V^*\|_G^2 := \eta\beta(\|E^k - E^*\|_F^2 + \|A^k - A^*\|_F^2) + \frac{1}{\beta}\|\Lambda^k - \Lambda^*\|_F^2.$$

*Proof.* Recall that the new iterate $(V^{k+1})^T = ((E^{k+1})^T, (A^{k+1})^T, (\Lambda^{k+1})^T)$ can be expressed as

$$V^{k+1} = V^k - d(V^k, \tilde{V}^k).$$

Due to (6.5), (6.6), (6.7), and the fact

$$2\|E^k - \tilde{E}^k\|_F^2 + 2\|A^k - \tilde{A}^k\|_F^2 \geq \|(E^k - \tilde{E}^k) + (A^k - \tilde{A}^k)\|_F^2,$$

we can easily derive that

$$\begin{aligned} \|V^{k+1} - V^*\|_G^2 &= \|V^k - d(V^k, \tilde{V}^k) - V^*\|_G^2 \\ &= \|V^k - V^*\|_G^2 - 2\langle V^k - V^*, G d(V^k, \tilde{V}^k) \rangle + \|d(V^k, \tilde{V}^k)\|_G^2 \\ &\leq \|V^k - V^*\|_G^2 - \Big\{ 2\langle V^k - \tilde{V}^k, H \cdot (V^k - \tilde{V}^k) \rangle \\ &\quad - \eta\beta\|E^k - \tilde{E}^k\|_F^2 - \eta\beta\|A^k - \tilde{A}^k\|_F^2 - \frac{1}{\beta}\|(\Lambda^k - \tilde{\Lambda}^k) - \beta(E^k - \tilde{E}^k) - \beta(A^k - \tilde{A}^k)\|_F^2 \Big\} \\ &\leq \|V^k - V^*\|_G^2 - \big\{ (\eta - 2)\beta\big( \|E^k - \tilde{E}^k\|_F^2 + \|A^k - \tilde{A}^k\|_F^2 \big) + \frac{1}{\beta}\|\Lambda^k - \tilde{\Lambda}^k\|_F^2 \big\}, \end{aligned}$$

which proves the assertion of this theorem. ∎

Based on the above theorem, we have the following corollary immediately, which paves the way towards the convergence of the VASALM.

COROLLARY 6.4. *Let the sequence* $\{V^k\}$ *be generated by the proposed VASALM. Then, we have the following.*

1. *The sequence* $\{V^k\}$ *is bounded.*
2. $\lim_{k\to\infty}\{\|E^k-\tilde{E}^k\|_F^2+\|A^k-\tilde{A}^k\|_F^2+\|\Lambda^k-\tilde{\Lambda}^k\|_F^2\}=0$.

*Proof.* The first assertion follows from (6.14) directly. We now prove the second assertion. Recall that $\eta>2$ and $\beta>0$ and from (6.14), we easily have

$$\sum_{k=0}^{\infty}\left\{(\eta-2)\beta(\|E^k-\tilde{E}^k\|_F^2+\|A^k-\tilde{A}^k\|_F^2)+\frac{1}{\beta}\|\Lambda^k-\tilde{\Lambda}^k\|_F^2\right\}\le\|V^0-V^*\|_G^2<+\infty,$$

which immediately implies that

$$\lim_{k\to\infty}\|E^k-\tilde{E}^k\|_F=0,\quad \lim_{k\to\infty}\|A^k-\tilde{A}^k\|_F=0,\quad \lim_{k\to\infty}\|\Lambda^k-\tilde{\Lambda}^k\|_F=0, \tag{6.15}$$

i.e., the second assertion. ∎

We are now ready to prove the convergence of the proposed VASALM.

THEOREM 6.5. *Let* $\{V^k\}$ *and* $\{W^k\}$ *be the sequences generated by the proposed VASALM. Then, we have the following.*

1. *Any cluster point of* $\{W^k\}$ *is a solution point of* (3.2).
2. *The sequence* $\{V^k\}$ *converges to some* $V^\infty\in\mathcal{V}^*$.

*Proof.* Because of the assertion (6.15), it follows from (6.9) that

$$\begin{cases}\lim_{k\to\infty}\langle Z'-Z^{k+1},-\tilde{\Lambda}^k\rangle\ge 0, & \forall Z'\in\mathbf{B};\\ \lim_{k\to\infty}\langle E'-\tilde{E}^k,(\tau G_1-\tilde{\Lambda}^k)\rangle\ge 0, & \forall\, G_1\in\partial\|\tilde{E}^k\|_1,\ \forall E'\in\mathcal{R}^{m\times n};\\ \lim_{k\to\infty}\langle A'-\tilde{A}^k,(G_2-\tilde{\Lambda}^k)\rangle\ge 0, & \forall\, G_2\in\partial\|\tilde{A}^k\|_*,\ \forall A'\in\mathcal{R}^{m\times n}.\end{cases} \tag{6.16}$$

Also, (6.8) and (6.15) together imply that

$$\lim_{k\to\infty}(Z^{k+1}+\tilde{E}^k+\tilde{A}^k-M)=0.$$

Recall that we have assumed that notations $E^{k+1}=\tilde{E}^k$, $A^{k+1}=\tilde{A}^k$. Hence, we have

$$\lim_{k\to\infty}(Z^{k+1}+E^{k+1}+A^{k+1}-M)=0. \tag{6.17}$$

On the other hand, combining (6.2) and (6.3), we get

$$\Lambda^{k+1}=\Lambda^k-\beta(Z^{k+1}+E^{k+1}+A^{k+1}-M).$$

Therefore, (6.3) and (6.15) indicate that $\lim_{k\to\infty}\|\Lambda^{k+1}-\tilde{\Lambda}^k\|=0$.

Based on all these facts, we then have

$$\begin{cases}\lim_{k\to\infty}\langle Z'-Z^{k+1},-\Lambda^{k+1}\rangle\ge 0, & \forall Z'\in\mathbf{B};\\ \lim_{k\to\infty}\langle E'-E^{k+1},\tau G_1-\Lambda^{k+1}\rangle\ge 0, & \forall\, G_1\in\partial\|E^{k+1}\|_1,\ \forall E'\in\mathcal{R}^{m\times n};\\ \lim_{k\to\infty}\langle A'-A^{k+1},G_2-\Lambda^{k+1}\rangle\ge 0, & \forall\, G_2\in\partial\|A^{k+1}\|_*,\ \forall A'\in\mathcal{R}^{m\times n};\\ \lim_{k\to\infty}Z^{k+1}+E^{k+1}+A^{k+1}-M=0.\end{cases} \tag{6.18}$$

Then, it is obvious that any cluster point of $\{W^k\}$ is a solution point of (3.2), which is the optimal condition of (3.1). The first assertion is thus proved.

The second assertion is immediately implied by the fact that $\{V^k\}$ is Fejèr monotone with respect to the set $\mathcal{V}^*$; see, e.g., [1]. □

Compared to the ASALM, the VASALM abandons some latest iterative information during iterations and involves additional computation of updating the Lagrange multiplier. By doing so, the difficulty of proving the convergence of the ASALM under mild conditions on $\beta$ (e.g., any positive constant as required by the VASALM ) can be much alleviated.

*Remark.* Although we present the VASALM for the particular problem (3.1), we must point out that this method can be extended to solve the more general case—the linearly constrained convex programming problem whose objective function is separable into three parts. More specifically, consider the problem

$$\begin{array}{rl} \min_{X_1,X_2,X_3} & \theta_1(X_1)+\theta_2(X_2)+\theta_3(X_3) \\ s.t. & \mathcal{B}_1(X_1)+\mathcal{B}_2(X_2)+\mathcal{B}_3(X_3)=D, \\ & X_1\in\Omega_1,\ X_2\in\Omega_2,\ X_3\in\Omega_3, \end{array} \tag{6.19}$$

where $\Omega_i$ $(i=1,2,3)$ are given closed convex subsets in $\mathcal{R}^{m\times n}$, $\theta_i:\Omega_i\to\mathcal{R}$ $(i=1,2,3)$ are given proper convex functions (not necessarily smooth), $\mathcal{B}_i:\Omega_i\to\mathcal{R}^{m\times n}$ $(i=1,2,3)$ are given linear operators, and $D\in\mathcal{R}^{m\times n}$ is a given matrix. Obviously, this general model (6.19) includes (3.1) as a special case. Hence, in some sense, our proposed VASALM also contributes to the extension of the classical ADM from the well-studied case with two separable parts to the more general case with three separable parts.

**7. Extension.** An alternative model to study (1.2) is the following nuclear-norm- and $l_1$–norm-regularized least squares problem

$$\min_{A,E}\|A\|_*+\tau\|E\|_1+\frac{1}{2\mu}\|P_\Omega(C-A-E)\|_F^2; \tag{7.1}$$

see, e.g., [4, 43, 54]. In this section, we extend the proposed ASALM to solve (7.1) for the completeness.

By reformulating (7.1) into the favorable form

$$\begin{array}{rl} \min_{A,E,Z} & \|A\|_*+\tau\|E\|_1+\frac{1}{2\mu}\|P_\Omega(Z)\|_F^2 \\ s.t. & A+E+Z=M, \end{array} \tag{7.2}$$

where $M=P_\Omega(C)$ as defined before, then the proposed ASALM is easily extended. In fact, the iterative scheme of the ASALM for (7.2) consists of the following subproblems:

$$\left\{\begin{array}{l} Z^{k+1}\in\arg\min_{Z\in\mathcal{R}^{m\times n}}\frac{1}{2\mu}\|P_\Omega(Z)\|_F^2+\frac{\beta}{2}\|Z+A^k+E^k-\frac{1}{\beta}\Lambda^k-M\|_F^2, \\ E^{k+1}\in\arg\min_{E\in\mathcal{R}^{m\times n}}\tau\|E\|_1+\frac{\beta}{2}\|E+A^k+Z^{k+1}-\frac{1}{\beta}\Lambda^k-M\|_F^2, \\ A^{k+1}\in\arg\min_{A\in\mathcal{R}^{m\times n}}\|A\|_*+\frac{\beta}{2}\|A+E^{k+1}+Z^{k+1}-\frac{1}{\beta}\Lambda^k-M\|_F^2, \\ \Lambda^{k+1}=\Lambda^k-\beta(A^{k+1}+E^{k+1}+Z^{k+1}-M). \end{array}\right.$$

Let $\beta>0$ and $(A^k,E^k,\Lambda^k)$ be given. Then, the ASALM for (7.2) generates the new iterate $(A^{k+1},E^{k+1},Z^{k+1},\Lambda^{k+1})$ via the following computation.

**The $k$-th iteration of the extended ASALM for (7.2):**
1. Compute $Z^{k+1}$ via

$$Z_{ij}^{k+1} = \begin{cases} N_{ij}^k, & \text{if } (i,j) \notin \Omega; \\ \frac{\mu\beta}{1+\mu\beta} N_{ij}^k, & \text{if } (i,j) \in \Omega \end{cases}$$

with $N^k = \frac{1}{\beta}\Lambda^k + M - A^k - E^k$.
2. Compute $E^{k+1}$ via $E^{k+1} = \mathcal{S}_{\tau/\beta}(\frac{1}{\beta}\Lambda^k + M - A^k - Z^{k+1})$.
3. Compute $A^{k+1}$ via $A^{k+1} = \mathcal{D}_{1/\beta}(\frac{1}{\beta}\Lambda^k + M - Z^{k+1} - E^{k+1})$.
4. Update $\Lambda^{k+1}$ via $\Lambda^{k+1} = \Lambda^k - \beta(A^{k+1} + E^{k+1} + Z^{k+1} - M)$.

In the same way, we can extend the VASALM to solve (7.2). Let $\eta > 2$, $\beta > 0$, and $(A^k, E^k, \Lambda^k)$ be given. Then, the VASALM generates the new iterate $(Z^{k+1}, E^{k+1}, A^{k+1}, \Lambda^{k+1})$ via the following computation.

**The $k$-th iteration of the VASALM for (7.2):**
1. Compute $Z^{k+1}$ via

$$Z_{ij}^{k+1} = \begin{cases} N_{ij}^k, & \text{if } (i,j) \notin \Omega; \\ \frac{\mu\beta}{1+\mu\beta} N_{ij}^k, & \text{if } (i,j) \in \Omega \end{cases}$$

with $N^k = \frac{1}{\beta}\Lambda^k + M - A^k - E^k$.
2. Compute

$$\tilde{\Lambda}^k := \Lambda^k - \beta(A^k + E^k + Z^{k+1} - M).$$

3. Compute $E^{k+1}$ via $E^{k+1} = \mathcal{S}_{\tau/\beta\eta}(E^k + \frac{1}{\beta\eta}\tilde{\Lambda}^k)$.
4. Compute $A^{k+1}$ via $A^{k+1} = \mathcal{D}_{1/\beta\eta}(A^k + \frac{1}{\beta\eta}\tilde{\Lambda}^k)$.
5. Update $\Lambda^{k+1}$ via

$$\Lambda^{k+1} = \tilde{\Lambda}^k + \beta(E^k - E^{k+1}) + \beta(A^k - A^{k+1}).$$

**8. Numerical experiments.** In this section, we test the proposed ASALM and VASALM (further abbreviated as AS and VA if needed) for some examples and report the numerical results. Through these numerical experiments, we verify empirically that the model (1.2) is very powerful for accurately recovering low-rank and sparse components from incomplete and noisy observations, and we also demonstrate numerically that the proposed algorithms for solving (1.2) are very efficient.

We need to specify some techniques to be used in the practical implementation of the proposed algorithms. As we have pointed out, the computation of each iteration of the proposed algorithms is dominated by an SVD. However, according to the concrete shrinkage operator (see (2.2)), at each iteration we only need those singular values that are larger than the particular threshold and their corresponding singular vectors. Hence, partial SVD can be implemented to save considerately the computation of SVD. The popular software package PROPACK [37] is widely acknowledged in the community for the purpose of partial SVD, and it has been used in some related references, e.g., [4, 54]. As is well known, however, PROPACK is not able to compute automatically those singular values larger than the particular threshold; instead it does have the ability to compute the first $p$ singular values for any given $p \leq r$. Hence, at

the $k$-th iteration, we have to heuristically determine the number of singular values to be computed by partial SVD, which is denoted by $sv^k$. Let $d = \min\{m, n\}$. According to our numerical experiments, we suggest to determine $sv^k$ in the following manner. When $d > 500$, we adjust the value of $sv^k$ dynamically. More specifically, let $svp^k$ denote the number of singular values in the latest iteration that are larger than $1/\beta$. Starting from $sv^0 = d/10$, we update $sv^k$ iteratively via

$$sv^{k+1} = \begin{cases} \min(svp^k + \text{round}(0.04 * d), d), & \text{if } svp^k < sv^k; \\ \min(svp^k, d), & \text{if } svp^k = sv^k; \\ \max(svp^k - \text{round}(0.02 * d), 10), & \text{if } svp^k > sv^k. \end{cases}$$

When $d \leq 500$, we first execute full SVD for $A_k$ until no variation of its rank occurs. Then, we switch to PROPACK to execute partial SVD.

In the following, `sr`, `spr`, and `rr` represent the ratios of sample (observed) entries (i.e., $|\Omega|/mn$), the number of nonzero entries of $E$ (i.e., $\|E\|_0/mn$), and the rank of $A^*$ (i.e., $r/m$), respectively.

All the codes were written by MATLAB 7.8 (R2009a) and were run on a T6500 notebook with the Intel Core 2 Duo CPU at 2.1 GHz and 2 GB of memory.

**8.1. Gaussian-noiseless case.** In this subsection, we apply the proposed ASALM and VASALM to solve the Gaussian-noiseless case of (1.2), where $\delta = 0$ and $m = n$, i.e., (1.4).

The implementation of the ASALM and VASALM were terminated whenever the following stopping criterion is satisfied:

$$\text{RelChg} := \frac{\|(A^{k+1}, E^{k+1}) - (A^k, E^k)\|_F}{\|(A^k, E^k)\|_F + 1} \leq 10^{-6}, \tag{8.1}$$

which measures the relative change of the recovered low-rank and sparse components. We denote by $(\hat{A}, \hat{E})$ the iterate when the criterion (8.1) for successful recovery is achieved.

Let $C = A^* + E^*$ be the data matrix, where $A^*$ and $E^*$ are the low-rank and sparse components to be recovered. We generate $A^*$ by $A = LR^T$, where $L$ and $R$ are independent $m \times r$ matrices whose elements are independently and identically distributed (i.i.d.). Gaussian random variables with zero means and unit variance. Hence, the rank of $A^*$ is `r`. The index of observed entries, i.e., $\Omega$, is determined at random. The support $\Gamma \subset \Omega$ of the impulsive noise $E^*$ (sparse but large) is chosen uniformly at random, and the nonzero entries of $E^*$ are i.i.d. uniformly in the interval $[-500, 500]$. We set $\tau = 1/\sqrt{n}$ in (1.2) and start the iteration with the initial iterate $(A^0, E^0, Z^0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$.

The SVD computation deserves further explanation. As we remarked before, PROPACK with an appropriate scheme determining $sv^k$ can save computation considerately because only a partial SVD is executed at each iteration. In our numerical results, we found out that the technique of partial SVD works very well for both the ASALM and VASALM. To illustrate this observation, for (1.4) with $m = n = 500$, `rr` = `spr` = `0.05`, and `sr` = `0.6`, we execute the full SVD (by Mex File in C interface) to compute the exact rank of the iterate $A^k$ generated respectively by the ASALM and VSALM, and we record the variation of rank for these two methods. The comparison of their respective variation is listed in the left of Figure 8.1. For both of the methods, $\beta = 0.15\frac{|\Omega|}{\|P_\Omega(C)\|_1}$, and the parameter $\eta = 1.4$ for the VASALM.

As shown in Figure 8.1, the rank of iterates generated by the VASALM changes a bit more radically than those by the ASALM for the first few iterations. But for

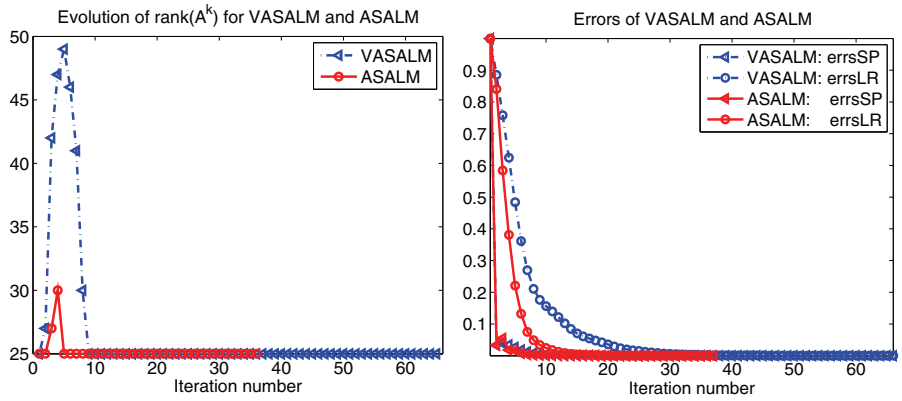


FIG. 8.1. *Evolution of rank (left) and the errors of low-rank and sparse components (right) for the ASALM and VASALM.*

both of these methods, the rank of iterates achieves the true low-rank immediately and the low-rank feature is preserved stably throughout the iterations. This nice feature makes the proposed ASALM and VASALM very suitable for the application of PROPACK, and it contributes much to the impressive numerical performance of the ASALM and VASALM, as we shall report soon. For exposing the difference of these two methods, we also compare their respective variations of the low-rank error ($errsLR := \frac{\|A^k - A^*\|_F}{\|A^*\|_F}$) and sparse error ($errsSP := \frac{\|E^k - E^*\|_F}{\|E^*\|_F}$) in the right of Figure 8.1. We amplify parts of the curves of the right figure of Figure 8.1 in Figure 8.2, in order to expose the different behaviors of these two splitting methods clearly.

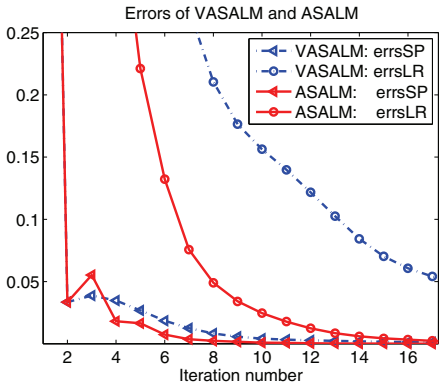


FIG. 8.2. *Amplification of local zoom for the right figure of Figure* 8.1.

In the first set of experiments, we test the proposed splitting methods for the case $m = n = 500$ with different values of `sr`, `rr`, and `spr` and compare their numerical performance. The involved parameters of these three methods are determined as follows. More specifically, let $\beta_0 = \frac{|\Omega|}{\|P_\Omega(C)\|_1}$. When implementing the ASALM and VASALM, we take the value of $\beta$ as suggested in Table 8.1 for various scenarios. Note that the value of $\eta$ for the VASALM is also included in Table 8.1.

TABLE 8.1
*Parameter values of the VASALM and ASALM*

| rr | spr | sr | VASALM | | ASALM |
|---|---|---|---|---|---|
| | | | $\beta$ | $\eta$ | $\beta$ |
| 0.05 | 0.05 | 0.9 | $0.12\beta_0$ | 1.3 | $0.15\beta_0$ |
| | | 0.8 | $0.12\beta_0$ | 1.3 | $0.14\beta_0$ |
| | | 0.7 | $0.09\beta_0$ | 1.3 | $0.14\beta_0$ |
| | | 0.6 | $0.1\beta_0$ | 1.4 | $0.12\beta_0$ |
| | 0.1 | 0.9 | $0.12\beta_0$ | 1.3 | $0.26\beta_0$ |
| | | 0.8 | $0.1\beta_0$ | 1.3 | $0.25\beta_0$ |
| | | 0.7 | $0.13\beta_0$ | 1.3 | $0.26\beta_0$ |
| | | 0.6 | $0.14\beta_0$ | 1.5 | $0.3\beta_0$ |
| 0.1 | 0.05 | 0.9 | $0.1\beta_0$ | 1.3 | $0.15\beta_0$ |
| | | 0.8 | $0.1\beta_0$ | 1.4 | $0.16\beta_0$ |
| | | 0.7 | $0.1\beta_0$ | 1.5 | $0.16\beta_0$ |
| | 0.1 | 0.9 | $0.12\beta_0$ | 1.3 | $0.26\beta_0$ |
| | | 0.8 | $0.17\beta_0$ | 1.3 | $0.26\beta_0$ |
| | | 0.7 | $0.26\beta_0$ | 1.6 | $0.25\beta_0$ |

TABLE 8.2
*Numerical comparison of the VASALM and ASALM for* (1.4).

| $m = n = 500$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| rr | spr | sr | $\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$ | | $\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$ | | Time (s) | | #SVD | |
| | | | VA | AS | VA | AS | VA | AS | VA | AS |
| 0.05 | 0.05 | 0.9 | 8.68e-6 | 2.87e-7 | 4.29e-5 | 2.95e-6 | 15.4 | 11.5 | 38 | 26 |
| | | 0.8 | 5.18e-7 | 3.05e-7 | 1.85e-5 | 3.67e-7 | 20.4 | 11.3 | 48 | 26 |
| | | 0.7 | 1.42e-5 | 2.47e-7 | 9.14e-5 | 3.67e-6 | 23.7 | 13.0 | 59 | 33 |
| | | 0.6 | 8.76e-6 | 8.73e-6 | 7.04e-5 | 6.29e-5 | 30.3 | 15.1 | 65 | 38 |
| | 0.1 | 0.9 | 2.21e-5 | 6.29e-6 | 1.51e-4 | 4.52e-5 | 14.8 | 15.0 | 35 | 25 |
| | | 0.8 | 2.22e-5 | 6.30e-6 | 1.75e-4 | 5.27e-5 | 19.6 | 15.3 | 47 | 28 |
| | | 0.7 | 2.07e-5 | 6.16e-6 | 1.96e-4 | 5.89e-5 | 23.5 | 17.4 | 50 | 35 |
| | | 0.6 | 2.58e-5 | 8.68e-6 | 4.86e-4 | 1.03e-4 | 35.5 | 22.1 | 74 | 37 |
| 0.1 | 0.05 | 0.9 | 8.50e-7 | 3.84e-7 | 1.41e-5 | 3.05e-6 | 23.7 | 15.5 | 48 | 27 |
| | | 0.8 | 7.47e-7 | 3.27e-7 | 1.82e-5 | 6.19e-6 | 32.6 | 19.4 | 60 | 32 |
| | | 0.7 | 9.75e-6 | 3.77e-7 | 1.17e-4 | 7.79e-6 | 41.7 | 23.5 | 76 | 39 |
| | 0.1 | 0.9 | 2.64e-5 | 9.47e-6 | 1.85e-4 | 6.84e-5 | 24.6 | 17.7 | 49 | 28 |
| | | 0.8 | 9.60e-6 | 9.34e-6 | 8.23e-5 | 7.81e-5 | 35.2 | 20.1 | 63 | 34 |
| | | 0.7 | 1.85e-4 | 1.73e-4 | 3.04e-3 | 3.85e-3 | 137.4 | 112.0 | 164 | 135 |

The numerical performance is reported in Table 8.2. We report the relative error of the recovered sparse component ($\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$), the recovered low-rank component ($\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$), the time in seconds (Time(s)), and the number of SVD (#SVD).

We point out that the VASALM should be more efficient than what we report in Table 8.2, as the subproblems that are eligible for parallel computation are actually solved in the consecutive order in our experiments. The efficiency of the VASALM can be improved if computational infrastructures for parallel computation are available.

To witness the efficiency of the ASALM, we test more scenarios and report the results in Table 8.3. The involved parameter $\beta$ is also determined by Table 8.1.

**8.2. Gaussian-noisy case.** In this subsection we further study the ASALM and VASALM to solve the Gaussian-noisy case, i.e., (1.2).

The numerical performance reported in Table 8.2 and Table 8.3 is dependent on the empirically chosen values of $\beta$. However, the efficiency of the ASALM and VASALM is not degraded much if the the value of $\beta$ is chosen by simpler strategy.

TABLE 8.3
*Recovery results of the ASALM for* (1.4) *with* sr = 80%.

| n | rr | spr | $\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$ | $\|\hat{E}\|_0$ | $\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$ | rank($\hat{A}$) | It. | Time (s) | #SVD |
|---|---|---|---|---|---|---|---|---|---|
| 200 | 0.05 | 0.05 | 3.36e-7 | 2000 | 2.73e-6 | 10 | 26 | 1.9 | 26 |
| | | 0.1 | 3.05e-5 | 4043 | 3.85e-4 | 10 | 29 | 2.0 | 29 |
| | 0.1 | 0.05 | 3.20e-7 | 2000 | 5.90e-6 | 20 | 26 | 2.9 | 29 |
| | | 0.1 | 1.47e-6 | 4004 | 1.30e-5 | 20 | 64 | 5.5 | 64 |
| 800 | 0.05 | 0.05 | 4.64e-6 | 31995 | 2.05e-5 | 40 | 34 | 48.5 | 34 |
| | | 0.1 | 7.46e-6 | 63979 | 4.92e-5 | 40 | 36 | 57.6 | 36 |
| | 0.1 | 0.05 | 1.03e-6 | 31997 | 7.96e-6 | 80 | 38 | 86.2 | 40 |
| | | 0.1 | 6.53e-6 | 64017 | 4.39e-5 | 80 | 45 | 105.0 | 47 |
| 1000 | 0.05 | 0.05 | 2.40e-6 | 49991 | 9.80e-6 | 50 | 31 | 80.1 | 31 |
| | | 0.1 | 5.51e-6 | 99972 | 3.32e-5 | 50 | 41 | 99.3 | 41 |
| | 0.1 | 0.05 | 8.94e-7 | 49999 | 6.97e-6 | 100 | 42 | 184.5 | 44 |
| | | 0.1 | 2.80e-6 | 99998 | 1.77e-5 | 100 | 44 | 212.3 | 46 |
| 1500 | 0.05 | 0.05 | 7.18e-7 | 112493 | 7.56e-6 | 75 | 31 | 284.8 | 31 |
| | | 0.1 | 3.34e-6 | 224955 | 1.60e-5 | 75 | 38 | 288.3 | 39 |
| | 0.1 | 0.05 | 1.19e-6 | 112492 | 5.80e-6 | 150 | 44 | 745.3 | 50 |
| | | 0.1 | 2.52e-6 | 224970 | 1.32e-5 | 150 | 45 | 722.0 | 49 |
| 2000 | 0.05 | 0.05 | 6.33e-7 | 199990 | 3.83e-6 | 100 | 34 | 830.6 | 35 |
| | | 0.1 | 1.10e-6 | 399965 | 6.92e-6 | 100 | 35 | 743.0 | 35 |
| | 0.1 | 0.05 | 1.25e-6 | 199984 | 5.27e-6 | 200 | 45 | 1616.3 | 48 |
| | | 0.1 | 8.76e-7 | 399981 | 6.54e-6 | 200 | 49 | 2665.2 | 51 |

To see it, in this subsection, when both the ASALM and VASALM are implemented, the value of $\beta$ is determined simply by

$$\beta = \begin{cases} 0.1\frac{|\Omega|}{\|P_\Omega(C)\|_1}, & \text{if } \texttt{spr} = 0.05; \\ 0.15\frac{|\Omega|}{\|P_\Omega(C)\|_1}, & \text{if } \texttt{spr} = 0.1. \end{cases} \tag{8.2}$$

In addition, for the VASALM, $\eta$ is taken by

$$\eta = \begin{cases} 1.4, & \text{if } \texttt{spr} = 0.05, \\ 1.5, & \text{if } \texttt{spr} = 0.1. \end{cases} \tag{8.3}$$

As clarified in [6], if the Gaussian noise is a white noise with standard deviation $\sigma$, then the parameter $\delta$ in (1.2) satisfies $\delta^2 \le (d + \sqrt{8d})\sigma^2$ with high probability. Therefore, we set $\delta = \sqrt{d + \sqrt{8d}}\sigma$ in (1.2) when the ASALM and VASALM are applied. Furthermore, note that relative errors that are much lower than the noise level would only prolong computational time without the benefit of getting a higher accuracy. Hence, for the Gaussian-noisy case (1.2), we revise the stopping criterion (8.1) into

$$\text{RelChg} \triangleq \frac{\|(A^{k+1}, E^{k+1}) - (A^k, E^k)\|_F}{\|(A^k, E^k)\|_F + 1} \le 0.1\sigma. \tag{8.4}$$

We generate the data $C$ exactly as the last subsection, and report some numerical results of both methods for (1.2) for the scenarios that $m = n = 500, 1000, 1500, 2000$, and $(\texttt{rr}, \texttt{spr}) = (5\%, 5\%)$, $(5\%, 10\%)$, $(10\%, 5\%)$, $(10\%, 10\%)$, for the particular case where $\texttt{sr} = 80\%$ and $\sigma = 10^{-3}$ in Table 8.4.

From Table 8.4, we see that the proposed model (1.2) is powerful for recovering low-rank and sparse components of matrices even for the Gaussian-noisy case, and that the proposed ASALM is more efficient for solving the model (1.2). In addition,

TABLE 8.4
*Recovery results of the VASALM and ASALM for* (1.2) *with* sr = 80% *and* $\sigma = 10^{-3}$.

| n | rr | spr | $\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$ | | $\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$ | | rank($\hat{A}$) | | Time (s) | | #SVD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | VA | AS | VA | AS | VA | AS | VA | AS | VA | AS |
| 500 | 0.05 | 0.05 | 7.84e-5 | 4.78e-5 | 1.31e-3 | 2.27e-4 | 25 | 25 | 11.0 | 8.2 | 25 | 16 |
| | | 0.1 | 8.29e-5 | 5.29e-5 | 1.69e-3 | 3.92e-4 | 25 | 25 | 11.7 | 7.7 | 27 | 16 |
| | 0.1 | 0.05 | 1.14e-4 | 5.34e-5 | 1.36e-3 | 3.97e-4 | 50 | 50 | 14.7 | 10.8 | 27 | 17 |
| | | 0.1 | 1.88e-4 | 8.56e-5 | 2.72e-3 | 8.50e-4 | 50 | 50 | 17.9 | 13.2 | 28 | 17 |
| 1000 | 0.05 | 0.05 | 7.44e-5 | 3.69e-5 | 1.28e-3 | 2.75e-4 | 50 | 50 | 85.0 | 61.5 | 29 | 16 |
| | | 0.1 | 8.87e-5 | 4.97e-5 | 1.77e-3 | 5.01e-4 | 50 | 50 | 94.2 | 60.2 | 30 | 16 |
| | 0.1 | 0.05 | 1.20e-4 | 6.48e-5 | 1.23e-3 | 5.97e-4 | 100 | 100 | 118.2 | 82.8 | 33 | 19 |
| | | 0.1 | 1.58e-4 | 7.74e-5 | 1.81e-3 | 8.34e-4 | 100 | 100 | 135.6 | 97.5 | 34 | 20 |
| 1500 | 0.05 | 0.05 | 7.51e-5 | 3.81e-5 | 1.25e-3 | 3.58e-4 | 75 | 75 | 265 | 180.9 | 32 | 17 |
| | | 0.1 | 9.35e-5 | 4.08e-5 | 1.74e-3 | 4.68e-4 | 75 | 75 | 288.2 | 207.3 | 33 | 18 |
| | 0.1 | 0.05 | 1.07e-4 | 4.37e-5 | 9.14e-4 | 3.57e-4 | 150 | 150 | 389.5 | 283.7 | 38 | 23 |
| | | 0.1 | 1.53e-4 | 7.03e-5 | 1.48e-3 | 7.01e-4 | 150 | 150 | 478.6 | 331.4 | 39 | 24 |
| 2000 | 0.05 | 0.05 | 6.52e-5 | 3.69e-5 | 1.02e-3 | 3.59e-4 | 100 | 100 | 693.6 | 423.0 | 35 | 18 |
| | | 0.1 | 9.06e-5 | 4.95e-5 | 1.57e-3 | 5.25e-4 | 100 | 100 | 756.6 | 475.3 | 36 | 19 |
| | 0.1 | 0.05 | 1.05e-4 | 5.21e-5 | 8.68e-4 | 3.91e-4 | 200 | 200 | 955.9 | 690.2 | 41 | 25 |
| | | 0.1 | 1.49e-4 | 5.15e-5 | 1.28e-3 | 4.54e-4 | 200 | 200 | 1099.9 | 838.8 | 43 | 28 |

TABLE 8.5
*Recovery results of the VASALM and ASALM for* (1.2) *with different* $\delta's$.

| $m = n = 800$, and noise level $\sigma = 10^{-3}$, $\delta_0 = \sqrt{(d+\sqrt{8d})}\sigma$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| rr | spr | $\delta$ | $\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$ | | $\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$ | | Time (s) | | #SVD | |
| | | | VA | AS | VA | AS | VA | AS | VA | AS |
| 0.05 | 0.05 | $\delta_0/10$ | 7.50e-5 | 3.85e-5 | 1.48e-3 | 2.40e-4 | 42.3 | 21.9 | 27 | 16 |
| | | $\delta_0$ | 7.50e-5 | 3.84e-5 | 1.48e-3 | 2.40e-4 | 43.2 | 21.2 | 27 | 16 |
| | | $10\delta_0$ | 7.50e-5 | 3.78e-5 | 1.48e-3 | 2.49e-4 | 42.1 | 21.2 | 27 | 16 |
| 0.05 | 0.1 | $\delta_0/10$ | 9.40e-5 | 5.10e-5 | 2.08e-3 | 4.32e-4 | 50 | 33.2 | 28 | 16 |
| | | $\delta_0$ | 9.40e-5 | 5.09e-5 | 2.08e-3 | 4.32e-4 | 49.3 | 32.6 | 28 | 16 |
| | | $10\delta_0$ | 9.41e-5 | 5.03e-5 | 2.08e-3 | 4.37e-4 | 49.5 | 32.7 | 28 | 16 |
| 0.1 | 0.05 | $\delta_0/10$ | 1.19e-4 | 6.39e-5 | 1.26e-3 | 5.61e-4 | 60.1 | 40.8 | 31 | 18 |
| | | $\delta_0$ | 1.19e-4 | 6.39e-5 | 1.26e-3 | 5.61e-4 | 59.8 | 40.2 | 31 | 18 |
| | | $10\delta_0$ | 1.19e-4 | 6.39e-5 | 1.27e-3 | 5.61e-4 | 59.8 | 40.7 | 31 | 18 |
| 0.1 | 0.01 | $\delta_0/10$ | 1.78e-4 | 7.07e-5 | 1.94e-3 | 7.43e-4 | 70.5 | 51.5 | 32 | 19 |
| | | $\delta_0$ | 1.78e-4 | 7.07e-5 | 1.94e-3 | 7.43e-4 | 70.5 | 51.8 | 32 | 19 |
| | | $10\delta_0$ | 1.78e-4 | 7.07e-5 | 1.94e-3 | 7.43e-4 | 69.5 | 51.5 | 32 | 19 |

to test the sensitivity of the ASALM and VASALM to the starting parameter $\delta$, we test them for (1.2) with various $\delta$ in Table 8.5 for the particular scenario where $m = n = 800$, $\sigma = 10^{-3}$, and sr = 80%.

Tables 8.4 and 8.5 show that the model (1.2) and both of the tested methods are very robust and efficient, and they are easy to handle since a wide range of values of $\delta$ can lead to successful recovery.

**8.3. The nuclear-norm- and $l_1$–norm-regularized least squares case.** In this section, we apply the ASALM and VASALM for solving (7.1), i.e., the nuclear-norm- and $l_1$–norm-regularized least square model for recovering low-rank and sparse components of matrices from incomplete and noisy observations.

We generate the data $C$ exactly as what we have done in section 8.1 and terminate the recovery once the criterion (8.4) is achieved. Again, $\beta$ is determined by (8.2) both for the ASALM and VASALM, and $\eta$ is taken by (8.3) for the VASALM. We set $\mu = \sqrt{d+\sqrt{8d}}\sigma/10$ in (7.1).

TABLE 8.6
*Recovery results of the VASALM and ASALM for (7.1) with* $\mathtt{sr} = 80\%$ *and* $\sigma = 10^{-3}$.

| n | rr | spr | $\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$ | | $\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$ | | rank($\hat{A}$) | | Time (s) | | #SVD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | VA | AS | VA | AS | VA | AS | VA | AS | VA | AS |
| 500 | 0.05 | 0.05 | 7.84e-5 | 4.78e-5 | 1.31e-3 | 2.27e-4 | 25 | 25 | 11 | 7.5 | 25 | 16 |
| | | 0.1 | 8.29e-5 | 5.25e-5 | 1.69e-3 | 3.92e-4 | 25 | 25 | 12.5 | 7.4 | 27 | 16 |
| | 0.1 | 0.05 | 1.14e-4 | 5.34e-5 | 1.36e-3 | 3.97e-4 | 50 | 50 | 15.2 | 10.6 | 27 | 17 |
| | | 0.1 | 1.88e-4 | 8.56e-5 | 2.72e-3 | 8.50e-4 | 50 | 50 | 18.3 | 12.7 | 28 | 17 |
| 1000 | 0.05 | 0.05 | 7.44e-5 | 3.68e-5 | 1.28e-3 | 2.76e-4 | 50 | 50 | 85.7 | 55.8 | 29 | 16 |
| | | 0.1 | 8.87e-5 | 4.96e-5 | 1.77e-3 | 5.01e-4 | 50 | 50 | 96.7 | 57.1 | 30 | 16 |
| | 0.1 | 0.05 | 1.20e-4 | 6.47e-5 | 1.23e-3 | 5.97e-4 | 100 | 100 | 120.6 | 74.6 | 33 | 19 |
| | | 0.1 | 1.58e-4 | 7.73e-5 | 1.81e-3 | 8.33e-4 | 100 | 100 | 138.7 | 90.6 | 34 | 20 |
| 1500 | 0.05 | 0.05 | 7.51e-5 | 3.82e-5 | 1.25e-3 | 3.59e-4 | 75 | 75 | 263.4 | 171.8 | 32 | 17 |
| | | 0.1 | 9.35e-5 | 4.08e-5 | 1.74e-3 | 4.68e-4 | 75 | 75 | 284 | 198.3 | 33 | 18 |
| | 0.1 | 0.05 | 1.07e-4 | 4.38e-5 | 9.14e-4 | 3.58e-4 | 150 | 150 | 391.5 | 273.2 | 38 | 23 |
| | | 0.1 | 1.53e-4 | 7.02e-5 | 1.48e-3 | 7.01e-4 | 150 | 150 | 457.2 | 317.7 | 39 | 24 |
| 2000 | 0.05 | 0.05 | 6.52e-5 | 3.96e-5 | 1.02e-3 | 3.59e-4 | 100 | 100 | 637.1 | 421.4 | 35 | 18 |
| | | 0.1 | 9.06e-5 | 4.96e-5 | 1.57e-3 | 5.26e-4 | 100 | 100 | 717.3 | 475.5 | 36 | 19 |
| | 0.1 | 0.05 | 1.05e-4 | 5.22e-5 | 8.68e-4 | 3.92e-4 | 200 | 200 | 961.8 | 688.4 | 41 | 25 |
| | | 0.1 | 1.49e-4 | 5.16e-5 | 1.28e-3 | 4.46e-4 | 200 | 200 | 1111.4 | 841.3 | 43 | 28 |

In Table 8.6, we report some numerical results of them for (7.1) for the scenario that

$$m = n = 500, 1000, 1500, 2000$$

and

$$(\mathtt{rr}, \mathtt{spr}) = (5\%, 5\%), (5\%, 10\%), (10\%, 5\%), (10\%, 10\%)$$

for the particular case where $\mathtt{sr} = 80\%$ and $\sigma = 10^{-3}$.

As we can see in Table 8.6, low-rank and sparse components of matrices can also be recovered very well in many cases via solving (7.1) by the proposed VASALM and ASALM.

Last, we test them for various starting parameter $\mu's$ in (7.1) to witness the sensitivity to the value of $\mu$. In particular, we choose $m = n = 800$, $\sigma = 10^{-3}$, $\mathtt{sr} = 80\%$, and report the numerical results in Table 8.7.

TABLE 8.7
*Recovery results of the VASALM and ASALM for* (7.1) *with different* $\mu's$.

| $m = n = 800$, and noise level $\sigma = 10^{-3}$, $\mu_0 = \sqrt{(d+\sqrt{8d})}\sigma/10$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| rr | spr | $\mu$ | $\frac{\|\hat{E}-E^*\|_F}{\|E^*\|_F}$ | | $\frac{\|\hat{A}-A^*\|_F}{\|A^*\|_F}$ | | Time (s) | | #SVD | |
| | | | VA | AS | VA | AS | VA | AS | VA | AS |
| 0.05 | 0.05 | $\mu_0/10$ | 7.50e-5 | 3.85e-5 | 1.48e-3 | 2.40e-4 | 43.3 | 21.2 | 27 | 16 |
| | | $\mu_0$ | 7.50e-5 | 3.84e-5 | 1.48e-3 | 2.40e-4 | 43.2 | 21.1 | 27 | 16 |
| | | $10\mu_0$ | 7.50e-5 | 3.76e-5 | 1.48e-3 | 2.53e-4 | 42.6 | 21.3 | 27 | 16 |
| 0.05 | 0.1 | $\mu_0/10$ | 9.40e-5 | 5.10e-5 | 2.08e-3 | 4.32e-4 | 49.9 | 33.1 | 28 | 16 |
| | | $\mu_0$ | 9.40e-5 | 5.09e-5 | 2.08e-3 | 4.32e-4 | 50.8 | 32.9 | 28 | 16 |
| | | $10\mu_0$ | 9.41e-5 | 5.00e-5 | 2.08e-3 | 4.42e-4 | 49.9 | 32.9 | 28 | 16 |
| 0.1 | 0.05 | $\mu_0/10$ | 9.55e-5 | 6.39e-5 | 1.26e-3 | 5.61e-4 | 61.7 | 41.3 | 31 | 18 |
| | | $\mu_0$ | 1.19e-4 | 6.38e-5 | 1.26e-3 | 5.60e-4 | 61.5 | 39.6 | 31 | 18 |
| | | $10\mu_0$ | 1.19e-4 | 6.31e-5 | 1.27e-3 | 5.55e-4 | 60.0 | 40.0 | 31 | 18 |
| 0.1 | 0.01 | $\mu_0/10$ | 1.78e-4 | 7.07e-5 | 1.94e-3 | 7.43e-4 | 75.6 | 50.7 | 32 | 19 |
| | | $\mu_0$ | 1.78e-4 | 7.06e-5 | 1.94e-3 | 7.43e-4 | 74.9 | 51.7 | 32 | 19 |
| | | $10\mu_0$ | 1.78e-4 | 7.05e-5 | 1.94e-3 | 7.43e-4 | 75.2 | 51.7 | 32 | 19 |

Tables 8.6 and 8.7 together show the attractive recovery ability of the model (7.1), and also the efficiency and robustness of the ASALM and VASALM for solving this model.

**9. Conclusions.** This paper puts forward the conceptual model for recovering low-rank and sparse components of matrices from incomplete and noisy observations to capture more concrete applications in reality. Then, some efficient algorithms for solving the proposed model are developed based on the ALM. Because of the full utilization of the desired structure emerging in the model, the proposed algorithms are very efficient even for large-scale cases and robust for various levels of noise. In addition, the resulting recovery components are very accurate. Hence, the recovery model is justified empirically.

As pointed out in [6], the topic of recovering low-rank and sparse components is a field in complete infancy. Research related to this paper, especially on the theoretical aspect, is abounding with great interests. Nevertheless, this paper launches the first model to accurately recover low-rank and sparse components of matrices from incomplete and noisy observations, and provides the efficient augmented-Lagrangian-based approach to solve the model numerically.

REFERENCES


[1] H. H. Bauschke and P. L. Combettes, *A weak-to-strong convergence principle for Fejr-monotone methods in Hilbert spaces*, Math. Oper. Res., 26 (2001), pp. 248–264.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[3] A. M. Bruckstein, D. L Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Rev., 51 (2009), pp. 34–81.

[4] J. F. Cai, E. J. Candés, and Z. W. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.

[5] E. J. Candés, X. D. Li, Y. Ma, and J. Wright, *Robust Principal Component Analysis?*, manuscript, 2009.

[6] E. J. Candés and Y. Plan, *Matrix completion with noise*, in Proceedings of the IEEE, 2009.

[7] E. J. Candés and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[8] E. J. Candés and T. Tao, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2009), pp. 2053–2080.

[9] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willskyc, *Rank-Sparsity Incoherence for Matrix Decomposition*, manuscript, 2009, http://arxiv.org/abs/0906.2220.

[10] C. H. Chen, B. S. He, and X. M. Yuan, *Matrix completion via alternating direction method*, IMA J. Numer. Anal., to appear.

[11] G. Chen and M. Teboulle, *A proximal-based decomposition method for convex minimization problems*, Math. Program., 64 (1994), pp. 81–101.

[12] S. Chen, D. Donoho, and M. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[13] J. Douglas and H. H. Rachford, *On the numerical solution of the heat conduction problem in 2 and 3 space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.

[14] D. Donoho, *High-dimensional data analysis: The curses and blessings of dimensionality*, AMS Math Challenges Lecture, 2000.

[15] J. Eckstein, *Some saddle-function splitting methods for convex programming*, Optim. Methods Softw., 4 (1994), pp. 75–83.

[16] J. Eckstein and M. Fukushima, *Some reformulation and applications of the alternating directions method of multipliers*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pordulos, eds., Kluwer Academic Publishers, Norwell, MA, 1994, pp. 115–134.

[17] E. Esser, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, manuscript, 2009, http://www.math.ucla.edu/applied/cam/.

[18] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity problems*, Springer Ser. Oper. Res. 1, Springer-Verlag, New York, 2003.
[19] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, Stanford, CA, 2002.
[20] M. Fazel and J. Goodman, *Approximations for Partially Coherent Optical Imaging Systems*, Technical report, Stanford University, Stanford, CA, 1998.
[21] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the American Control Conference, 2001, pp. 4734–4739.
[22] M. Fazel, H. Hindi and S. Boyd, *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, in Proceedings of the American Control Conference, 2003.
[23] M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming problems*, Comput. Optim. Appl., 1 (1992), pp. 93–111.
[24] D. Gabay, *Application of the method of multipliers to varuational inequalities*, in Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problem, M. Fortin and R. Glowinski, eds., Elsevier–North Holland, Amsterdam, The Netherlands, 1983, pp. 299–331.
[25] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.
[26] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, Heidelberg, 1984.
[27] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*, SIAM Stud. Appl. Math. 9, SIAM, Philadelphia, PA, 1989.
[28] B. S. He, L. Z. Liao, D. Han, and H. Yang, *A new inexact alternating directions method for monontone variational inequalities*, Math. Program., 92 (2002), pp. 103–118.
[29] B. S. He, M. H. Xu, and X. M. Yuan, *Solving large-scale least squares covariance matrix problems by alternating direction methods*, SIAM J. Matrix Anal. Appl., to appear.
[30] B. S. He and H. Yang, *Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities*, Oper. Res. Lett., 23 (1998), pp. 151–161.
[31] M. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.
[32] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.
[33] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
[34] R. H. Keshavan, A. Montanari, and S. Oh, *Matrix completion from a few entries*, IEEE Trans. Inform. Theory, to appear.
[35] R. H. Keshavan, A. Montanari, and S. Oh, *Matrix completion from noisy entries*, J. Math. Learn. Res., to appear.
[36] S. Kontogiorgis and R. R. Meyer, *A variable-penalty alternating directions method for convex optimization*, Math. Program., 83 (1998), pp. 29–53.
[37] R. M. Larsen, *Lanczos Bidiagonalization with Partial Reorthogonalization*, Technical report, Department of Computer Science, Aarhus University, Aarhus, Denmark, 1998.
[38] S. L. Lauritzen, *Graphical Models*, Oxford University Press, New York, 1996.
[39] Z. C. Lin, M. M. Chen and L. Q. Wu, and Y. Ma, *The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices*, manuscript, 2009.
[40] Z. C. Lin, A. Ganesh, J. Wright, L. Wu, M. Ghen, and Y. Ma, *Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-Rank Matrix*, preprint, 2009, http://perception.csl.illinois.edu/ jnwright/Lin2009-Algorithms.pdf.
[41] Y. J. Liu, D. F. Sun, and K. C. Toh, *An Implementable Proximal Point Algorithmic Framework for Nuclear Norm Minimization*, manuscript, 2009.
[42] Z. Liu and L. Vandenberghe, *Interior-point method for nuclear norm approximation with application to system identification*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1235–1256.
[43] S. Q. Ma, D. Goldfarb, and L. F. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., to appear.
[44] M. Ng, P. A. Weiss, and X. M. Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, SIAM J. Sci. Comput., 32 (2010), pp. 2710–2736.
[45] Y. C. Pati and T. Kailath, *Phase-shifting masks for microlithography: Automated design and mask requirements*, J. Opt. Soc. Amer. A, 11 (1994), pp. 2438–2452.
[46] M.J.D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 283–298.

[47] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.
[48] E. Sontag, *Mathematical Control Theory*, Springer-Verlag, New York, 1998.
[49] J. F. Sturm, *Using SeDuMi* 1.02*, a Matlab toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653.
[50] J. Sun and S. Zhang, *A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs*, European J. Oper. Res., to appear.
[51] M. Tao and J. F. Yang, *Alternating Direction Algorithms for Total Variation Deconvolution in Image Reconstruction*, http://www.optimization-online.org/DB_HTML/2009/11/2463.html (August 2009).
[52] P. Tseng, *Alternating projection-proximal methods for convex programming and variational inequalities*, SIAM J. Optim., 7 (1997), pp. 951–965.
[53] R. H. Tütüncü, K. C. Toh, and M. J. Todd, *Solving semidefinite-quadrtic-linear programs using SDPT*3, Math. Program., 95 (2003), pp. 189–217.
[54] K. C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, Pac. J. Optim., 6 (2010), pp. 615–640.
[55] L. G. Valiant, *Graph-theoretic arguments in low-level complexity*, in Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science, 1977, pp. 162–176.
[56] Z. W. Wen, D. Goldfarb, and W. Yin, *Alternating Direction Augmented Lagrangian Methods for Semidefinite Programming*, manuscript, 2009.
[57] J. F. Yang, W. T. Yin, Y. Zhang, and Y. L. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 569–592.
[58] J. Yang and Y. Zhang, *Alternating Direction Algorithms for l1-problems in Compressive Sensing*, Technical report 09-37, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2008.
[59] X. M. Yuan, *Alternating Direction Methods for Sparse Covariance Selection*, manuscript, 2009.
[60] X. M. Yuan and J. Yang, *Sparse and Low-Rank Matrix Decomposition Via Alternating Direction Methods*, manuscript, 2009.
[61] Z. Zhou, X. Li, J. Wright, E. J. Candés, and Y. Ma, *Stable principal component pursuit*, in Proceedings of the International Symposium on Information Theory, 2010.